UNIVERSITÀ DEGLI STUDI DI TORINO

DIPARTIMENTO DI SCIENZE VETERINARIE

Dottorato di Ricerca in

SCIENZE VETERINARIE PER LA SALUTE ANIMALE
E LA SICUREZZA ALIMENTARE

CICLO XXXIII

TOWARDS THE DEVELOPMENT OF INFORMATION SYSTEM IN PRECISION FARMING FOR THE REARING OF PIEMONTESE BOVINES

Tesi presentata da: Francesca Abbona

Tutor: Mario Giacobini

Coordinatore del dottorato: Maria Teresa Capucchio

ANNI ACCADEMICI 2017/2018-2018/2019-2019/2020

SETTORE SCIENTIFICO-DISCIPLINARE DI AFFERENZA INF/01

Contents

List of Figures

List of Tables

Abstract

The aim of this thesis is a comprehensive investigation of the possible improvements in modelling beef farm performance, in order to subsequently integrate it with information systems.

Using the Piemontese breed as a case study, the assessment of beef farm performance was investigated, as well as the attributes influencing the corresponding parameters. Critical points were identified in the measurement of breeding production efficiency, that revolves around cows' fertility and production, i.e., the calf quota generated yearly. With a particular focus on the weaning period, approximately two months after birth, it emerged that losses related to calf management are consistent, as viable calves go through a very delicate phase, conditioned by multiple factors. The need for a methodology towards the construction of a more appropriate model was outlined. In order to adequately address the issue, two main points needed to be handled. First, the necessity to cope with the management of Big Data and, second, the need for the identification of patterns among the variables, without introducing a priori knowledge or bias into the model. The approach that responds suitably to this complex issue is the popular Machine Learning, hence proposed and investigated as a flexible tool that, rather than making a priori assumptions, allows the system to learn directly from data. This approach uses indeed the data to continuously build and refine a model for making predictions. An introduction to the problem is given in details over the first three chapters, where a sufficiently thorough description of the scenario within which the research study sets is carried out. Similarly, the Chapter 4 is entirely dedicated to the description of Machine Learning principles, starting from the basic concepts behind its use, to then move on to the illustration of all the approaches applied in this research, their strengths, and their conceptual differences.

The research involved an initial pool of 725 representative farms, among which different subsets were extracted thereafter. Information about the farms was elicited from the National Herd-Book, managed by ANABORAPI, and preprocessed in order to apply different techniques. Additionally, information was collected through on field questionnaire, regarding also production systems, farm size, animal density, environmental conditions, and diet. Among the different sets of farms, distinct Machine Learning methods were applied. As the main purpose of this research was the identification of a technique able to exploit at the same time feature extraction and simple, intelligible models, the choice of applying Genetic Programming

seemed straightforward. It resulted appropriate for the development of the analysis, as it allowed also to exploit all the information contained in the dataset: for each breeding, it was possible to make a comparison using all the data recorded over several years, refining the prediction. Comparative studies with other usually enrolled prediction methods were investigated with promising results in the context of modeling the breeding performance of Piemontese cattle farms.

Chapter 1

Outline of the Study

The *Piemontese* breed is a cosmopolitan breed with prevailing meat characteristics which, until a few years ago, was limited within the boundaries of the region from which the name derives, i.e., Piemonte, in the north-west of Italy. Today it has crossed the boundaries of its primitive settlement area and is spreading in various foreign countries. The environment in which it is raised does not present the extensiveness conditions for the breeding of traditional beef breeds. It is an early and long-lived breed, suitable for being raised in the most diverse climates. Being an excellent food processor adaptable to more diverse environmental conditions, it can be advantageously bred on both flat and hilly pastures, as well as on the poorest mountain ones. Indeed, it is widespread in the plains stable breeding, sometimes integrated with the exit to grazes near the farm, but, for some farms, the practice of mountain grazing it is also common during the summer months, when the herd migrates to mountain pastures, remaining there until autumn. The feeding of the Piemontese cow is very simple and consists mainly of green, dried, or ensiled farm fodder, supplemented by cereals or legumes grown in the area. Calves are generally sold as soon as they are weaned. Weaning usually takes place on average within 2 months of age, rarely beyond. When they are not sold, they remain in the farm, implementing what goes by the name of the cow-calf type of breeding.

The peculiar characteristic of the Piemontese is the muscular hypertrophy, that appeared over the ′900 and it progressively spread among almost all animals registered in the *Herd-Book*. Caused by a genetic mutation of the gene of the myostatin, it entails a significant increase in muscle mass, due to an increase in the number of muscle fibres. The increased muscularity is also accompanied by a decrease in intramuscular fat and in connective tissue as well as, resulting in greater tenderness of the meat. Among the adverse effects originally associated with muscle hypertrophy, there is the reduction in reproductive efficiency, lower vitality, and the appearance of calf birth defects such as arthrogryposis and macroglossia. However, subjects that are poorly efficient from a reproductive point of view or not very viable generate a limited progeny. Artificial selection

performed by ANABORAPI (National Association of Piemontese Cattle Breeders) and natural selection have resulted in a strong reduction in the incidence of such problems. The Association keeps the Herd-book, runs a Genetic Station, where performance tests and progeny tests are carried out, and an Artificial Insemination (A.I.) Station where semen from A.I. bulls is produced. Therefore, it establishes the selection criteria.

1.1 Definition of Beef Farm Performance

If well managed, the current Piemontese cow is able to produce and raise almost one calf per year. The direction towards which modern Piemontese breeding aims is the production of calves for fattening. To maximize revenues, it is therefore essential that each mare produces as many calves as possible during her productive career, in full respect of her physiology. The indicator parameter of a cow's reproductive efficiency is represented by the *"calf quota"* per cow, derived form the *calving interval*. The latter parameter expresses the number of days between two deliveries. It includes two time intervals, i.e., the time span between birth and conception, and the time span of the actual gestation duration. Among these two time intervals, the period between birth and conception is the one that has the predominant importance for the purpose of reducing the length of the birth, while the duration of pregnancy is a limited and incompressible natural variable of 290 days. The calf quota can be either lower or higher than the unit. When smaller, the more it deviates from the unit, the lower the fertility of the mare was. Among particularly efficient cows, it is also possible to produce more than one calf in the same year, as cases of twin births are common. The two parameters can be considered as the meters of breeding production efficiency, in order to raise its technical and economic competitiveness. Therefore, by making the calf quota converge to 1, considering the calving interval to be 360 days long, the farm is economically profitable. The reproductive capacity of the cows that lodge on the farm significantly affects the farmer's income. Damage derives from the loss of income from the failure to give birth to calves and from the cost of feeding the cows. An accurate diagnosis of pregnancy is crucial to achieve and maintain the optimal reproductive performance of the farm. If a fertilized cow is not pregnant, it can be fertilized again. The minimum delay in this case is of fundamental importance. Birth is the moment in which the attention paid to the cows is harvested for at least a year. Normally the time between a delivery and a new conception counts around two and three months, after which a healthy and well-fed cow is able to face a new pregnancy. If the first insemination was successful, after nine months or a little later, a calf is born. But things do not always go as desired, and then insemination must be repeated perhaps several times and this inevitably extends the length of the birth, to the detriment of profitability. The Piemontese bovine, in the past, was characterized by considerable difficulties calving, also accepted by breeders as a necessary evil to obtain quality calves. Currently, the work of the ANABORAPI made it possible to implement a selection plan for calving ease and delivery. In this direction, great strides have

been made, above all, as regards the selection of animals capable of giving birth well and calves that are not excessively large, but able to develop excellent growth. These are the aspects that improve the breed's aptitude for giving birth. Since the process to improve calving ease is slow, it is also necessary to take advantage of all the technical and managerial factors of the herd, that can affect the trend of births on the farm. Cow management, in terms of feed and type of housing, a correct choice of mating, the possibility of having a suitable environment where to give birth, knowledge about birth event, allow one to set the conditions necessary for the optimal performance of this event. It is obvious that, among other things, the calving depends strictly on the fertility of the cow. Among the possible causes of a herd's fertility reduction, those intrinsic to feeding system, infectious, hygienic-sanitary, or endocrine-gynecological ones, and those of environmental nature are of major importance. Not to forget that all stressful conditions, such as uncomfortable housing, insufficient lighting, and crumbling shelters can negatively affect fertility and therefore the calving. Indeed, the free housing allows a greater mobility and a greater exposure to light with a positive influence on biological activity and consequently recovery after birth.

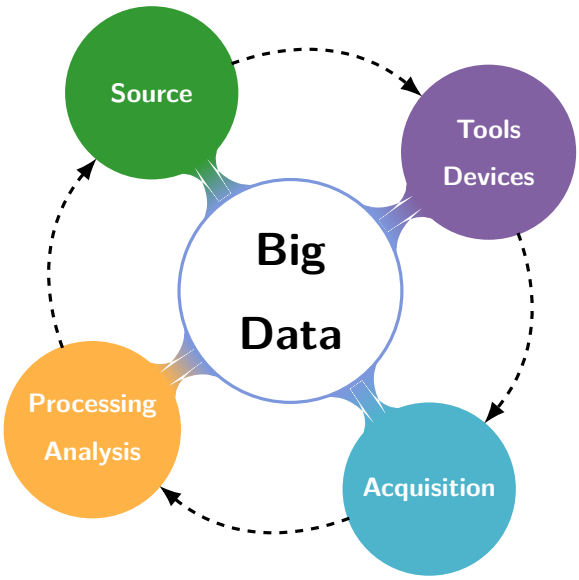
Calving and mortality detected on the farm at birth are combined through a model that provides the calf quota as performance measure. However, it is reductive to measure breeding performance by observing only fertility and maternal condition. The calf, on its side, goes through evolutionary stages that depend on its own condition. The phases immediately following birth, i.e., the intake of colostrum and the healthiness of the environment in which it lives, are of utmost importance. After the initial colostrum and milky feeding phase, weaning begins which, unlike the first period common to all and all young calves, differs considerably from one company to another. The velocity with which a calf is weaned varies according to the type of farming. As for the mother, during the weaning months it needs specific food and a favorable environment for growth. The physiological development process of the animal reaches completion in 60 days after birth. Calf mortality is also an important cause of economic damages in Piemontese cattle farms: for the farmer it represents the loss of the economic value of the calf, and the reduction of both the herd's genetic potential and the size of the breeding. It is straightforward that the gestational phase alone is not exhaustive. The breeding performance should be modelled considering also neonatal mortality, outlining the calf's ability to survive, and the sources of stress such as congenital calf's defects, compromising eventually the immune response and the growth rate, as well as environmental and food conditions, that affect the quality of life of the newborn. Zootechnical influential variables must be identified among the numerous parameters. Furthermore, the applied model is based on a priori zootechnical knowledge. It consists in a classic statistical model that provides a value for the parameter, deriving an argument from obvious propositions. Traditional statistical forecast analysis is preprogrammed, based on past data being a good forecasting indicator for the future.

This makes it necessary to formulate, among the data, a proper prediction measure for the yield of the farm, i.e., the number of calves weaned per cow per year.

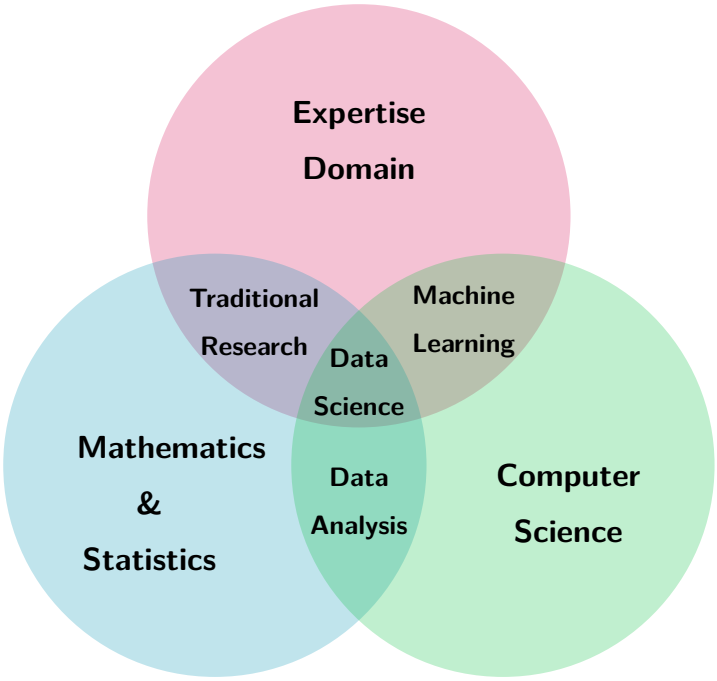
1.2 Improving Beef Farm Performance with Machine Learning

The digitization of data collection made it possible to streamline and accelerate the procedures of data collection and processing over time, permitting the registration and consequent processing of many additional data. Moreover, the management of livestock is also increasingly handled by continuous automated real-time monitoring, contributing to the increase of the amount of information and complexity among databases. This surveillance, going under the name of Precision Livestock Farming, allows the enhance production, reproduction, health and welfare of the herd, and its environmental impact. It supplements the skills of the farmer, the veterinarian, and the technician by a continuous collection of livestock information, with the support of information technologies. It can play a crucial role in the early detection of diseases and it objectively assesses animal condition and welfare in modern livestock production, representing a tool that supports many farmers as decision-makers. Each animal is monitored, contributing to a better definition of the whole breeding performance. The resulting increased knowledge, elaborated through mathematical models, may provide the offset of overall incurred costs of the farm, as these issues are identified in advance, allowing decisions to be made in time.

The major consequence of continuous monitoring of animals is a huge amount of data, the so-called "Big Data". Around big data there are various interconnected protagonists, starting from the source of information, in this case productive animals, and from whom (or what) it records the data, i.e., the farmer, the technicians, or the sensors installed. Beside this, the farmer not only records the data, but formalizes a specific need among a problem emerging within the breeding, for which he makes a request. Through the introspection of the acquired data, it is possible, by means of techniques and analyses, to provide an adequate answer. On the other hand, it is necessary to actively involve the breeder, as requiring information and precision are necessary to offer an answer as adequate as possible answer. It is not just a matter of exchanging useful services. Awareness and continuous active involvement of both parts are required. In particular, involving the farmer in an inclusive manner by requesting training and direct participation in the registration places the activity in the *citizen science* field.



If, on the one side, Precision Livestock Farming approach aims at a greater "accuracy" on the quantity and quality of information, entailing the development of monitoring systems, on the other side it must deal with the transformation of big data into meaningful information. New data is available, as well as increasingly cheaper and faster computing power. However, visual inspection is not adequate. The increase in the amount of data requires the introduction of new data management and prediction techniques, i.e., new tools that allow one to deal with a large amount of data offer the possibility of processing intrinsic information. Machine Learning is based on the availability of large amounts of data and on computing power. Rather than making a priori assumptions and following preprogrammed algorithms, Machine Learning allows the system to learn from data. It uses data to continuously build and refine a model for making predictions. It helps one to understand patterns between the attributes, detecting and modeling the variables along with the defined target. For this reason, the use of Machine Learning techniques is becoming increasingly common. Machine Learning, big data, knowledge, mathematics and statistical methods can be all combined powerfully, as all methods contribute to the inter-disciplinary field of Data Science. By means of scientific methods, algorithms, systems to extract knowledge, it is possible to investigate data. Machine Learning offers a wide range of techniques. There are many methods that can produce excellent results, by building accurate prediction models. Depending on intrinsic characteristics, they differ from each other and they can better address different tasks.



1.3 Aim and Outline

While working at ANABORAPI, after a first period spent training on the information systems implemented by the Association's technicians, I studied and acquired the notions of animal husbandry and Piemontese main characteristics. ANABORAPI provides a web service, accessible to registered users, which supplies the situation of the farm. Data entered by farmers and technicians by mean of PC and other devices are sent in real time to the servers, stored and processed, to return the updated situation at last. Data and statistics are finally provided to the farmers, that can consult them on their own. Thereafter, in order to investigate the production of Piemontese calves and to understand the mechanisms of breeding performance, I inspected the corresponding data and model. For an optimal management, besides the current situation of the farm, it is relevant for the breeder to know the prediction of the future trend, as well as relevant attributes. The main objective of this work was to perform a comprehensive investigation of the possibilities for the improvement of modelling farm performance, in order to be subsequently integrated with information systems. In order to manage the amount of data and use data to build predictive models, without introducing a priori knowledge, it is necessary to exploit the potential of Machine Learning techniques. Taking advantage of Machine Learning means entrusting the analysis of the available data to complex algorithms. Not only that: the potential of this type of approach is not limited to facilitating the analysis process, but is also linked to the possibility of recognizing complex patterns and of adapting easily to new data acquired over time. A broad variety of research studies is available in the dairy cattle sector, based on the application of

Machine Learning techniques in farm management, also due to the consistent use of sensor-based technology. On the other hand, the literature regarding beef cattle is not as large, in particular regarding Piemontese farm. Indeed, this work offers a novel line of investigation in the field of prediction modeling in the beef Piemontese breeding. Since the task and the data are full of zootechnical meaningful features, I explored the potential of different algorithms. Each of them adapts differently to the data and, above all, processes them according to different algorithms. The result may be better, worse, or similar to others in terms of accuracy of the result. However, they carry with them characteristics that make Machine Learning more interesting and appropriate, with respect to the objective to be pursued. Thanks to their structure, the algorithm can automatically create models "learning" from the data and produce accurate results. It was in fact possible to build prediction models starting from the data recorded during a specific historical period, i.e., one year, isolating the target for the same subsequent time period, i.e., the next year. The structure of the database allowed me to distinguish two types of analysis. As the dataset is a historical archive, it was possible to build models on the data extracted for a certain moment. On the other hand, it was also possible to fully exploit all the sequential information, contained in data varying over time. Indeed, a Genetic Programming approach was adopted, as generated models are resumed in simple and interpretable expressions, and they extract critical information, i.e., informative attributes. Why are such models sought? Mainly because the result should be available for further analysis, in an attempt to understand which link has been detected independently by the algorithm. Furthermore, if the expression turns out to be simple and legible, it can also be simple for the farmer to interpret the result. Genetic Programming also offers the possibility to handle vectorial variables representing time series, exploiting all the available information. Among other Machine Learning methods, some common methods were selected to compare the results obtained with Genetic Programming, that is known to be able to capture the strong non-linearity underlying data. I applied the methodology to different benchmark problems. Several datasets were isolated, first investigating the performance of Genetic Programming on the data contained in the Herd-Book, by selecting different subsets of variables. The results were compared with other techniques, first on "instant" data extracted at specific points in the timeline, referred to with the expression of *standard Genetic Programming* along the thesis. Subsequently, I investigated the behavior on vectorial variables, increasing the amount of information available as input for the different techniques, referred to as *vectorial Genetic Programming*. The preliminary results obtained with standard Genetic Programming (Chapter 6, *A GP approach for precision farming*) were presented at the 2020 international virtual conference WCCI, appearing in the conference proceedings [2]. Additional investigations among standard Genetic Programming including the corresponding comparative methods results (Chapter 7, *Towards modelling beef cattle management with Genetic Programming*) were published thereafter in *Livestock Science* journal [1]. Since the promising results were positively received by

the scientific community, while exploring the vectorial approach (Chapter 8), I conducted a parallel research on an enriched dataset, built by adding to the Herd-Book dataset a series of very informative zootechnical information. It was necessary to draw up a specific questionnaire (Chapter 9), to be filled in through farm visits in order to acquire the additional data on a set of representative breedings. In this regard, after planning the methodology to be pursued, I was assisted in learning the necessary notions to fill the questionnaire directly on the farm, and elicit the useful information.

Chapter 2

The Piemontese Cattle Breeding Outline

2.1 The Origin of the Piemontese Breed

The Piemontese is an Italian bovine breed native of Piedmont, a region in Northwestern Italy, highly specialized for beef production. This breed recalls distant roots, having its origins in two ancestors: a Pleistocene bovine of the Aurochs type and the Zebu. In a period between the Middle and Upper Palaeolithic, due to a massive migration, the Zebuin population originating in the Indian subcontinent, specifically western Pakistan, occupied various sectors of the European continent. After reaching Piedmont, the migratory wave halted, finding its path blocked by the Alps. Gradually, the Aurochs and Zebuin populations merged, creating a new one showing the current characteristics of the Piemontese breed. The first Piemontese farms appeared at the end of the 1800s when the population became uniform, fulfilling the criteria for being classified as a breed. Mainly spread between the Piedmontese hilly areas, i.e., Langhe and Canavese, and the plains on the right side of the Po river, the animals had a triple aptitude, that is, they were suitable for the production of both milk and meat, and for working purposes [10].

The history of the Piemontese reached then a turning point. In 1886, in the municipality of Guarene d'Alba, in the province of Cuneo, a bull appeared with considerable muscle mass, much more pronounced on thighs and buttocks. Not compliant with the breed standard of the time, the individuals showing this characteristic were seen with distrust and initially considered as pathological. The anomaly, which later attracted the interest of breeders, consumers, and scientists, was due to a genetic mutation causing muscle hypertrophy, which gave the animals pronounced shapes at the rump and thigh. However, the trait became soon the peculiarity, outlining a third category defined as the double muscling (**DM**), i.e., "doppia coscia" or "doppia groppa". The selection of DM individuals increased over time, allowing the specialization of the breed in this direction. A well-bred Piemontese head of cattle can nowadays exceed the 70% yield at the slaughterhouse, an exceptional result favoured also by the light bone and thin skin of the animal, reducing the waste to the

minimum amount. Besides carcass conformation and dressing percentage, DM exerts positive effects also on nutritional and palatability qualities, appreciated by both consumers and butchers, as the meat shows low intramuscular connective tissue and low-fat content [10, 21, 76]. The work provided in the past by these bovines has been replaced by agricultural mechanization, while milk production has been addressed with other specific cattle breeds. Milk yield of Piemontese cows is nevertheless more than sufficient for the maintenance needs of the calf, and some breeders use the additional milk to produce typical cheese. According to the data provided by the other breed associations, nowadays the Piemontese is the second breed in Italy, with around 300.000 cattle registered in the National Herd-Book, following only the Italian Holstein-Friesian cattle (Table 2.1).

Breed	**n. of cows**	**cows/farm**
Italian Holstein-Friesian	1.079.338	110
Piemontese	**133.425**	**31**
Italian Brown Swiss	71.333	14
Italian Simmental	64.554	12

Table 2.1: Cows registered in Herd-Books.
Data sources: A.N.A. National Associations of the corresponding Breed

Being the latter a dairy breed, the Piemontese is, therefore, the most raised among beef cattle, mainly bred in the provinces of Asti, Cuneo and Turin. However, many cattle farms are also present in other Italian regions and foreign countries, denoting its spread all over the world [10].

2.2 The Traits Determine the Type of Management

The hypertrophy characterizing the breed is due to a mutation of the gene located on chromosome 2, encoding for myostatin, a protein responsible of the balance between the muscular mass and the skeleton, that interacts by limiting growth. A lower production of myostatin entails indeed an increase in the number of muscular fibres, most prominently, but not their diameter [27, 33, 54]. There are several known gene mutations among beef cattle showing the hypertrophy trait. In the case of the Piemontese, the transition of a single Guanine nucleotide into Adenine is responsible for the genetic alteration, generating the inactive form of myostatin protein [42]. Homozygous individuals, i.e., possessing two copies of the mutated gene, exhibit a higher development of muscle mass than individuals in which the mutation is absent. Since the 1970s, all the bulls enabled for artificial insemination are homozygous for the gene mutation, entailing the establishment of the

character in the breed. Animals showing the mutated gene in heterozygosity, i.e., with one copy, exhibit instead rather intermediate characteristics [10].

The cattle is usually bred in beef intensive farms, which are therefore provided with the installation of stables to control the animals. The breeding of Piemontese cows is traditional with free-stall housing, currently much more common than fixed housing. The diet consists mainly of farm forages, green, dried or ensiled, supplemented by a feed based on cereals or legumes grown in the area. Also concerning fattening calves, the traditional fixed-post rearing systems are now frequently replaced by free-housing systems in boxes on permanent litter and their diet consists of feed produced on the farm, based on cereals and a fibrous source, i.e., hay or straw. However, the physical conformation determined by the genetic mutation permits also easy grazing management. The cows can be bred not only on flat and hilly pastures but also on the poorest mountain ones. The Piemontese bovine is thereby an excellent food processor, adaptable to the most diverse environmental conditions. Indeed, if stable breeding is widespread in the plain land, sometimes integrated with the grazing exploitation near the farm, it is also common for some farmers to use mountain pastures during summer, when the herd migrates to mountain even over 2,000 meters of altitude and remains there until autumn. It is a long-lived breed, well adaptable to the most diverse climates, responding well both in shed breeding and in wild or semi-wild breeding. Alongside the organoleptic qualities, the bovines exhibit remarkable zootechnical traits.

The number of registered farms is in constant growth, exceeding the threshold of 4300 breedings in 2019. Figure 2.1 shows the trend for the decade 2010-2020, showing the total number of heads of cattle and cows.

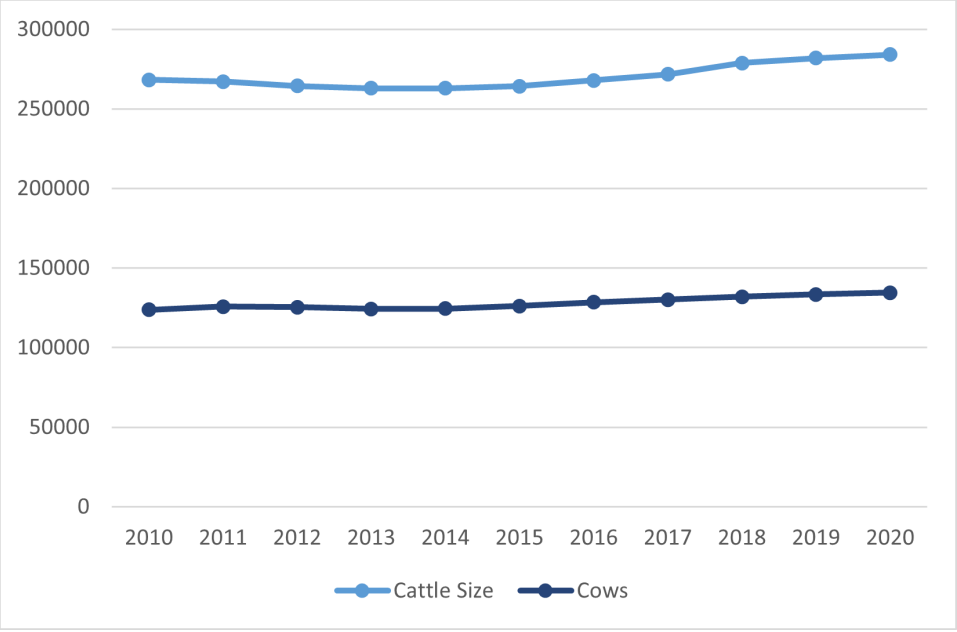


Figure 2.1: Trends in the overall population size during the 10-year period between 2010 and 2020. For each year, the graph shows the total number of bovines and cows.

It is clear from the graph (Figure 2.2) that 33% of breedings has very low cows consistencies, with less than 11 cows per farm. This represents only 5% of the whole cow population. On the contrary, almost 60% of the total cows are concentrated in 20% of the largest farms, counting more than 50 per farm.

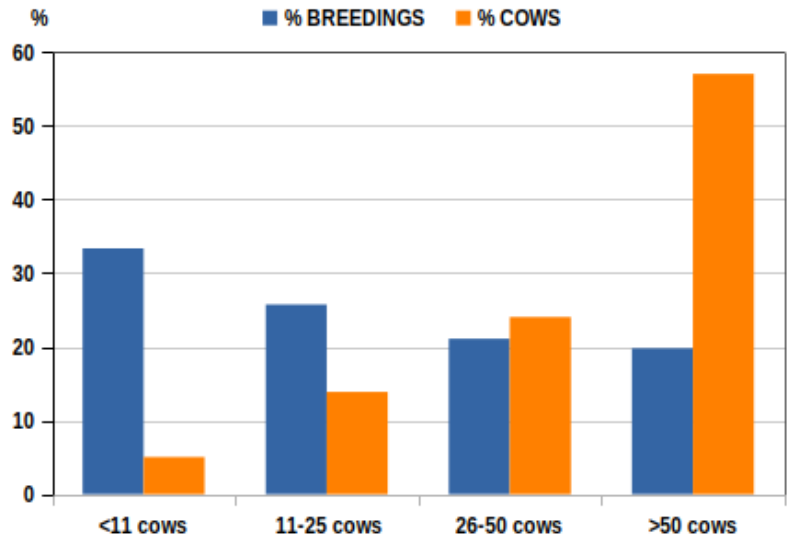


Figure 2.2: Distribution of breedings per size class and percentage of total cows per breeding size

2.3 The Enhancement of the Piemontese Cattle

The preservation of the Piemontese breed is guaranteed by the *National Association of Piemontese Cattle Breeders*, ANABORAPI in brief. ANABORAPI is responsible for promoting the breed through the study of the productive, reproductive, and management processes of the Piemontese breeding. One of the main activities includes the management of the Herd-Book of the Race. It is a complex database that preserves the pedigrees (i.e., 2.919.877 total pedigrees) of all the registered animals and a series of additional information, such as validation of breed characters, reproductive career, morphological studies, and genetic values. In this way, the farmers have access to the constant monitoring of the average situation of the cattle, receiving regular updates on fertility and productive cattle parameters. The values are also supported by a brief economic summary, which compares the gross revenue with the mortality losses, providing the farmer with an indicator of breeding performance.

Since DM can cause various degrees of subfertility, calving difficulty, lower calf viability and increased stress susceptibility, the breeding goal of the Piemontese population includes traits related to quantitative beef production (e.g. growth potential and muscularity) and reproduction, i.e., direct and maternal calving performance of animals [4, 6, 20, 53]. Genetic models take into account factors that influence the traits, such

as relationships between subjects and environmental factors. Regarding beef production traits, young bulls are tested on their performance. To this purpose, the Association avails a Genetic Station, in which 45 days old calves are monthly introduced based on their pedigree and morphology and reared in homogeneous conditions. The selected animals are then progeny tested for direct and maternal calving ease.

The selection contributes to the increase of genetic potential, by choosing the animals with the best gene combination of the different traits of interest. The goal is determined by the breeding structure, that in the beef cattle sector consists in a two-level system, i.e., suckler cows production and fattening production. For this reason, the selection indices are considered. A Selection Index is a linear combination of the values of the different traits, weighed appropriately according to the genetic correlations with the other ones, as well as its economic importance [5]. ANABORAPI provides regularly updated estimates so that the breeder can manage the breeding, plan the matings, check the progress constantly [10].

Chapter 3

The Role of Calf Weaning in Breeding Performance Measure

3.1 The Weaning of the Piemontese Calf: Aspects not to Underestimate

The breeding of the Piemontese does not provide a specialized orientation in the production. Compared to other beef breeds, which implement a cow-calf line with the calf sale after weaning to specialized fattening stations, three different farming systems emerge. The first one involves the 35-50 days-old veal sale, whereas the second one aims at purchasing the weaned calf when it is about 5-6 months old, between 180 and 220 kg heavy. In the third breeding type, the young animals, be it males or females, are fattened within the farm, implementing the so-called closed-cycle breeding. Genetic selection largely contributes to calf definition, as, depending on traits, the additive genetic component is more or less heritable. However, during the gestation, postpartum, and weaning phases the contribution is determined also by other factors, partly genetic (i.e., congenital defects), but mainly environmental, attitudinal, nutritional, and healthcare related. Different breeding aspects related to the cow itself and the calf but not associated with genetics influence the production. In particular, the calves rearing represents a very delicate stage among the entire production cycle, both for replacement and slaughter: any mistake committed in this phase affects future production performance. Immediately after birth, the foods ingested are exclusively liquid, i.e., colostrum and milk. The immunoglobulins contained in the colostrum are large protein molecules with an antibody effect that are transferred from the cow to the calf. The absorption by the intestinal mucosa, considerably permeable in the newborn, guarantees the calf protection from environmental pathogens, particularly aggressive towards young cattle. The concentration of antibodies in colostrum reduces over time, and the absorption decreases, disappearing 24-36 hours after birth. The amount of colostrum, between 4 and 6 litres, to be administered in this period and its distribution are relevant. In the case of controlled breastfeeding, the dose should be

administered over two to three meals. One-third of the total quantity is provided with the first meal, within 2-3 hours of birth. In the case of primiparous cows, a reduced or total unavailability of maternal colostrum. To compensate the lack, a reserve of colostrum to draw from can be stored and renewed at least every six months. The provision is usually collected from multiparous mares on the farm, as they are immunized against more pathogens.

The transition from a liquid-based diet to solid food, including also concentrates and fodder, appears later in the diet. After the initial colostral and milky feeding phase, identical for all calves, the weaning period begins and differs considerably among breedings. This stage determines relevant anatomical, histological, and physiological changes in the digestive system of a ruminant. When assuming only liquid food, the ingested milk reaches the glandular stomach directly (abomasum), i.e., the calf acts as a "functional" monogastric, as the pre-stomaches are not mature yet. A subject is weaned when its diet includes exclusively fibrous and concentrate food. The earliness with which a calf is weaned varies according to the type of farming. If the veal is raised with the mother, free to suckle, weaning ends when milk production stops. If the calf stays with the mother temporarily, i.e., in certain moments under the breeder's control, weaning can end earlier, regardless of breast milk production. In this case, it is up to the farmer to decide when to suspend liquid feeding, earlier or later, considering, however, that feeding with mother's milk can not continue later than six months: the quantity of secreted milk is very small and the calf is not easy to manage afterwards.

In the past, the Piemontese breeders claimed the young calf's inability to use foods other than mother's milk in the first 2-3 months of life. The weaning calf has then always traditionally been late. Over the last few years, as for other intensively reared breeds, earlier weaning techniques have been adopted, to contain the costs of labour, feeding and, in general, episodes of enteric disorders. Intentional ingestion of solid food begins at 2-3 weeks of age, favoured by the coexistence with other older individuals. To stimulate the intake, highly palatable and qualitatively flawless products should be supplied, remembering to always remove leftovers, to avoid the onset of abnormal fermentations that reduce the palatability of the available foods. Early ingestion of solid foods supports the rapid growth of the rumen and its physiological microbiological activation. The development of the ruminal papillae, responsible for the absorption of the products derived from fermentation occurring inside it, is physiologically stimulated by the starchy substances fermentation. It is necessary to start weaning with cereals, as rumen development can then absorb also products derived from fodder fermentation. Weaning begun with only hay entails a morphological development of rumen, but not physiological, delaying the ability to absorb the nutrients. An early ruminal activation also means protection from some pathogenic germs of the digestive system, against which the ruminal microflora exerts a strong competition. A suitable concentrate for weaning must contain adequate and balanced protein, energy supplies, and vitamin-mineral supplements. The hay should be neither too fibrous nor excessively leafy, to

maintain palatability and mechanical stimulus. Unbalanced intakes can compromise the adequate skeletal, muscular, and physiological development of the calf. Suitable feed for weaning can be composed exclusively of raw materials, such as cereals added to protein flours, like soybeans, or of raw materials suitably mixed with complementary feed for weaning.

3.2 How is the Breeding Farm Performance Modeled?

Among the several statistical analyses, a detailed farm monitoring is available on ANABORAPI's website, where the registered breeder can check and examine the breeding performance. Indeed, summary data are provided, partly representing the trend developed during the 365 days previous to the consultation day and in part depicting the trend of certain variables among several past years. The tool allows the farmer to measure the effectiveness of the management strategies. As shown in Figure 3.1, many details highlighting the progress are proposed: from the top left to right, some general information about consistencies, primiparous and pluriparous deliveries,and percentages of use of Artificial Insemination are accessible; the trend of the average inbreeding among the farm over the last seven years ensues the average genetic value of the animals on the farm, highlighting the breeding goal. Finally, reproductive efficiency data are provided, transposed and explained with corresponding economic values, in order to compare the gross revenue with the losses due to mortality.

The main information that represents the yield of a Piemontese cattle farm is given by the count of calves per cow per year [10, 18]. The quantity is modelled as follows:

$$Y_p = \frac{365}{intp}\Big(1 - \frac{m}{100}\Big) \tag{3.1}$$

The model estimates the number of calves born alive produced per cow per year. It is a classic statistical model, formulated based on zootechnical hypotheses, and it incorporates two variables extracted from the information of the single farm: the average *calving interval* (intp), that is the time span measured in days between a birth and the previous one, and the average calves *mortality at birth*, i.e., perinatal mortality. In particular, in order to forecast an estimation for the following year, the same model is applied, taking into account the calving interval calculated among the currently pregnant cows instead. Calf mortality is an important cause of economic damages in Piemontese cattle farms: for the farmer, it represents the loss of the economic value of the calf, and the reduction of both the herd's genetic potential and the size of the breeding. It should be noted, however, that breeding gains and losses are not exclusively related with the calving, but are often deeply influenced by the calf development after the first 24 hours following birth. Clearly the gestational phase alone is not exhaustive.

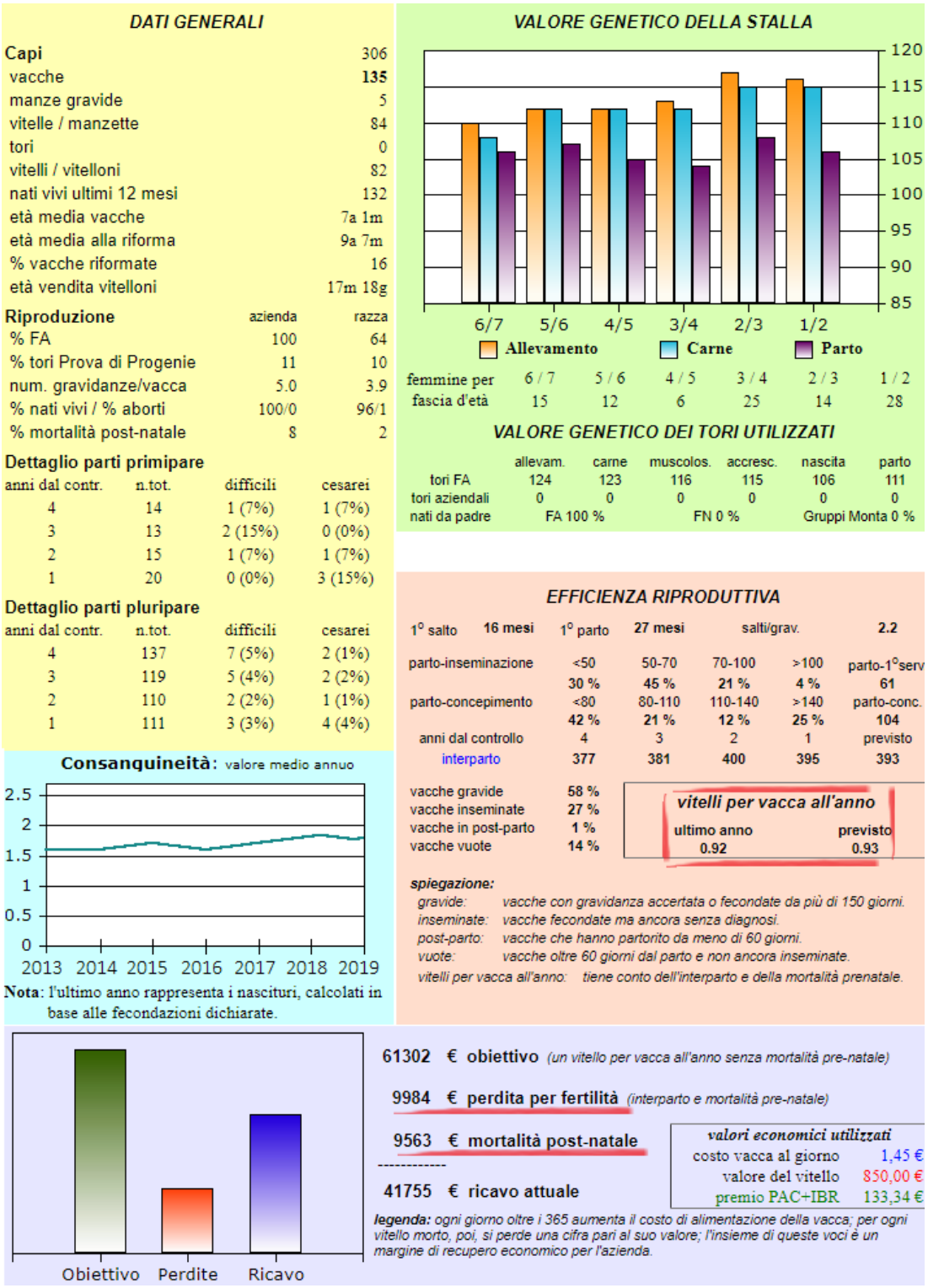


Figure 3.1: Example of a summary breeding report. Top-left section: current consistencies based on most recent data upload. Central-right section, i.e., reproductive efficiency: the alive calves per cow produced in the last 365 days and those predicted for the following year, both obtained with (3.1), are highlighted in red. On the left side, the calving interval used is the overall calving interval recorded among the farm during the last year; on the right side it is calculated upon the currently pregnant cows. Furthermore, in the final section, revenues and losses concerning the economic objective are shown.

It is crucial to consider neonatal mortality, outlining the calf's ability to survive, and the source of possible stress, induced for example by congenital calf's defects (i.e., arthrogryposis and macroglossia [39, 47, 48]). Besides, food and environmental conditions can eventually compromise the immune response and the growth rate [51, 63, 72]. The mortality parameter refers to on-farm deaths, thus excluding any other type of death, such as slaughter, referred to as culling. It is necessary to focus on important aspects, as hygienic-sanitary standards that can reduce the risk of respiratory diseases and compromising enteritis contraction, the need for immediate colostrum administration, the presence of fresh straw in the stall, and good ventilation.

In general, it should be considered that the variables involved in performance evaluation are many, often complex to identify. Indicator 3.1 includes a couple of parameters, neglecting the traits describing the weaning period.

Among the farms considered in the study (description in Chapters 5 and 6), we compared the reported number of calves that died at birth and by the sixtieth day after birth. As it is straightforward to notice from Figure 3.2, during birth almost all farms did not report any deaths, while at the end of weaning the number of farms with zero deaths dropped drastically.

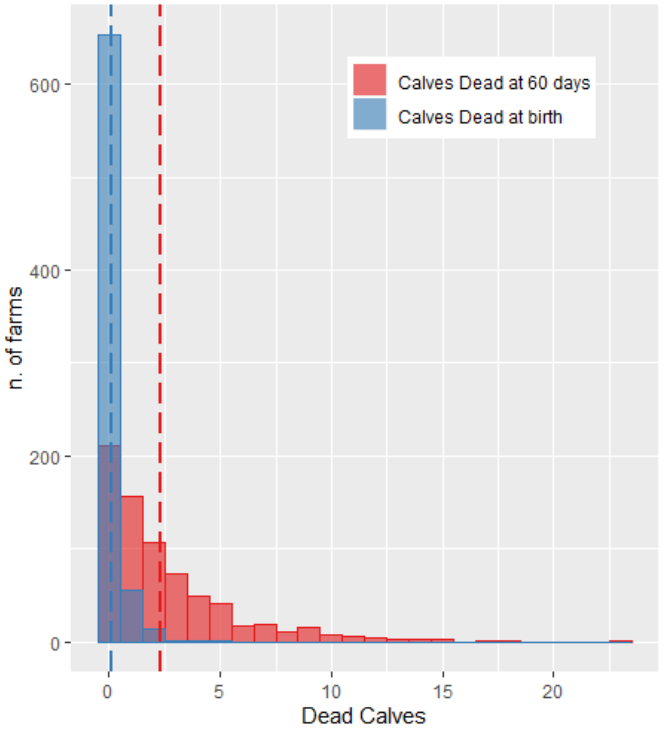


Figure 3.2: Distribution of the number of dead calves at birth and during the weaning period in 2017. Mean values are represented with the dashed line at the two different time reference. The data derive from the dataset described in Chapters 5 and 6. All the breedings (725) show extremely different values between the dead calves at birth (in blue) and (in red) 60 days after it (Kruskal-Wallis test: p-value « 0.001).

It is discernible that the breeders reported a large number of dead calves at 60 days. For the farmer, the loss of the calf means the loss of economic value. The high mortality rate then also reduces then the number of young animals to be used to increase the farm size and the genetic potential of the herd. This makes it necessary to formulate a model predicting the number of calves weaned per cow per year, based on data. Similarly to equation 3.1, it should incorporate the influential variables affecting the output and, at the same time, provide a simply interpretable expression, in order to be able to understand and explain zootechnically the link with the output afterwards.

3.3 Towards the Search of a Prediction Model for the Farm Performance

To describe the breeding performance, it would be more appropriate to obtain an estimate of the number of weaned calves produced per cow in a year. However, an estimate obtained with a classical statistical model such as 3.1 is not predictive: it is a model formulated on a priori knowledge in the field, statically receiving in input the annual data and returning the estimate for that same year as output. An estimate for the immediate future should be provided, to supply the farmer with a more informative indicator, i.e., a suitable tool to evaluate what to expect as a result of the choices made, and how much they are influential. Indeed, the latter are mostly evident over a few years, showing their effects with a certain delay from their introduction. The identification of influential variables within big databases is extremely difficult. Given the huge size of databases, recognizing many of the substantial factors and being able to hypothesize a prediction model can become a complex task. To investigate the production of Piemontese calves and its modelling, it is necessary to examine which variables available in the dataset impact the performance of a breeding farm. Besides, a priori assumptions about data or the relationship between the response and independent variables should be limited only to the preparation of the dataset and the analyses of produced models, i.e., an a posteriori interpretation.

The issue consists of handling an enormous amount of data, without imposing substantial hypotheses on the model to be extracted, which must be predictive. A large amount of data is nowadays collected in the livestock sector, even through the use of sensors, ear tags, collars, images and video recordings in many fields [14, 16, 23, 50]. It is increasingly common to monitor animals, for greater accuracy on the quantity and quality of information, to achieve the economic and environmental sustainability of farms. The breeder must generally deal with animals' problems, like their health conditions and social behaviour, that affect the quality of the product, the life of the animal, and the performance of the farm. The management of livestock by continuous automated real-time monitoring of production, reproduction, health and welfare of the herd, and its environmental impact is defined as Precision Livestock Farming (**PLF**) [15, 16]. PLF supplements

the skills of the farmer, the veterinarian, and the technician by a continuous collection of livestock information, with the support of data transfer technologies [35]. It can play a crucial role in the early detection of diseases and it objectively assesses animal condition and welfare in modern livestock production, representing a tool that supports many farmers as decision-makers. Despite the biological process being too complex to replace farmers by technology, it still offers more possibilities to save money and to change farmers' lives, as a more accurate management system can be achieved, leading to a better approach to the genetic potential of today's livestock species [14, 49, 58].

The creation of prediction models on a specific result in the zootechnical field is increasingly addressed with the use of Machine Learning (**ML**) techniques [24, 49]. ML is a subfield of Artificial Intelligence (**AI**), originally intended to mimic human intelligence [55], addressed to the study of algorithms for prediction and inference [79]. Learning from data is at the core of ML, and hence this field of research is suitable for the management of large data sets, without assuming specific hypotheses among data [30]. As in animal science a growing amount of data is being gathered, ML is used to predict livestock issues, such as time of disease events, risk factors for health conditions, failure to complete a production cycle, as well as the genome of complex traits [32, 58]. Studies have been conducted, based on the application of ML techniques, to model the individual intake of cow feed [80] optimizing health and fertility, to predict the rumen fermentation pattern from milk fatty acids [24] which influence the quantity and composition of the milk produced but also the sensorial and technological characteristics of the meat. The use of ML techniques is also often exploited to identify potential disease predictors, e.g. Bovine Viral Diarrhoea Virus (BVDV), Infectious Bovines Rhinotracheitis (IBR), Bovine Tuberculosis (TB), lameness, and mastitis [28, 34, 43, 68, 77]. Beside these, successful results were achieved for traits prediction as methane production and milk production [31, 81], as well as predicting individual survival [75], classifying grazing and social behaviour [8, 52, 59], and predicting carcass conformation [9], an important component of price negotiations between beef producers and market operators. These works are mostly carried out on dairy cattle, which are more critical to manage from a health point of view. Dairy and beef cattle generally have a different average lifespan, as they are intended for different production. Clearly, the difference between the two types of cattle is determined by distinct production purposes [17, 60]. Dairy cows are thin and lean. Their angular shape can make them look underfed, but it is just their build, like marathon runners. They are milked for about 300 days over the year, two to three times a day, and then they take a break or rest for two months to restore body conditions for another calving. They are also fed a balanced diet, but their energy goes into producing milk, rather than producing mass, i.e., building muscle and storing fat. The two characteristics are indeed usually mutually exclusive. As bulls can not produce milk, they are sometimes used in beef production, albeit with lower yields than specialized beef cattle. In contrast to dairy herd, beef cattle can be compared to weight lifters. They are

characterized by rounded stocky bodies with muscular shoulders and rumps, short necks, and thick backs. Bullocks and steers are intended to produce lean meat with marbling for texture and flavour. Their energy is invested toward building muscle. Beef cows produce milk, but only a little more than enough to feed their calves. In particular for Piemontese cattle, heifers and cows are mainly raised to produce calves, but they are also fattened to produce meat. Consequently, different issues arise among their lifespan and health conditions. In dairy, diseases and metabolic problems affect the cows, and they only survive on average up to three or four lactations [25]. Exceeding this average is rather a feat. For a better performance and a higher yield, they are hence usually crossed with beef cattle [37, 64] and above all a wide range of devices is available to expressively monitor the delicate health conditions.
In beef cattle, the use of devices is still moderate. Mostly concentrated in the Italian region of Piedmont, the Piemontese is a beef cattle raised in intensive breedings, mainly because available pastures are not sufficient for the total number of animals [18, 66]. Furthermore, lifetime is quite short among fattening calves, heifers, and steers, as the animals are slaughtered as soon as they reach the necessary characteristics. Cows and heifers destined for reproduction require instead to be monitored, as raised to give birth several times over a longer lifespan. Consequently, in order to optimize their management, it is necessary to constantly monitor the animals, introducing and adapting to beef cattle the necessary tools implemented for the dairy sector. Moreover, the breeding cycle is reduced compared to other income-producing species, and there is no daily movement of the animals (e.g. milking). The aspects of greatest interest are the composition of the ration and the consumption of food and water, behavioral remarks, the quality of the structures that host the cattle (temperatures, humidity, lighting), growth rate, slaughter yield, and carcass quality. The lower impact of critical points onto the meat sector entails that the adoption of sensors, not yet specific for this type of animal and with a high cost, is probably not worth the economic investment.

As illustrated throughout this chapter, for an optimal farm management, besides the contemporary situation, it is relevant for the breeder to predict future trend. Precisely, the proper parameter to model Piemontese breeding performance is the number of viable calves after the weaning phase, which each cow will produce over the next 365 days. The conditions that permit the survival of the calf are related with its genetic characters and those of its ancestors. However, the calf itself and its aptitude are in large part accountable for good or bad performance, as well as the farmer's managerial choices, and hence the multitude of environmental conditions. In order to investigate the production of Piemontese calves and its modelling, the goal, transversal to all investigations performed during the project, was the identification of the variables influencing the performance of a breeding. Since the goal is to build a predictive model, based on hidden relationships intrinsic to the dataset and possibly readable a posteriori, ML techniques were applied. In con-

trast with previous studies conducted by ANABORAPI, in which models are based on traditional statistical identification approaches, a priori assumptions on the relationship between the response and independent variables were not imposed to construct the best patterns. The appropriate features to predict the defined target were selected among the variables listed in Table 5.1. After the data were filtered, a general simple scheme was pursued in all the stages defined along the study. The dataset was divided into learning and test sets, according to the different study cases. The instances were processed by the different chosen algorithms, to learn relations and to find the hidden patterns between the input variables (i.e., consistencies, CI, mortality, EBVs, consanguinity, etc.) and the specific output variable (i.e., the number of calves weaned per cow per year). The effectiveness of the implemented models was finally assessed. To do so, a test set was used, as its purpose is to determine the validity of the models when it is applied to unseen instances .

The choice of the most appropriate ML approach depends on the goal to be achieved. There are many methods that can produce excellent results by building accurate prediction models [36]. What differentiate them are the distinct algorithms structures and the characteristics intrinsic to the techniques, addressing more or less properly the issue. A simple classical technique as linear regression is often chosen to model the data. However, it can not be the best choice to catch the non-linearity of data. Beside this, a wide range of ML methods is available, properly designed to exploit the underlying non-linearity. If animal husbandry data, particularly in the dairy cattle sector, are widely addressed with simple linear regression and the application of black-box ML methods, the same cannot be affirmed about the use of Evolutionary Algorithms (**EA**), a family of population-based algorithms, mimicking the process of natural evolution (Chapter 4). Such techniques are seldom applied, especially considering the beef cattle sector, regarding which the literature review did not produce relevant results for EAs exploitation. In this regard, this project results in being quite innovative, considering that the adopted approach, based on a particular category of EA, i.e., Genetic Programming (**GP**), for the first time exploited to deal with the Piemontese cattle. Models arising from GP are resumed as intelligible expressions [3]. The produced models can be completely accessed and analytically investigated, qualifying the technique as a white-box method. The algorithm can automatically create feature selecting models, i.e., composed with the most influential variables, delivering accurate results through clear and intelligible expressions. More details are given in Section 4.4. On the contrary, when the internal logic is not accessible, the methods fall into the black-box category. Black-box models generally outperform the others, since their structure is able to capture the high non-linearity underlying data. However, as their definition suggests, the internal processes can be very unclear and do not provide the pursued logic and mechanisms between input and output variables leading to the results. Differently from GP, these methods do not carry out an automatic feature selection, as they encapsulate all the input features within the final model.

Chapter 4

Introduction to Machine Learning

4.1 What is Machine Learning Purpose

Although the terms *Artificial Intelligence* and *Machine Learning* are nowadays "the talk of the town", to the point of making them appear very innovative concepts, their roots lie around the mid-1900s. In 1936 Alan Turing firstly provided a mathematical description of a very simple device capable of arbitrary computations. He proposed an abstract computing machine, provided with a scanner reading and writing symbols while moving along a limitless memory, guided by a program made up of instructions stored in the memory itself. The basic concept behind the abstract machine lies in its ability to operate by modifing and improving its own program. Considered the founder of AI, with his *'Turing Machine'* he became highly influential in the development of theoretical computer science, providing a formalization of computation. Turing indeed supplied the general properties of computation that are at the core of modern computer programming [73]. Turing stated later that computers would one day play very good chess, laying the foundations of AI as a broad concept according to which machines can perform certain tasks intelligently, emulating human decision-making behaviour. In [74], Turing introduced this concept based on the *imitation game*, also known as *Turing test*, wondering about a machine's ability to behave intelligently, taking decisions that are indistinguishable from those of a human. Originally, Turing hypothesized a game involving three players, one of them (the interrogator) was unable to see the others, a man and a woman. By asking questions to the two hidden players by means of notes, he tries to guess their sex. The man's role is to assist the interrogator in making the right decision, whereas the woman attempts to trick the interrogator into making the wrong decision. In the 1950 paper, the only difference introduced in the game was the woman's role, performed by a computer. Provided with an appropriate program, the computer is supposed to play the part imitating the human in the game, as the interrogator attempts to guess who is the human being and who is the computer. Based on these fundamental principles, many efforts were made by different scientists to create successful programs, capable

of replicating in some way human behaviour, trying to incorporate learning instructions. The earliest AI programs capable of playing draughts searching the space were written by Christopher Strachey [71] and Arthur Samuel between 1951 and 1952. In the following years, Samuel improved his work, giving the main definition of ML: *"A computer can be programmed so that it will learn to play a better checkers game than can be played by the person who wrote the program. [...]. Programming computers to learn from experience should eventually eliminate the need for much of this detailed programming effort."* [65]. These two concepts expressed by Samuel condense ML definition into a *"Field of study that gives computers the ability to learn without being explicitly programmed"*. In particular, Samuel dealt with the short available memory supplied with the computers at that time. Indeed he provided the program with a function able to analyze the position of the drought in every moment of the game, calculating then the chances of victory for each side and acting accordingly. Further developments conducted the program towards the progressive incorporation of past experience and "memory" of already encountered positions. Samuel's program was refined over time, becoming such a powerful learner after playing hundreds of times, even against itself, that it was able to challenge medium and high-level professional amateurs. ML is a category of computer programs based on the idea that machines learn to perform specific tasks without being programmed to do so, thanks to the recognition of patterns in data, thus settling as a branch of AI. Algorithms are implemented in order to learn from data and even unknown information is recognized, without explicitly indicating where to look for it. Computers are able to learn by processing data, producing results that are reliable and possibly replicable. One of the most relevant aspects of the process lies indeed in the repetition of instances, as the more the models are exposed to the data, the better they are able to adapt. Thanks to new processing technologies, ML has undergone a significant evolution. This science is not new, but has been developed more recently in order to hone the skills in applying complex mathematical calculations to bigger data. The renewed interest in ML is due to certain factors, namely the growth in the volume and variety of data, as well as the cheaper and more powerful processors. Nowadays, it is possible to automatically build models for analyzing larger and more complex data, and to quickly produce more accurate results even on a large scale. Building precise models entails the identification of new profit opportunities, as well as to avoid unforeseen risks.

The basic premise of ML is to build programs relying on input data to predict an output of interest, handling statistical analysis, and updating outputs as a new input becomes available. The algorithm is selected and set by a human. Thereafter, the computer program learns from data about the underlying relationships, fitting data with mathematical models for making predictions. A dataset is used to train the mathematical model so that it knows what to do when dealing with similar data. ML is most simply the application of statistical models to data using computers. Techniques are constantly evolving, improving structurally,

to handle more complex datasets and hence making fewer assumptions about the underlying data. Recent progress in ML attained an exceptional level of information extraction and semantic understanding, as the ability to detect abstract patterns entails greater accuracy than human experts. A wide variety of ML algorithms is nowadays handled. The characteristics of the data and the type of the desired outcome determine the choice of a model instead of another. The required skill lies in the machine's generalization ability, that is the ability to accurately accomplish new tasks, never tackled before. To develop this skill, it must first gain experience on a set of learning data. The training examples come from a probability distribution, which is generally unknown, but representative of the occurrence space. The machine builds a general probabilistic model of the space of occurrences, *learning* how to recognize cases, and thereafter producing sufficiently accurate predictions among new cases, i.e., *test* instances (Figure 4.1).

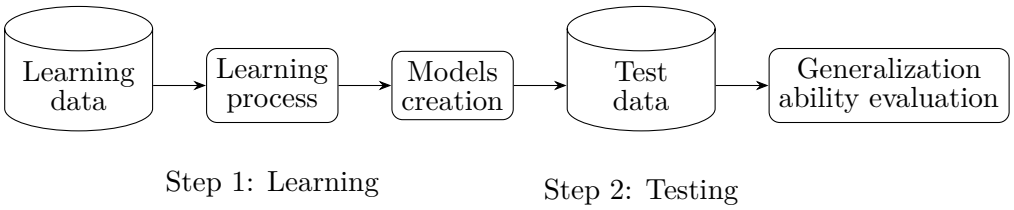


Figure 4.1: ML process flowchart

Depending on the data and the task, machines learn in different ways with various amount of supervision [44]. For this reason, there are several sub-categories of learning, i.e supervised, unsupervised, and semi-supervised. In unsupervised learning, the machine learns from data for which the outcomes are not known, while only input samples are given. However, machines often learn from sample data providing both an example input and an example output. This class of ML is called supervised learning and it is the most deployed form of ML, since the desired predicted outcome is given. In semi-supervised learning, also referred to as weak learning, the two previous categories are combined. As the dataset is not fully provided with output samples, a model uses unlabeled data to gain insight into the data structure, then handles labelled data to learn how to organize the whole information. A further aspect of supervised learning is the type of task to perform [44]. The definition of what group a given input belongs to is the goal of a classification problem, for instance the determination of the presence or absence of a disease, or the categorization of animal pictures into groups. Outputs for classification are typically discrete. Continuous output variables characterize regression predictive modeling instead: the algorithms attempt to estimate the mapping function to continuous numerical quantities, i.e., a size or an amount.

4.2 Preliminary Settings for Good Prediction Models

One of the common key-problems in supervised ML tasks is the phenomenon of over- and under-fitting [40]. In the first case, a model fits the data so well that even the noise is memorized while learning, whereas its performance drops when the model is tested on unknown instances, as the underlying relationships were not detected. The overfitting issue leads to the deterioration of generalization properties of the model, whose performance is unreliable. High variance and low bias estimators usually characterize overfitting patterns (Figure 4.2). On the contrary, when the bias is high and the variance is low underfitting effect occurs, i.e., the opposite of overfitting. The model is unable to map data and to capture the variability, with weak predictive power, failing to generalize on both the learning set and the test set (Figure 4.3). Over- and under-fitting are the result of the attempts to use respectively too complex and too simple models, far from being well-balanced. Pursuing a good configuration can be challenging. Good models arise from optimal parameter settings defining the algorithms, a proper measure to evaluate its error, i.e., a fitness function, and representative datasets (Figure 4.4).

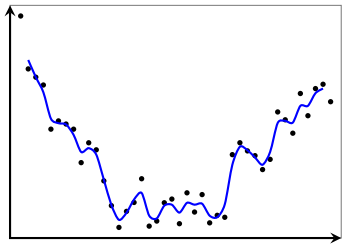


Figure 4.2: Overfitting

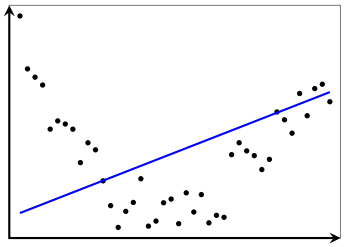


Figure 4.3: Underfitting

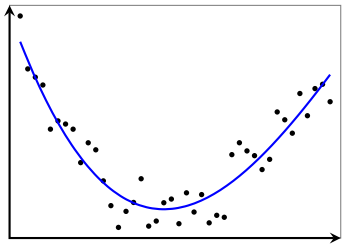


Figure 4.4: Balance

Setting correctly the regularization hyperparameters with which the technique learns and adding complexity to the model are crucial steps. Once the parameters are set and the model is obtained, its fitness is measured, to indicate quantitatively how fit a given solution is in solving the problem. To determine how close the prediction models came to represent the desired solution, they are awarded a score generated by evaluating the fitness function computed on the test. Each problem requires its fitness measure, and hence its proper score. When it comes to formulating a problem, defining the objective function can result as one of the most complex parts, as some requirements should be satisfied. The fitness function should be clearly defined, generating intuitive results. The user should be able to intuitively understand how the fitness score is calculated as well. Besides, it should be efficiently implemented, as it could become the bottleneck of the algorithm. When dealing with a regression problem, the choice usually falls onto the Root Mean Square

Error (RMSE):

$$RMSE = \sqrt{\sum_i \frac{(y_i - \varphi(x_i))^2}{n}}, \tag{4.1}$$

where $i = 1, \ldots, n$ and n is the number of instances (which may differ in the learning and in the test set). The predictor φ is evaluated at X_i, i.e., the input variables values, and Y_i are the target values. A good fitness value means a small RMSE, and viceversa. RMSE is expressed in the response variable's unit and it is an absolute measure of accuracy. The choice of this fitness function is further determined by the application of different ML techniques, that build mostly non-linear models. This issue can exclude a discussion based on the coefficient of determination R^2, as its definition assumes linearly distributed data. When the assumption is violated, R^2 can lead to misleading values [70].

It is common to use a *k random sampling* to split the data into k pairs of training and test sets. The performance is estimated as the average over all k test sets. Any pair of training and test set is disjoint, i.e., does not have any cases in common, whereas any given two training sets or two test sets may overlap. Thus, all the data are considered as learning instances and generalization ability can not be investigated, requiring external additional data. *Cross validation* is a basic approach consisting in extracting a second set from the learning dataset, involving a training set and a validation set. The use of three different sets is advisable to exploit all the available data, as the training set can be used to fit the models, the validation set to the evaluate the predictive performance for model selection, and finally the test set to assess the generalization ability of the final selected model. Thereby, the training and validation sets are considered part of the learning phase, as both participate in the construction of the models. The validation set is used to select the trained model to be tested later. The test gets in the process only at the end of it, to simulate the behaviour of the model on new data, i.e., instances never seen before, and to evaluate its generalization ability. However, it is advisable to dispose of many models to choose from, built with different combinations of the parameters that define the algorithm. Indeed, it can be more or less efficient while learning, depending on the dataset structure. The different values assigned to the parameters require hence to be investigated on different subsets of the learning set. Thus, after the definition of the test set, k random validation sets should be randomly sampled (*k random sampling*), obtaining then k pairs of training-validation sets among the learning set. The key is represented by the multiple training-validation cycles. In each one of the k cycles, a different fold is used for evaluation. In the end, the overall performance of a given model is determined by mean of average scores for each of the folds. An optimization of this technique is *k-fold cross-validation*. It is a better implementation of the k random sampling concept: the learning set is divided into k disjoint folds, i.e., k non-overlapping subsets of equal size (Figure 4.5).

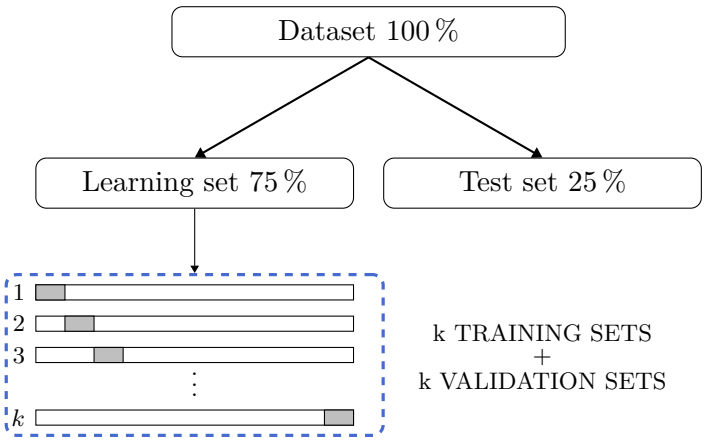


Figure 4.5: Example of k-fold cross validation: for each fold, validation set is highlighted in gray, whereas the remaining data is the training portion

4.3 Adopted Techniques to Model the Farm Performance

As illustrated throughout Chapter 3, starting from the breeding data available in the Herd-Book, the pursued goal consists in investigating the Piemontese zootechnical field, in order to find a prediction model for the performance of the farm. Defined as the number of weaned calves per cow produced in a year, the target is predicted based on the data describing the breeding aspects recorded during a defined year and extracted at a precise moment (within this case study, at the end of 2017 and 2018, according to the specific investigated benchmark problem). Correspondingly, the target is drawn at the end of the following year (2018 and 2019), in order to measure the fitness, as the RMSE between the real value and the predicted one, and minimize it. In the vectorial approach, the input data is extracted at the end of all the years between 2014 and 2017, and the target from 2018. The task belongs to the supervised learning category, as the structure of the available dataset consists in input and output variables. More specifically it is a regression task, since the given output variable (the yearly number of weaned calves per cow) is continuous. As the data are rich in zootechnical meaningful features, a GP approach was adopted (Section 4.4), compared also to classic ML approaches, used for regression tasks. GP was explored through a public user-friendly Matlab implementation of GP, i.e., GPLab [69]. In particular, for what concerns the vectorial approach, a recently introduced version handling vectorial variables representing time series was used [11]. On the other hand, comparisons with standard GP were performed using the R software library "caret" [46], while the vectorial approach was compared with the available deep learning toolbox, implemented in Matlab.

4.3.1 Linear Regression

Linear regression (**LM**) is the simplest form of ML, as the technique consists in a model that assumes a linear relationship between the input and the single output variables, and fits them using ordinary least squares regression, assuming normally distributed error terms in the model. More specifically, given a vector $\mathbf{X}$ of n input features, the representation is a linear function $f(X, \vartheta) = \beta \cdot \mathbf{X} + q$ with parameter set $\vartheta = (\beta, q)$, representing respectively the slope and the intercept of the line. The parameters β and q are estimated during the training process. Input values are combined to produce the expected output as solution by mean of a simple calculation, i.e., linear least-squares, whose result minimises the sum of squares of the difference $(\mathbf{y} - \hat{\mathbf{y}})^2$, where $\mathbf{y}$ is output variable and $\hat{\mathbf{y}}$ the predicted value.

4.3.2 k-Nearest Neighbors

Characterized by a very simple implementation and low computational cost, the k-Nearest Neighbors algorithm (**kNN**) is known as "lazy learning", as it does not build a model, but it is an instance-based method, exploited for both classification and regression tasks. The input consists of the k closest instances (i.e., neighbours) in the features space, and the corresponding output is the most frequent label (classification) or the mean of the output values (regression) of k nearest neighbours. Otherwise stated, in the latter case the k nearest points are computed to predict the value of any new data point, and the values of their output is averaged, to be assigned as the prediction to the given point. The distance used is usually the Euclidean function, defined as:

$$(\mathbf{P}_i, \mathbf{P}_j) = \sqrt{\sum_{\substack{i,j=1 \\ i \neq j}}^{n} (\mathbf{P}_i - \mathbf{P}_j)^2} \tag{4.2}$$

where $\mathbf{P}_i$ are the instances composed of m entries (features), i.e., $\mathbf{P}_i = (x_{i,1}, ..., x_{i,m})$), and n is the total number of instances. However, also other distance metrics are used, depending on the particular dataset [78] The number of k nearest neighbors should be chosen properly, since the predictive power can be strongly affected afterwards. A small value of k leads to overfitting and results can be highly influenced by noise. On the contrary, a large value results in very biased models and can be computationally expensive.

4.3.3 Random Forest

Random forest (**RF**) is an ensemble learning method, which operates by constructing a multitude of decision trees during the learning phase, i.e., a forest. The output results in a class obtained with a majority vote (classification) or the mean prediction (regression) of the individual trees. The decision tree, referred to as a regression tree when the target is continuous, is incrementally developed by splitting the dataset into smaller

and smaller subsets.

The process makes a prediction simply through a sequence of queries about the available data, until a prediction is available. The final result is a tree composed of *decision nodes* and *leaf nodes*. Figure 4.6 shows visually the basic principle of a regression tree. A decision node is labelled by one of the available attributes and splits into two (or more) branches, each representing interval of values for the tested feature. A leaf node represents a decision on the numerical target. The topmost decision node is called the root node.

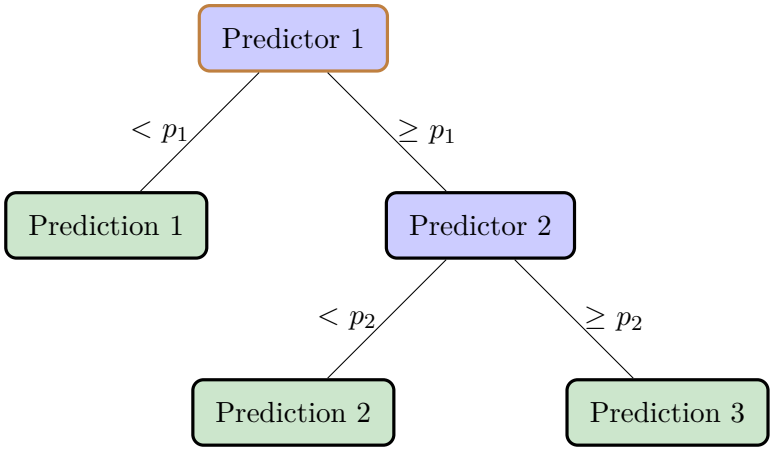


Figure 4.6: A regression tree, i.e., a decision tree whose output is continuous, builds a regression in the form of a tree structure. The scheme shows the basic principle of a regression tree. The decision nodes contain a predictor, and each branch leaving the node represents values for the attribute tested. A leaf node contains the decision on the numerical target. The topmost decision node is called the root node.

Why is it better to create a *forest of decision trees*, instead of a single tree? Decision trees are prone to overfitting. Indeed, when their maximum depth is not limited, they can keep growing until exactly one leaf node for every single observation is produced. Exact prediction outcomes for all instances in the learning set are available, but the trees fail to develop generalization ability. As an alternative, instead of limiting the depth of the tree, which would reduce variance while increasing bias, many decision trees can be combined into a single ensemble model, i.e., a forest, whose output is the average of all the prediction models outcomes. The adjective *random* is given by two key concepts: each tree in the forest learns from a random sample of the data points and only a random subset of all the features is considered for splitting each node in each decision tree. It is possible to grow the trees by splitting nodes at fully randomly chosen cut-points. In this case the algorithm uses the whole learning sample and all the variables are selected at each split. However, a *bootstrapping* is usually first performed by the algorithm to extract the random sample of the learning observations. The samples are drawn with replacement, in order to process samples of the same size. This means that some samples are used multiple times while constructing a single tree. Training every learner

on many samples could produce models with higher variance with respect to a single set. However, as the final output depends on the overall trend among the forest, both bias and variance can be maintained low. The procedure of training each individual learner on different bootstrapped subsets of the data and then averaging the predictions is known as *bagging* (**B**ootstrap **Agg**regat**ing**). Figure 4.7, describes the process to obtain a prediction with RF.

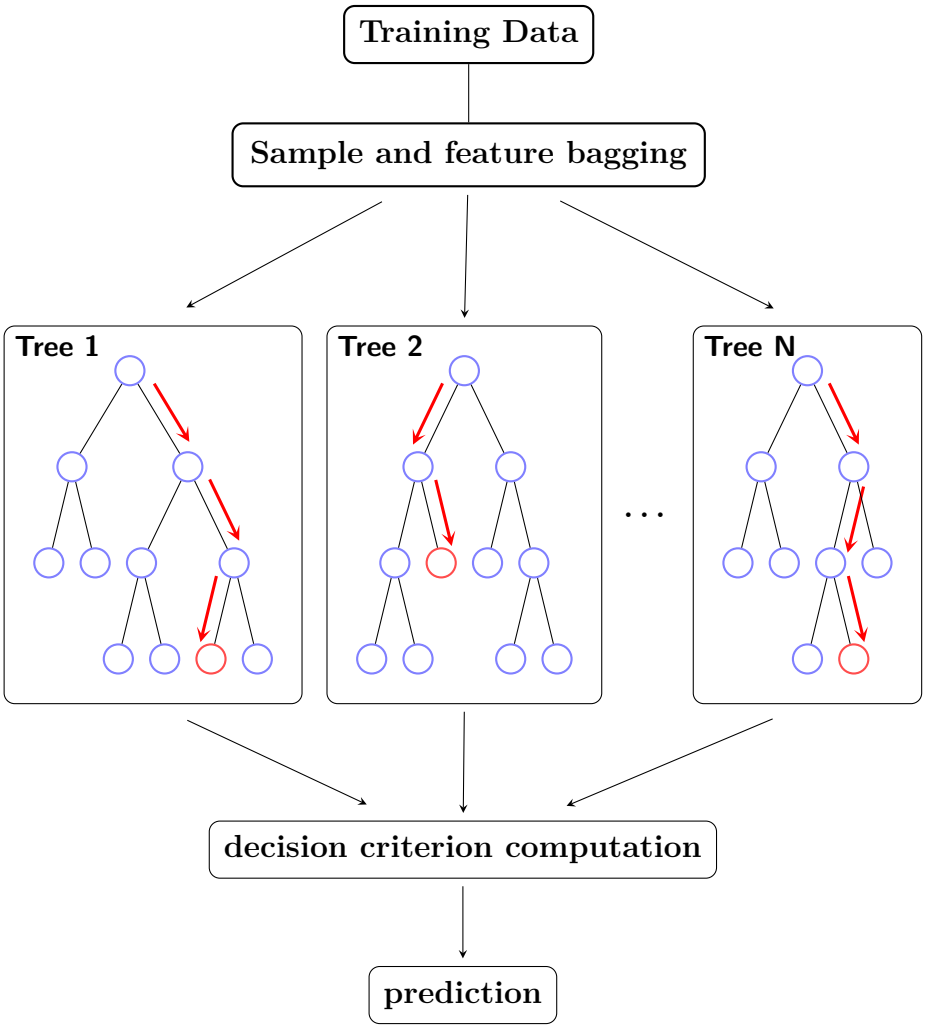


Figure 4.7: Scheme of the Random Forest algorithm process. After the dataset is subsampled, i.e., bootstrapping is performed, a forest of N trees is grown. All the trees contribute to the final prediction, whatever decision criterion is set: whether it is a classification or regression problem, the majority vote or the mean value of all N predictions is the final outcome. In other words, instances to be tested are given in input to all the trees, and the obtained predictions (highlighted in red) are finally .

A consequence of bootstrapping is the possibility to deal with *Out-of-Bag* instances, i.e., data that were not sampled. As the *OOB* dataset was not used to create the trees, the algorithm runs them through and checks

the prediction ability of the models. The OOB error is similar to the cross-validation error, i.e., after many iterations, OOB error stabilizes converging to the cross-validation error. OOB error has the advantage of requiring less computation. However, OOB can overestimate the generalization error to a greater extent than cross-validation [19, 41, 57].

4.3.4 Extreme Gradient Boosting

Gradient Boosting (**GB**) is a technique producing ensembles of models, usually small decision trees. Boosting is a procedure that combines the outputs of different "weak" models to produce a powerful upgraded one. From this perspective boosting recalls a resemblance to the bagging approach used by RF. However, it is fundamentally different. First of all, while RF does not set boundaries on tree dimension, boosting technique usually implements trees composed of few nodes and leaves. In other words, RF grows forests of big trees, whereas boosting algorithms build forests of very small trees, stumps if composed of one node and two leaves. Clearly, stumps and very small trees are not good at making accurate predictions, as they can use only one or few variable at a time, and are indeed called "weak learners". Nonetheless, it is frequently reported that boosting methods outperform RF.

The first algorithm proposed to exploit the boosting problem was introduced among classification tasks by Shapire [67] in 1990, based on the idea that a set of weak learners could form a single strong one. Accordingly to this hypothesis, the algorithm worked by weighting the samples observations at each iteration and by forcing one stump to adapt to the incorrectly predicted samples. During the first step, the weak learner is simply trained on the original data. Thereafter, at each iteration, the weights are individually modified and examples that are difficult to predict receive ever-increasing influence. However, the technique could not take full advantage of the weak learners, as the process was not *adaptive*, consisting only of one weak learner, trained to become stronger. It took no time for the technique to be developed as an *Adaptive Boosting* algorithm (AdaBoost), adapted also for regression tasks [26, 29] . Unlike previous algorithms, AdaBoost makes use of many weak learners, usually decision trees, added subsequently. At each iteration, samples are still re-weighted, and weak learners that are added sequentially are forced to concentrate on the examples that are missed by the previous ones in the sequence. The final prediction consists of a median of all the weak learners; more accurate learners contribute with larger weights. AdaBoost is regarded as a special case of *Gradient Boosting*, to which it was later generalized. As a generalization, GB inherits the main characteristic from AdaBoost, that is the idea of building a single strong learner, by training a set of weak learners added sequentially. The trees that GB grows are larger than stumps, but their size is still restricted, usually limited to four leaves. Differently from AdaBoost, that works basically assigning the weights to the samples, GB trains learners based upon minimising the loss function of the learner, by setting

optimal parameters. Instead of receiving weighted samples, at each iteration the algorithm directly fits the added learner on the residuals errors committed by the previous learner. In GB, the order of construction of stumps is crucial: the prediction error committed by the first iteration learner influences the construction of the second iteration learner, and so forth. The final ensemble model takes into consideration the average of all learners, with the possibility of emphasising the most accurate ones with a larger weight (particularly with AdaBoost). On the contrary, RF grows large trees independently and the final prediction ensemble takes into account the average of all the models, with no importance given to the order of construction.
In order to find the parameters used to minimize the loss function, usually convex[1], the algorithm applies *Gradient Descent*. It is a powerful optimization algorithm used in many methods in order to find the loss function minimum. Given an n-dimensional vector X and a set of parameters ϑ, the algorithm operates by computing the gradient[2] of the loss function $f(x, \vartheta)$ with respect to ϑ. The value of the gradient at a point is a tangent vector, that indicates the direction and rate of the increase of the function in that point. When this rate is equal to zero at a given point, i.e., the gradient vector has null values, the function is stationary in that point. Convex functions have a key-role in this point. Indeed, if the function is strictly convex the stationary point is a minimum, and furthermore a convex function has no more than one minimum. Even in infinite-dimensional spaces, under suitable additional hypotheses, convex functions continue to satisfy such a property. The descent is an iterative process, in which, step by step, the parameters are tweaked in the opposite direction of the gradient, i.e., in the direction of steepest descent (steps in the direction of the gradient are performed when the focus is a maximum search). This process is repeated for different points in the space of inputs until a minimum of f is found.

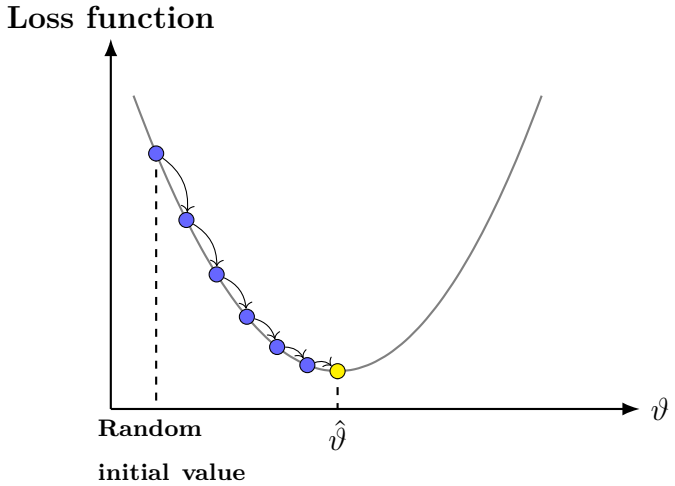


Figure 4.8: Gradient descent is the process of gradually decreasing the loss function by nudging the parameters ϑ iteratively until a minimum is reached. In figure: a representation in a two-dimensional space.

Extreme gradient boosting (**XGBoost**) is one of the most popular variants of gradient boosting [22]. It is a better and more efficient performing implementation of gradient boosted regression trees. Indeed, the process is way faster as parallel computation on a single machine is performed and it can handle also sparse data sets. However, these are not the only "tricks" that make gradient boosting *extreme*. One of the main features of XGBoost is the provision of more regularization options among the loss function (both $L1$ and $L2$ are available[3]), in order to avoid increasing complexity and overfitting, improving the generalization performance. Furthermore, the algorithm computes the second-order gradients, i.e., the second partial derivatives of the loss function. Regular GB uses generally a convex loss function. When descending the gradient, it is possible to take smaller or bigger steps. In the first case, growing the number of steps clearly increases the number of approximations to be computed, even if they lead towards the minimum. In the second case, bigger steps could reduce the number of iterations, but would include the possibility to jump too far in the opposite direction of the gradient, increasing the computational costs. The second-order derivative would provide more information about the direction of gradients and how to get to the local minimum of the loss function, when dealing with any kind of function. Moreover, the principle of gradient descent can be extended to any kind of loss function by computing the second-order derivative. Indeed, dealing with any function, the minima are searched by computing the second order derivative. If a function f that is twice differentiable at a stationary point x_0 has $f''(x_0) > 0$, then f has a local minimum at x_0. After all, a twice-differentiable function is convex if and only if its second derivative is non-negative over its entire domain. XGBoost exploits the second-order derivative, taking more time to compute the direction where to go, in order to take fewer steps to get there and avoiding unnecessary computations.

4.3.5 Neural Networks

A Neural Network (**NN**), usually denoted with the term of Artificial Neural Network (**ANN**) emulates the complex functions of the brain. An ANN is a simplified model of the structure of a biological neural network and consists of interconnected processing units organized according to a specific topology. The behavior

[1]A function $f : I \subseteq \mathbb{R}^n \to \mathbb{R}$ is convex if its domain is a convex set and for all x, y in its domain and all $\lambda \in [0, 1]$: $f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)$. Stated otherwise, a function is convex for all the points of the graph if the line segment between any two points on the graph of the function lies above the graph of the function between the two points.

[2]Given a scalar-valued differentiable function f of several variables, the *gradient* of f is the vector ∇f, whose value at a point p are the partial derivatives of f at p. The ∇ symbol denotes the vector differential operator. Stepping in the direction of the gradient leads to a local maximum (gradient ascent) or minimum (gradient descent) of that function.

[3]The regularization $L1$ and $L2$, also known respectively as *Lasso* (least absolute shrinkage and selection operator) and *Ridge* regularizations, are based on *shrinkage* parameters introduced into the loss function. The two kinds of shrinking penalties are defined based on the considered norms: $L1$ norm is calculated as the sum of the absolute values of a vector, whereas $L2$ norm is calculated as the square root of the sum of the squared vector values.

of the nodes recalls that of biological neurons. A neuron integrates the signals received from various other neurons via synaptic connections. If the resulting activation exceeds a certain threshold, an action potential is generated and is propagated through the axon to one or more neurons. The "neuronal units", i.e., nodes, that compose a NN are arranged in successive layers. Each neuron is connected to all the neurons in the next layer via weighted connections. The connection is nothing more than a numerical value, i.e., a weight, that is multiplied by the value contained in the neuron. All the neurons connected to a next layer node contribute to its output value, by mean of a weighted sum. An activation function, i.e., a (generally non-linear) mathematical transformation, is applied to the result before passing it to the next layer. In this way, the input values are propagated through the network, until the output node is reached. The gist of it consists in regulating weights and bias, to obtain the desired result.

As shown in Figure 4.9, a NN is formed by a set of nodes arranged in at least three layers. The network is fed with features values through an input layer. Thereafter, the learning takes place among one or more hidden layer, composing the internal network. Finally, the network includes an output layer, where the prediction is given. Learning occurs by changing connections weights, based on the error affecting the output. At each update, the weights of the connection between nodes are multiplied by a factor in order to prevent the weights from growing too large and the model from getting too complex.

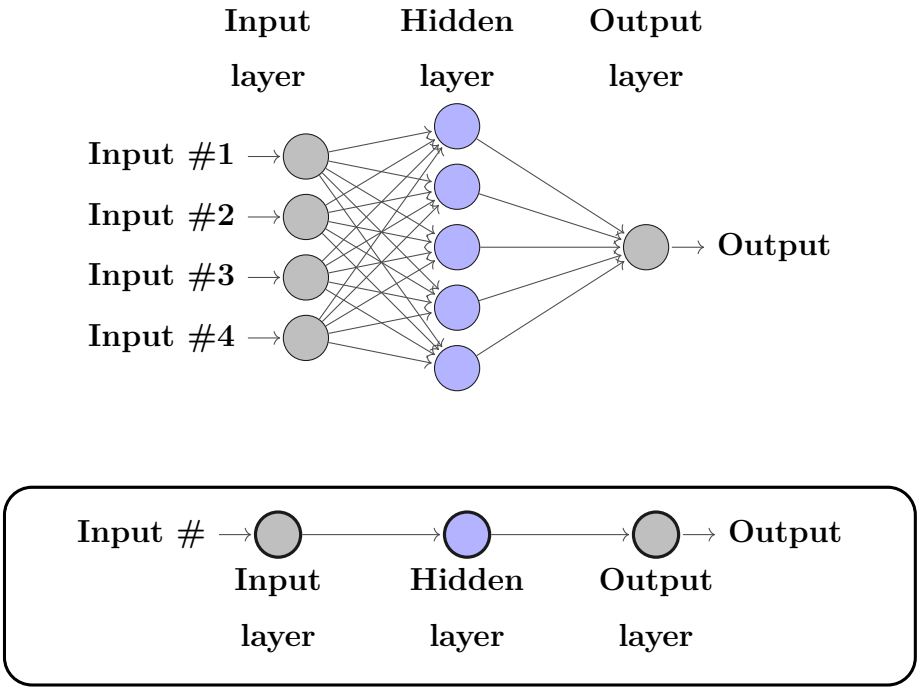


Figure 4.9: Representation of a single hidden layer Artificial Neural Network, with four input features. In the lower part of the figure, the folded representation is depicted.

ANNs are popularly known as universal function approximators, as they are capable of learning any non-linear function. They can learn weights that map any input to the output. One of the main reasons behind this ability is the activation function. Indeed, it introduces non-linear properties to the network, helping the network learn any complex relationship between input and output. Without an activation function, the network can only learn a linear function and can not do complex relationships.
Bias is rather a constant that is added to the linear combination of inputs and weights. It is applied before the activation function, and has the effect of shifting by a constant amount the activation function. In Figure 4.10 the steps pursued to assign the value to each node in the hidden and output layers.

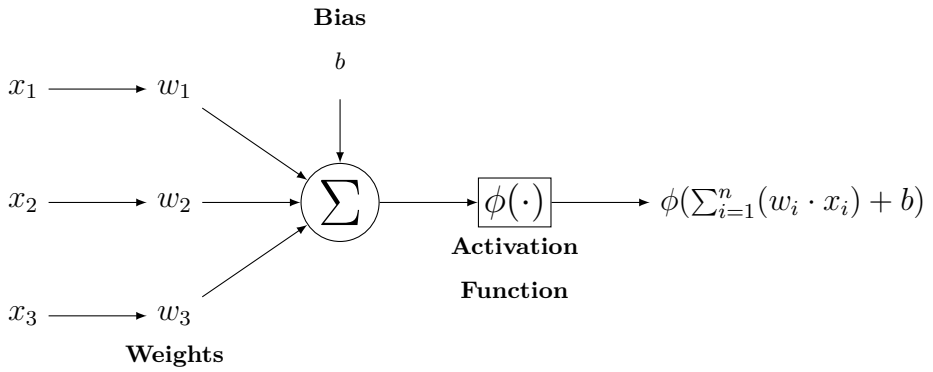


Figure 4.10: Representation of the steps exploited to process information among the single nodes of a Neural Network. A nonlinear activation function ϕ is applied to the weighted sum of the input values and the bias term b in order to compute the value for the next node.

Networks provided with this structure are also known as *Feed-Forward Neural Networks*, as inputs are processed only in the forward direction. In order to minimize the error, the gradient descent algorithm can also be applied to an ANN. The task is performed by through a *backpropagation* algorithm , that works, by computing one layer at a time, the gradient of the loss function with respect to each weight. Iterations are computed backwards starting from the last layer.

One of the disadvantages of an ANN, is that it cannot capture sequential information in the input data. An ANN can deal with fixed-size input data, that is all the item features feed the network at the same time, such that there is no time interval between the data features. Even if there were a time interval between two input data, a basic ANN simply could not detect it. When dealing with sequential data, in which there are strong dependencies between the data features, i.e., in text or speech signals, a basic ANN is not able to address properly the task. In this regard, basic ANNs were developed to make way for a more efficient algorithm, particularly useful for time series. **RNN** is a type of ANN, that has a recurring connection to

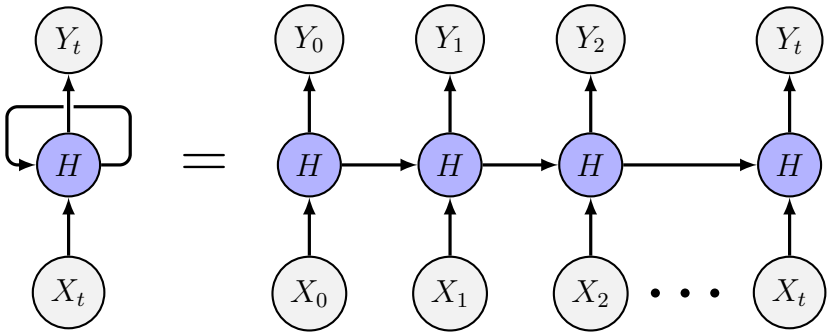


Figure 4.11: Scheme of a RNN and the unfolding in time of the forward computation. On the left side, the neural network is fed with input $X_{(t)}$, and outputs a value $Y_{(t)}$. A loop among the hidden layer H allows information to be passed from one step of the network to the next. A RNN can be seen as multiple copies of the same network, where each copy passes sequentially information to the next one (right side).

itself. While predicting the output y at time t, the algorithm maintains an "internal state" that is used to store the context of the data being fed into the network. This functionality makes it possible to exploit the previous input $x_{(t-1)}$ along with the current input $x_{(t)}$, as the hidden layer activations calculated at time $t-1$ are fed in as an input at time t. This gives RNN a sense of time context.

Depending on the issue, to perform the task it is sometimes sufficient to look at recent information. Sometimes, more context is rather needed. The gap between information may become very large and the amount of sequential information can be complex to retain. As that gap grows, RNNs lose their ability to learn connections. Besides, when the RNNs are trained, the gradient calculation becomes quite a feat, as it is performed throughout many layers, including time. To overcome the short-term memory weakness, Long Short-Term Memory (**LSTM**) architecture was designed to solve this problem with RNNs. By mean of internal mechanisms, they keep track of the dependencies between the input sequences, storing and removing unnecessary information. The LSTM introduces the concept of cell states. By using special neurons called "gates" placed in the cell state, LSTMs can remember or forget information. Three kinds of gates are available inside the cell, in order to filter information from previous inputs (forget gate), to decide what new information to remember (input gate), and to decide which part of the cell state to output (output gate). These gates are a sort of highway for the gradient to flow backwards through time.

4.4 Genetic Programming

Since GP is the technique adopted as a baseline and major investigations are conducted, the entire section is dedicated to its description. GP is a family of population-based Evolutionary Algorithms (**EA**), mimicking the process of natural evolution. In other words, the principles of Darwin's theory of evolution were expressed

algorithmically, in order to evolve models and obtain a strong predictor. Indeed, individuals that represent one candidate solution in the population are phenotypes, characterized by *chromosomes* (or alternatively *genome*), i.e., sets of parameters outlining a proposed solution to the problem to be solved. Finding a suitable representation for a chromosome limits the search space, making the search easier. A poor representation, on the contrary, entails a larger search space. Each characteristic of the individuals' chromosomes is referred to as a *gene*, and all its possible values are referred to as *alleles*. A genetic representation can encode physical qualities of individuals, its appearance, and behavior. Modelling the genotype is a branch of Evolutionary Computation (**EC**), referred to as *Genetic Algorithm* (**GA**). Similarly to GA, GP models the genetic material. Their difference lies in the representation of individuals [45, 62]: while GA processes binary strings, GP accomplishes a tree-based representation. The nodes contain operators, whereas the leaves (terminal nodes) are fed with operands, i.e., the features' values. (Figure 4.12).

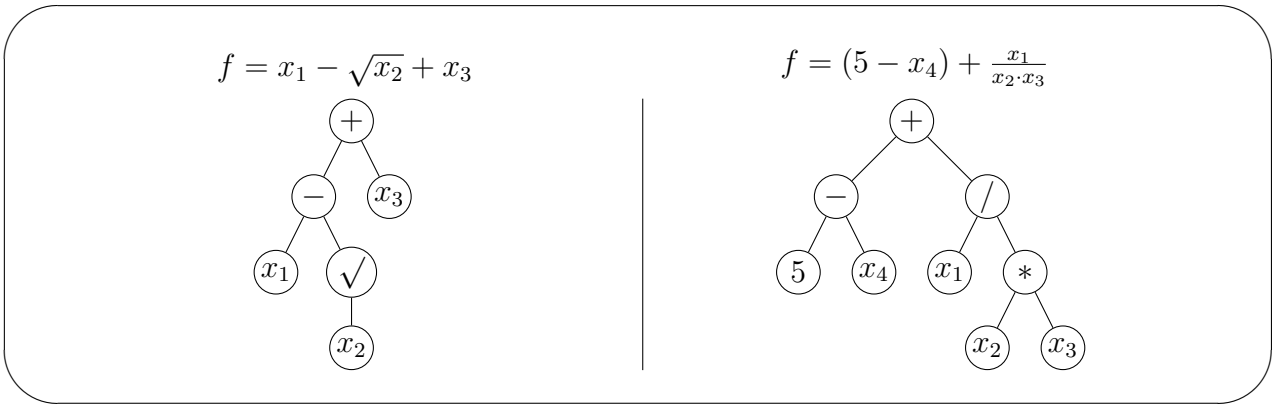


Figure 4.12: GP programs examples: functions are represented as a tree structure, making mathematical expressions easy to evolve and evaluate. Every tree node contains an operator function, whereas every terminal node has an operand.

As in an evolutionary biological process, the initial population evolves through the course of generations, exploiting the mechanisms of selection, mutation, and recombination of individuals. For each generation, individuals compete to reproduce offsprings. Individuals may undergo culling or survive to the next generation. As the individuals showing the best survival capabilities have the best chance to reproduce, they form elites of valuable candidates contributing to the creation of new individuals for the next generation. Offsprings are generated by a *crossover* mechanism, i.e., the recombination of parts of the parents, and by *mutation*, that is the alteration of some of the alleles. The survival strength of an individual is measured using a fitness function. The population is transformed iteratively based on the training set, inside the main generational loop of a GP run. An initial population of individual computer programs is randomly generated (generation

0), consisting in simple trees, composed of the available functions and terminals. If greater complexity is necessary, during the process their size can grow. Thereafter, sub-steps are iteratively performed within each generation, until the termination criterion is satisfied. Termination consists in a limit imposed upon the number of generations, or the number of fitness function evaluations, or even a convergence criterion for the population. At that point, the population is evaluated on the validation set, to pick the best model.

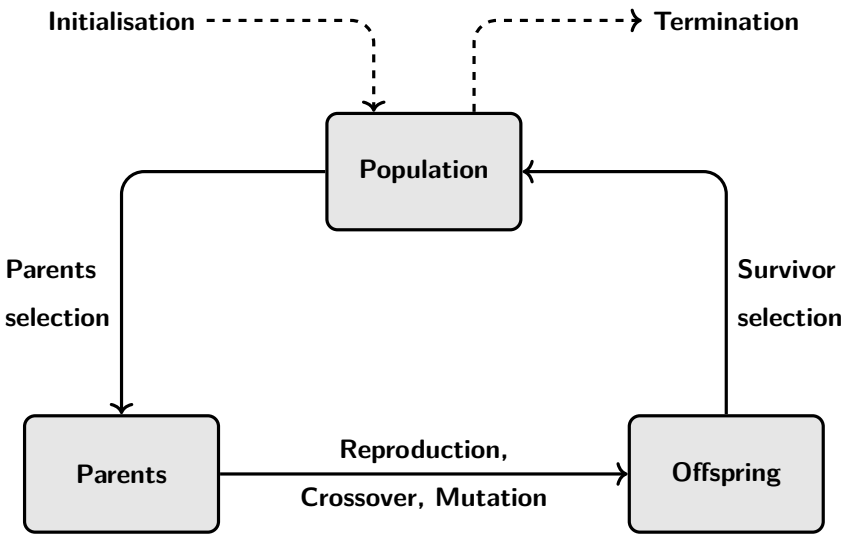


Figure 4.13: A flowchart representing the main generational loop among a run of GP. The run starts with an initial randomly generated population. Parents are selected from the initial population for mating: the genotype is altered to generate new offsprings. Finally these offsprings replace the existing individuals in the population and the process is repeated, until a terminal criterion is satisfied.

At every generation, each program in the population is executed and its fitness ascertained on the training set using the proper fitness measure. One or more individuals are selected to participate in the genetic operations. The selection probability is based on the measured fitness. New programs are created by applying the following genetic operations:

- *reproduction*, that is the copy of the selected individual to the new population,
- *crossover*, consisting in the recombination of randomly chosen parts from two selected programs,
- *mutation*, a random mutation of a randomly chosen part of one selected program.

When a the termination criterion is satisfied, the whole final population is evaluated on the validation set and the single best program produced during the learning phase is designated as the result of the run. If the run is successful, the result may be a solution to the problem. Generalization is finally evaluated on the test dataset.

By selecting, recombining, and mutating the best individuals, at each evolutionary step (i.e., each new generation) the members of the new population are, on average, fitter than the previously generated ones, i.e., they show a smaller error. Trees are built assuming different sizes and shapes. Among the parameters defining the technique, the preservation of the best individual at each run is feasible, and fitness can be treated as the primary objective, whereas tree size is a secondary parameter, when ranking models. This peculiarity leads to the conservation of the most influential variables over generations. The algorithm performs, hence, an implicit feature selection and, among all the input variables, only the most relevant are encapsulated in the solutions.

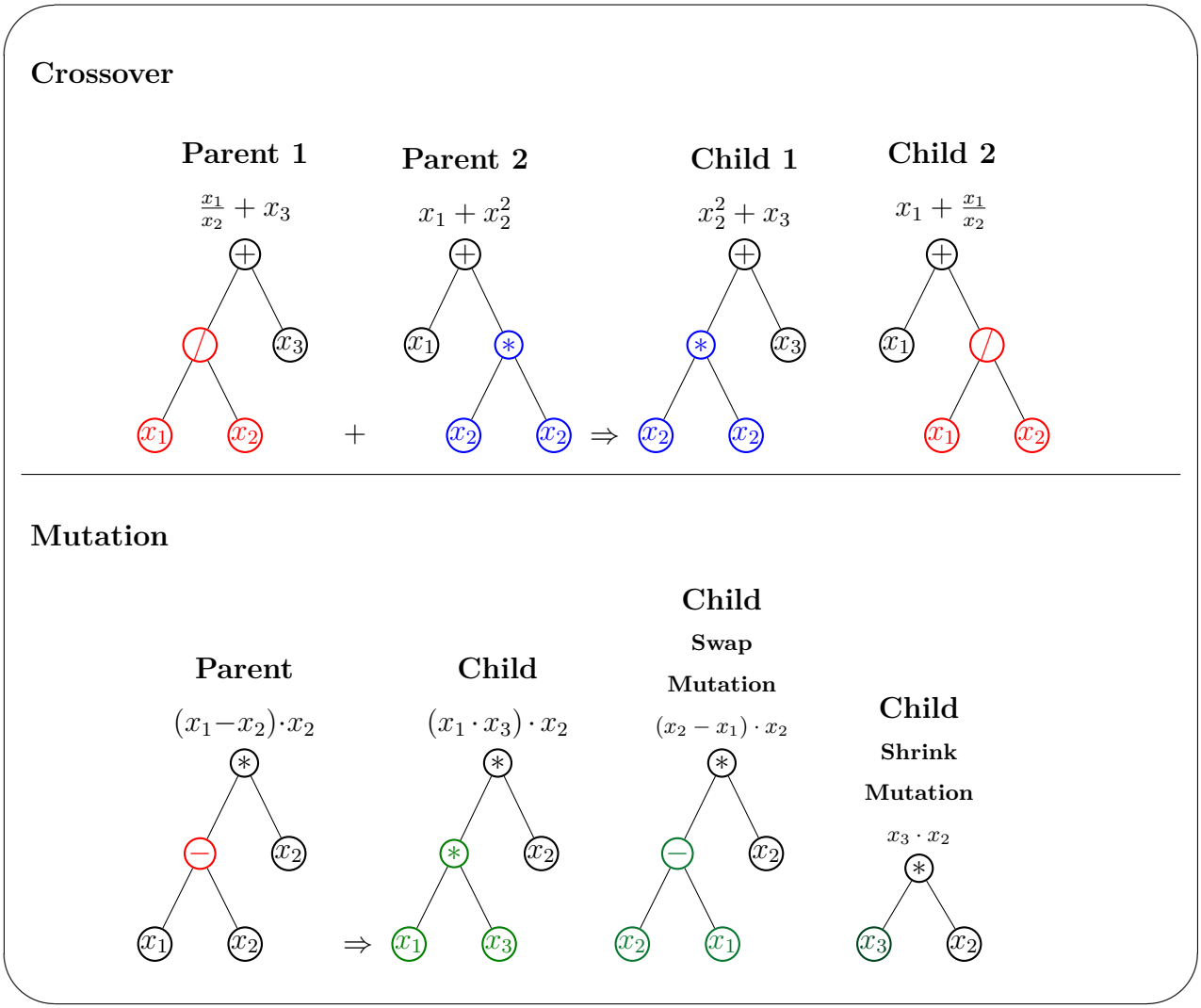


Figure 4.14: Mutation and crossover operations in GP

Standard GP (**ST-GP**) is a powerful algorithm, suitable to perform symbolic regression on any dataset. However, as many other standard techniques do, instances are treated independently. showing a potential

disadvantage when dealing with sequential data. This may result in a loss of knowledge in pattern recognition of the temporal information. Besides RNNs, whose structure is suitable for managing a collection of observations at different equally spaced time intervals, **Ve**ctorial **G**enetic **P**rogramming (**VE-GP**) can manage vectorial variables representing time series [7, 11–13, 38]. Indeed, the development of ST-GP led to techniques exploiting terminals in the form of a vector. With this representation, all the past information associated to an entity is aggregated into a vector, giving a sense of memory and helping keeping track of what happened earlier in the sequential data (see Figure 4.15). VE-GP comes with enhanced characteristics of ST-GP exploiting a proper data representation processed with suitable operators to handle vectors, reinforcing the identification ability of correlations and patterns. The target can be scalar, as well as vectorial. The technique can indeed treat both vectors, even of different lengths, and scalars together, performing both vectorial and element-wise operations.

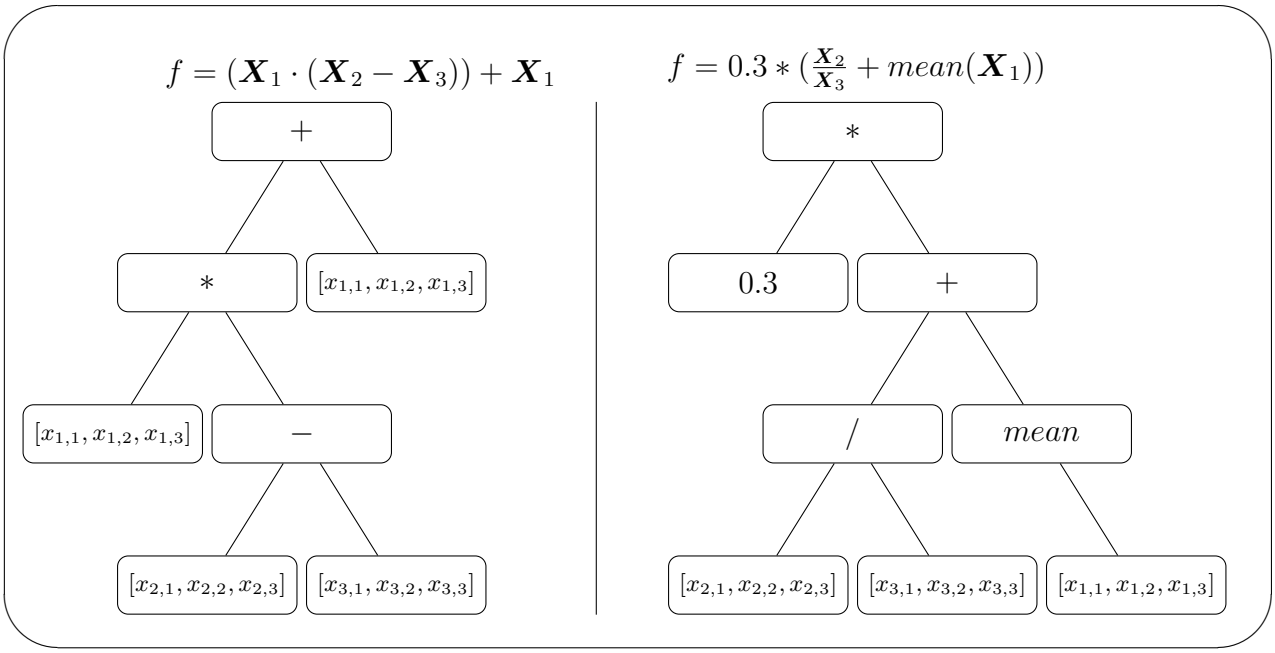


Figure 4.15: VE-GP programs examples: like ST-GP representation, functions are embodied as a tree structure. Terminals are exploited in the form of vector and processed through suitable operators

Chapter 5

The Dataset and Data Preparation

5.1 The Available Data

This chapter describes the available dataset and all its variables, the scheme with which the farms were selected, and the definition of the target variable. As previously mentioned in Chapters 2 and 3, ANABORAPI designed a web service, accessible to registered users, which provides summary breeding reports. The data are entered from PCs and other devices, i.e., smartphones and specific devices, sent in real-time to the servers, stored and processed. The available database provided by ANABORAPI is an event history for all farms registered in the Herd-Book of the Race. For every farm, current data and average statistics recorded by technicians during routine controls, veterinarians, and directly by farmers, are elaborated. There are several records for each farm since the track of every visit are kept. The content of the database is processed by the system on the elaboration date. The average statistical values are calculated over the previous 365 days, starting from the last visit date or last data entry. Statistics are finally provided (Figure 3.1). In addition to ID data of the breeding farms, all information on the consistencies, the deliveries and births, the type of inseminations carried out (natural or artificial), visits dates, Estimated Breeding Values (EBV), Selection Indices, consanguinity of all registered bovines, perinatal mortality rates are kept. Globally, the database contains the last twenty years of data, including farms that are no longer active, for total of 219 descriptive variables reported in Table 5.1. Among them there is the number of calves alive per cow obtained with equation 3.1 (line 102 in Table 5.1).

	Field Name	**Type**	**Width**	**Description**
1	DATA_ELAB*	Date	8	Data elaboration date
2	PROPRIETAR	Char	7	Farm code, i.e., Farmer ID
3	CAPI*	Num	6	Cattle size
4	VACCHE*	Num	6	Consistency for cows, i.e., number of cows

Continued on next page

Table 5.1 – continued from previous page				
	Field Name	**Type**	**Width**	**Description**
5	MANZE*	Num	6	Consistency for heifers, i.e., number of heifers
6	VITELLE*	Num	6	Consistency for heifers, i.e., number of heifers
7	TORI*	Num	4	Consistency for bulls, i.e., number of bulls
8	VITELLI*	Num	6	Consistency for male calves, i.e., number of male calves
9	PUNTEGGIAT*	Num	6	N. of morphologically evaluated animals
10	PUNT_MEDIO	Num	4	Morphological evaluation
11	PERCENT_FA	Num	3	Percentage of Artificial Insemination
12	PERC_PROG*	Num	3	Percentage of calves selected for progeny test
13	SALTI_TOT*	Num	6	N. of total inseminations
14	ETA_VACCHE	Num	5	Cows age expressed in days
15	C_ETA_VAC	Char	7	Cows age expressed in years
16	ORD_PARTO	Num	4	Average parity
17	N_PARTI*	Num	6	Total n. of occurred deliveries
18	PERC_VIVI*	Num	3	Percentage of calves born alive
19	ETA_SALT_1	Num	4	First insemination age
20	ETA_PART_1	Num	4	First calving age
21	INTERPARTO	Num	4	Calving interval in days, based on currently pregnant cows
22	PAR_SALT	Num	4	Avarage interval between calving and insemination
23	PAR_CONCEP	Num	4	Average interval between calving and conception
24	SALXGRAV	Num	4	Insemination order on pregnant cows
25	N_SALT_1*	Num	6	N. of heifers at the first insemination
26	N_PART_1*	Num	6	N. of primiparous
27	N_INTERPAR*	Num	6	N. of bovines on which INTERPARTO is calculated, i.e., n. of cows with at least 2 deliveries currently pregnant
28	N_PAR_SALT*	Num	6	N. of cows with a calving before insemination (as last event)
29	N_PAR_CONC*	Num	6	N. of cows with a calving before impregnation (as last event)
30	N_SALXGRAV*	Num	6	N. of pregnant cows
31	ETA_RIFORM*	Num	5	Cows age in days at cull
32	C_ETA_RIF*	Char	7	Cows age in years at cull
33	PERC_RIFOR	Num	3	Percentage of culled cows
34	PRIM_RIFOR	Num	3	Percentage of culled primiparous
35	CORRETTI	Num	3	Percentage of calves born without defects(e.g. Macroglossia, Arthrogryposis)
36	COGNOME	Char	56	Farmer's surname
37	COMUNE	Char	25	Municipality
38	COD_CONTR	Char	5	Visiting technician ID code
39	QUALIFICA	Char	35	

Continued on next page

Table 5.1 – continued from previous page				
	Field Name	**Type**	**Width**	**Description**
40	VIA	Char	30	Address
41	FRAZIONE	Char	30	Borough, village
42	FAC_MANZE	Num	3	N. of primiparae that delivered with easy calving
43	DIF_MANZE	Num	3	N. of primiparae that delivered with difficult calving
44	TAG_MANZE	Num	3	N. of primiparae that delivered through a caesarean section
45	FAC_VACCHE	Num	3	N. of cows that delivered with easy calving
46	DIF_VACCHE	Num	3	N. of cows that delivered with difficult calving
47	TAG_VACCHE	Num	3	N. of cows that delivered through a caesarean section
48	VACC_N_IND	Num	3	Birth ease (EBV for cows)
49	VACC_PARTO	Num	3	Calving ease (EBV for cows)
50	VACC_ACCR	Num	3	Growing (EBV for cows)
51	VACC_MUSC	Num	3	Muscularity (EBV for cows)
52	VACC_CARNE	Num	3	Meat Index (cows)
53	VACC_ALLEV	Num	3	Breeding Index (cows)
54	CONS5	Num	5	Consanguinity during the 5th previous year
55	CONS4	Num	5	Consanguinity during the 4th previous year
56	CONS3	Num	5	Consanguinity during the 3rd previous year
57	CONS2	Num	5	Consanguinity during the 2nd previous year
58	CONS1	Num	5	Consanguinity during the previous year
59	CONS0	Num	5	Actual Consanguinity (Empty field, until the end of the year)
60	NASCITUR	Num	5	Consanguinity calculated on future calves
61	N_CONS5*	Num	4	N. of animals born during the 5th previous year
62	N_CONS4*	Num	4	N. of animals born during the 4th previous year
63	N_CONS3*	Num	4	N. of animals born during the 3rd previous year
64	N_CONS2*	Num	4	N. of animals born during the 2nd previous year
65	N_CONS1*	Num	4	N. of animals born during the previous year
66	N_CONS0	Num	4	N. of animals born during the current year (Empty field, until the end of the year)
67	N_NASCITUR*	Num	4	N. of future calves, i.e., ongoing pregnancies
68	I_CONTR_LG*	Date	8	First visit date
69	ULT_CONTR*	Date	8	Last visit date
70	NUM_CONTR*	Num	6	Overall n. of visits
71	ABORTI	Num	3	Percentage of abortions
72	N_ABORTI	Num	6	N. of abortions
73	PNASC_M	Num	2	Males weight at birth
74	PNASC_F	Num	2	Females weight at birth
75	N_PNASC_M	Num	6	N. of males with PNASC_M
76	N_PNASC_F	Num	6	N. of females with PNASC_M
77	N_PART_VAC	Num	6	N. of pluriparae with a calving as last event

Continued on next page

Table 5.1 – continued from previous page				
	Field Name	**Type**	**Width**	**Description**
78	N_PART_MAN	Num	6	N. of primiparae with a calving as last event
79	ULT_ELAB	Char	1	Previous elaboration date
80	MANZ_N_IND	Num	3	Birth ease (EBV for heifers)
81	MANZ_PARTO	Num	3	Calving ease (EBV for primiparae)
82	MANZ_ACCR	Num	3	Growing (EBV for heifers)
83	MANZ_MUSC	Num	3	Muscularity (EBV for heifers)
84	MANZ_CARNE	Num	3	Meat Index (heifers)
85	MANZ_ALLEV	Num	3	Breeding Index (heifer)
86	CODICE_ASL	Char	8	Farm ID code for ASL, Azienda Sanitaria Locale
87	TFA_N_IND	Num	4	N. of TFA bulls used to calculate EBVs and Selection Indices
88	TFA_ALLEV	Num	3	Breeding Index (A.I. bulls)
89	TFA_CARNE	Num	3	Meat Index (A.I. bulls)
90	TFA_MUSC	Num	3	Muscularity (EBV for A.I. bulls)
91	TFA_ACCR	Num	3	Growing (EBV for A.I. bulls)
92	TFA_NASC	Num	3	Birth ease (EBV for A.I. bulls)
93	TFA_PARTO	Num	3	Calving ease (EBV for A.I. bulls)
94	TFN_ALLEV	Num	3	Breeding Index (N.I. bulls)
95	TFN_CARNE	Num	3	Meat Index (N.I. bulls)
96	TFN_MUSC	Num	3	Muscularity (EBV for N.I. bulls)
97	TFN_ACCR	Num	3	Growing (EBV for heifers)
98	TFN_NASC	Num	3	Birth ease (EBV for N.I. bulls)
99	TFN_PARTO	Num	3	Calving ease (EBV for N.I. bulls)
100	TFN_N_IND	Num	4	N. of TFN bulls used to calculate EBVs and Selection Indices
101	MORTALITA	Num	5	Perinatal mortality
102	VIT_X_VACC	Num	4	N. of viable calves per cow per year predicted with 3.1
103	INS1	Num	3	N. of females inseminated between [1-50] days after calving
104	INS2	Num	3	N. of females inseminated between [50-70] days after calving
105	INS3	Num	3	N. of females inseminated between [70-100] days after calving
106	INS4	Num	3	N. of females inseminated between >100 days after calving
107	CONC1	Num	3	N. of females pregnant between [1-80] days after calving
108	CONC2	Num	3	N. of females pregnant between [80-110] days after calving
109	CONC3	Num	3	N. of females pregnant between [110-140] days after calving
110	CONC4	Num	3	N. of females pregnant between >140 days after calving

Continued on next page

Table 5.1 – continued from previous page				
	Field Name	**Type**	**Width**	**Description**
111	INTP1	Num	3	Contains CONC1
112	INTP2	Num	3	Contains CONC2
113	INTP3	Num	3	Contains CONC3
114	INTP4	Num	3	Contains CONC4
115	SAL1	Num	3	N. of pregnant females with one insemination
116	SAL2	Num	3	N. of pregnant females with two insemination
117	SAL3	Num	3	N. of pregnant females with three insemination
118	SAL4	Num	3	N. of pregnant females with four insemination
119	VAC1*	Num	3	Percentage of currently pregnant cows
120	VAC2*	Num	3	Percentage of currently inseminated cows
121	VAC3*	Num	3	Percentage of cows currently ready for insemination or which insemination failed
122	VAC4*	Num	3	Percentage of currently post-partum cows
123	INTP_EST1	Num	3	Calving interval based on the season of calving (mid spring-mid summer)
124	INTP_EST2	Num	3	Calving interval based on the season of calving (mid summer-mid fall)
125	INTP_INV1	Num	3	Calving interval based on the season of calving (mid fall-mid winter)
126	INTP_INV2	Num	3	Calving interval based on the season of calving (mid winter-mid spring)
127	PRIMIPARE	Num	4	N. of primiparae
128	PLURIPARE	Num	4	N. of pluriparae
129	CAPI_ALTRI	Num	8	N. of non-Piedmontese cattle
130	VACC_ALTRE	Num	8	N. of non-Piedmontese cows
131	UBA	Num	8	Unità Bovino Adulto - LIVESTOCK UNIT
132	UBA1	Num	8	UBA referred to bovines elder than 2 years
133	UBA06	Num	8	UBA referred to bovines 6 months-2 years old
134	UBA04	Num	8	UBA referred to bovines 4-6 months old
135	SP1	Char	1	Blank Field
136	INTPS_6*	Num	8	Sum of Calvin Intervals days on cows that delivered during the 6th previous year
137	N_INTPS_6*	Num	3	N. deliveries occurred during the 6th previous year
138	INTPS_5*	Num	8	Sum of Calvin Intervals days on cows that delivered during the 5th previous year
139	N_INTPS_5*	Num	3	N. deliveries occurred during the 5th previous year
140	INTPS_4*	Num	8	Sum of Calvin Intervals days on cows that delivered during the 4th previous year
141	N_INTPS_4*	Num	3	N. deliveries occurred during the 4th previous year

Continued on next page

Table 5.1 – continued from previous page				
	Field Name	**Type**	**Width**	**Description**
142	INTPS_3*	Num	8	Sum of Calvin Intervals days on cows that delivered during the 3rd previous year
143	N_INTPS_3*	Num	3	N. deliveries occurred during the 3rd previous year
144	INTPS_2*	Num	8	Sum of Calvin Intervals days on cows that delivered during the 2nd previous year
145	N_INTPS_2*	Num	3	N. deliveries occurred during the 2nd previous year
146	INTPS_1*	Num	8	Sum of Calvin Intervals days on cows that delivered during the previous year
147	N_INTPS_1*	Num	3	N. deliveries occurred during the previous year
148	TOT_M_4*	Num	3	N. of total primiparous deliveries occurred during the 4th previous year
149	DIFF_M_4*	Num	3	N. of difficult primiparous deliveries occurred during the 4th previous year
150	CESA_M_4*	Num	3	N. of cesarean primiparous deliveries occurred during the 4th previous year
151	TOT_M_3*	Num	3	N. of total primiparous deliveries occurred during the 3rd previous year
152	DIFF_M_3*	Num	3	N. of difficult primiparous deliveries occurred during the 3rd previous year
153	CESA_M_3*	Num	3	N. of cesarean primiparous deliveries occurred during the 3rd previous year
154	TOT_M_2*	Num	3	N. of total primiparous deliveries occurred during the 2nd previous year
155	DIFF_M_2*	Num	3	N. of difficult primiparous deliveries occurred during the 2nd previous year
156	CESA_M_2*	Num	3	N. of cesarean primiparous deliveries occurred during the 2nd previous year
157	TOT_M_1*	Num	3	N. of total primiparous deliveries occurred during the previous year
158	DIFF_M_1*	Num	3	N. of difficult primiparous deliveries occurred during the previous year
159	CESA_M_1*	Num	3	N. of cesarean primiparous deliveries occurred during the previous year
160	TOT_V_4*	Num	3	N. of total pluriparous deliveries occurred during the 4th previous year

Continued on next page

Table 5.1 – continued from previous page				
	Field Name	**Type**	**Width**	**Description**
161	DIFF_V_4*	Num	3	N. of difficult pluriparous deliveries occurred during the 4th previous year
162	CESA_V_4*	Num	3	N. of cesareans pluriparous deliveries occurred during the 4th previous year
163	TOT_V_3*	Num	3	N. of total pluriparous deliveries occurred during the 3rd previous year
164	DIFF_V_3*	Num	3	N. of difficult pluriparous deliveries occurred during the 3rd previous year
165	CESA_V_3*	Num	3	N. of cesareans pluriparous deliveries occurred during the 3rd previous year
166	TOT_V_2*	Num	3	N. of total pluriparous deliveries occurred during the 2nd previous year
167	DIFF_V_2*	Num	3	N. of difficult pluriparous deliveries occurred during the 2nd previous year
168	CESA_V_2*	Num	3	N. of cesareans pluriparous deliveries occurred during the 2nd previous year
169	TOT_V_1*	Num	3	N. of total pluriparous deliveries occurred during the previous year
170	DIFF_V_1*	Num	3	N. of difficult pluriparous deliveries occurred during the previous year
171	CESA_V_1*	Num	3	N. of cesareans pluriparous deliveries occurred during the previous year
172	ALLEVF_6*	Num	8	Sum of breeding index on females currently alive grouped by age (6 years-old) with breeding index
173	N_ALLEVF_6*	Num	3	N. of females currently alive grouped by age (5 years-old) with breeding index
174	ALLEVF_5*	Num	8	Sum of breeding index on females currently alive grouped by age (5 years-old) with breeding index
175	N_ALLEVF_5*	Num	3	N. of females currently alive grouped by age (5 years-old) with breeding index
176	ALLEVF_4*	Num	8	Sum of breeding index on females currently alive grouped by age (4 years-old) with breeding index
177	N_ALLEVF_4*	Num	3	N. of females currently alive grouped by age (4 years-old) with breeding index

Continued on next page

Table 5.1 – continued from previous page				
	Field Name	**Type**	**Width**	**Description**
178	ALLEVF_3*	Num	8	Sum of breeding index on females currently alive grouped by age (3 years-old) with breeding index
179	N_ALLEVF_3*	Num	3	N. of females currently alive grouped by age (3 years-old) with breeding index
180	ALLEVF_2*	Num	8	Sum of breeding index on females currently alive grouped by age (2 years-old) with breeding index
181	N_ALLEVF_2*	Num	3	N. of females currently alive grouped by age (2 years-old) with breeding index
182	ALLEVF_1*	Num	8	Sum of breeding index on females currently alive grouped by age (1 year-old) with breeding index
183	N_ALLEVF_1*	Num	3	N. of females currently alive grouped by age (1 year-old) with breeding index
184	CARNEF_6*	Num	8	Sum of meat index on females currently alive grouped by age (6 years-old) with meat index
185	N_CARNEF_6*	Num	3	N. of females currently alive grouped by age (6 years-old) with meat index
186	CARNEF_5*	Num	8	Sum of meat index on females currently alive grouped by age (5 years-old) with meat index
187	N_CARNEF_5*	Num	3	N. of females currently alive grouped by age (5 years-old) with meat index
188	CARNEF_4*	Num	8	Sum of meat index on females currently alive grouped by age (4 years-old) with meat index
189	N_CARNEF_4*	Num	3	N. of females currently alive grouped by age (4 years-old) with meat index
190	CARNEF_3*	Num	8	Sum of meat index on females currently alive grouped by age (3 years-old) with meat index
191	N_CARNEF_3*	Num	3	N. of females currently alive grouped by age (3 years-old) with meat index
192	CARNEF_2*	Num	8	Sum of meat index on females currently alive grouped by age (2 years-old) with meat index
193	N_CARNEF_2*	Num	3	N. of females currently alive grouped by age (2 years-old) with meat index
194	CARNEF_1*	Num	8	Sum of meat index on females currently alive grouped by age (1 year-old) with meat index
Continued on next page				

Table 5.1 – continued from previous page				
	Field Name	**Type**	**Width**	**Description**
195	N_CARNEF_1*	Num	3	N. of females currently alive grouped by age (1 year-old) with meat index
196	PARTOF_6*	Num	8	Sum of Calving Ease EBVs on females currently alive grouped by age (6 years-old) with Calving Ease
197	N_PARTOF_6*	Num	3	N. of females currently alive grouped by age (6 years-old) with Calving ease
198	PARTOF_5*	Num	8	Sum of Calving Ease EBVs on females currently alive grouped by age (5 years-old) with Calving Ease
199	N_PARTOF_5*	Num	3	N. of females currently alive grouped by age (5 years-old) with Calving ease
200	PARTOF_4*	Num	8	Sum of Calving Ease EBVs on females currently alive grouped by age (4 years-old) with Calving Ease
201	N_PARTOF_4*	Num	3	N. of females currently alive grouped by age (4 years-old) with Calving ease
202	PARTOF_3*	Num	8	Sum of Calving Ease EBVs on females currently alive grouped by age (3 years-old) with Calving Ease
203	N_PARTOF_3*	Num	3	N. of females currently alive grouped by age (3 years-old) with Calving ease
204	PARTOF_2*	Num	8	Sum of Calving Ease EBVs on females currently alive grouped by age (2 years-old) with Calving Ease
205	N_PARTOF_2*	Num	3	N. of females currently alive grouped by age (2 years-old) with Calving ease
206	PARTOF_1*	Num	8	Sum of Calving Ease EBVs on females currently alive grouped by age (1 year-old) with Calving Ease
207	N_PARTOF_1*	Num	3	N. of females currently alive grouped by age (1 year-old) with Calving ease
208	ETA_ELIM_M*	Num	8	Age in days of culled males, between 12-24 months-old
209	N_ELIM_M*	Num	3	N. of culled males, between 12-24 months-old
210	N_ELIM_NN*	Num	3	N. of dead calves in the first 60 days after birth
211	NATI_TOT*	Num	3	Total number of calves born
212	N_ELIM_M_1*	Num	3	N. of culled males, between 12-15 months-old
213	N_ELIM_M_2*	Num	3	N. of culled males, between 15-24 months-old
214	NATI_FA*	Num	3	Total number of calves born from Artificial Insemination
215	NATI_FN*	Num	3	Total number of calves born from Natural Impregnation
216	NATI_GM*	Num	3	Total number of calves born from one multiple-sire group
217	NATI_VIVI*	Num	3	Total number of calves born alive

Continued on next page

Table 5.1 – continued from previous page				
	Field Name	**Type**	**Width**	**Description**
218	RGV*	Num	8	Total number of cows registered in the corresponding Herd-Book section
219	RGT*	Num	8	Total number of bulls registered in the corresponding Herd-Book section

Table 5.1: Raw variables contained in the available original data set. Variables refers to mean values among periods of 365 days. Punctual values referred to the last data-load or cumulative ones (among the past 365 days) are denoted with * instead.

5.2 Data Selection and Editing

As already mentioned in Chapter 2, the highest concentration of farms is established in Piedmont. Moreover, the most representative ones are the breedings counting more than 26 cows (Figure 2.1). To work with a descriptive data set, breedings located in Piedmont with at least 30 cows and a percentage of artificial insemination between 90% and 100% were selected. In a first study (Chapter 6), farms exhibiting updated visits during 2017 and 2018 were selected. However, as later the possible investigation involving also vectorial approach was considered, only farms exhibiting continuous visits over a reasonable period, e.g. five years, were acquired (Chapters 7 and 8). Constant recordings between 2014-2019 were then considered. As a result, most recent farms were discarded from the study, as their management still could not be completely defined. Similarly, breedings closed between '14 and '19 were excluded, to maintain a pool of contemporary farms with comparable data. In brief, the main filters commonly imposed to select farms to work with include the following criteria:

- location in Piedmont,
- consistencies for cows greater than or equal to 30,
- percentage of artificial insemination between 90% and 100%,

The herds were extracted differently for the analysis described in Chapter 9, as illustrated in the corresponding chapter. Once these farms were selected, it was possible to extract the reports referred to any period in the time window, e.g. 2017-2018 (Chapter 6), 2018-2019 (Chapter 9), or to use all the five-years information (Chapter 7 and 8). Along with the research project, various selections were performed, based on the different pursued goals. In each chapter that follows, the reference data sets are described, focusing on the extraction

carried out in individual cases. Similarly, the variables were filtered differently according to the cases under analysis. The parameters considered are described in each chapter.

Finally, for all the cases under analysis, the variable used by the ML methods as the target variable was constructed, as it was not directly available in the original data set. As illustrated in Section 4.3, the addressed issue falls into the category of supervised learning. In other words, the algorithms exploit real values to calibrate the prediction models. Since the aim is the prediction of the number of weaned calves per cow produced annually (Chapter 3), the actual amount was extracted for the years 2018 and 2019. Depending on the time frame involved, the appropriate target was isolated. In other words, working on data referred to the year 2017 implied the target extraction from 2018, since the goal is the prediction for the following year. The target was originated from 2019 when managing the data recorded in 2018. Operating on the 2014-2017 vectors, described in Chapter 8, the target was extrapolated from 2018. For each farm and all years, the target attribute Y was obtained with the formula below, including the values of the number of the calves born alive, those unable to survive during weaning period, and the number of cows (i.e., NATI_VIVI, N_ELIM_NN, VACCHE), in the corresponding year:

$$Y = \frac{NATI\_VIVI - N\_ELIM\_NN}{VACCHE}. \tag{5.1}$$

As for Calving Interval, it is necessary to highlight that variable 20 ($INTERPARTO$) in Table 5.1 is a quantity based on currently pregnant cows, hence contributing to the prediction for the number of viable calves for the next year. It is not a representative variable for the herd Calving Interval since it does not consider the information on the total number of cows in the herd. In order to give the ML algorithms the possibility to process all the available information, it is more appropriate to provide both the total calving interval and the number of cows currently pregnant, so that the two variables can eventually be combined with other features, in the most appropriate way according to the algorithm to predict the target. The Calving Interval $INTP$ used throughout the research study is hence derived from the division of variables 146 and 147, i.e., between the sum of Calving Intervals days and the overall number of deliveries occurred, both referring to the previous year:

$$intp = \frac{INTPS\_1}{N\_INTPS\_1}. \tag{5.2}$$

Many variables contain sensitive data (e.g. farm ID, owner name, address, visiting technician code) and were immediately removed. Others are redundant information, divided by years, or incomplete over years and

AZIENDA	ANNO_CONTR	PRIMIPARE	PLURIPARE	MANZE	VITELLE	VITELLI	INTP
Farm 1	2014	22	36	7	35	30	365
Farm 1	2015	10	46	13	43	31	375
Farm 1	2016	16	47	12	41	34	381
Farm 1	2017	14	46	11	49	41	375
Farm 1	2018	16	47	12	43	30	374
Farm 1	2019	15	43	10	41	36	378
Farm 2	2014	11	90	9	33	25	396
Farm 2	2015	10	93	9	40	24	391
Farm 2	2016	9	95	7	33	33	380
Farm 2	2017	7	97	10	28	25	387
Farm 2	2018	9	92	11	35	29	385
Farm 2	2019	13	85	13	30	35	380
Farm 3	2014	7	42	3	21	17	414
Farm 3	2015	4	43	4	27	6	439
Farm 3	2016	4	44	10	25	12	452
Farm 3	2017	10	44	11	14	10	425
Farm 3	2018	9	60	4	33	30	473
Farm 3	2019	12	58	7	31	34	465

Table 5.2: Standard Data Panel. Structure of the data set. The farms are listed horizontally, as well as the reference year, the variables from Table 5.1 vertically.

breedings. Therefore they were not considered. Anyway, the involved ones were renamed, for an immediate reference to their meaning, but the original names are listed in the Table 5.1. After renaming the farms, sorting by breeding and increasing year, the general data set has the structure shown in Table 5.2:

5.3 Dataset Configuration for ML Process

The study carried out took shape from the analysis of the summary data from 2017, to build the best predictive model for the number of weaned calves per cow produced in 2018. Setting this goal, it was, therefore, necessary to manage a data set containing input variables for each farm. Given n instances and m variables, the dataset configuration among 2017-2018 (shown in Table 5.3) consisted in m input scalar attributes $X_{17,i}$ where $i = 1, ..., m$ for each of the n farms. The number of weaned calves produced per cow in 2018 was obtained with 5.1, which was named Y_{18} in this case.

Different breedings were selected among the two-year period 2018-2019. However, the structure of the data set is similar, i.e., a series of m variables for n breedings collected as input among 2018, i.e., $X_{18,j,i}$ where

	2017				2018
	$X_{17,j,1}$ VACCHE	$X_{17,j,2}$ ETA_VACCHE	$X_{17,j,3}$ INTERPARTO	$X_{17,j,4}$ N_PARTI	$Y_{18,j}$
FARM 1 -	104	3020	387	60	0,95
FARM 2 -	54	3112	425	54	0,9
FARM 3 -	63	2824	515	48	0,69
...	49	3131	466	49	0,67
	108	2766	407	50	0,85
	74	3448	459	62	0,84

Table 5.3: Dataset configuration among 2017-2018. On the left side the input scalar variables $X_{17,1}, X_{17,2}, ..., X_{17,m}$. On the right side the scalar target Y_{18}

$i = 1, ..., m$ and $j = 1, ..., n$, and the target Y_{19} was extracted from 2019.

Concerning a time series approach, the standard panel data set (Table 5.2) is not suitable. Such structured records do not show a temporal dependency to the 'eyes' of the algorithm, which could not handle the information properly. The data were then arranged in order to manage the temporal information from 2014 to 2017 for each farm. Variables were adapted to a vectorial structure, as each attribute assumes different values over time for each instance. Their values were hence collapsed neatly, starting from the farthest value in time (y. 2014) to the most recent value (y. 2017). In such a case, each of the m variable is represented as a vector $\boldsymbol{X}_{t,j,i}$ where t varies between the years 2014 and 2017, $i = 1, ..., m$, and $j = 1, ..., n$. The result is configured as

$$\boldsymbol{X}_{t,j,i} = [X_{14,j,i}, X_{15,j,i}, X_{16,j,i}, X_{17,j,i}],$$

where $t \in \{14, ..., 17\}$, $i = 1, ..., m$, and $j = 1, ..., n$, and one scalar target Y_{18} is assigned for each breeding. A graphical representation is given in Table 5.4.

Finally, the division of the dataset into a learning and a test set was performed. For each benchmark problem, illustrated in the following chapters, the splitting is described, as it was performed differently for each of the pursued investigations. In general, once the corresponding training, validation, and test sets are obtained, the same mechanism is applied in order to build predictive models. The test is "kept hidden", i.e., it is not shown to any of the techniques involved in the learning phase. Only the training and the validation sets are initially involved. The test set is only used in the final step, when, once the predictive models specifically set up by administering the learning instances are available, it is necessary to test their generalization skills. The hard core of the process consists above all in this. Setting the parameters so that the techniques can

	2014-2017			2018
	$X_{t,1,j}$ VACCHE	$X_{t,2,j}$ ETA_VACCHE	$X_{t,3,j}$ INTERPARTO	$Y_{18,j}$
FARM 1 -	[98,101,107,104]	[2999,3001,2998,3020]	[391,391,380,387]	0,95
FARM 2 -	[61,49,53,54]	[3076,3002,3056,3112]	[408,376,402,425]	0,9
FARM 3 -	[53,55,64,63]	[2799,2813,2802,2824]	[367,376,406,515]	0,69
...	[31,36,47,49]	[3102,3075,3009,3131]	[434,480,461,466]	0,67
	[102,99,105,108]	[2704,2795,2789,2766]	[404,371,395,407]	0,85
	[69,71,75,74]	[3401,3388,3406,3448]	[387,367,373,459]	0,84

Table 5.4: Vectorial panel data set configuration for 2014-2018. On the left side the input vectorial variables $\boldsymbol{X}_{t,j,i} = [X_{14,j,i}, X_{15,j,i}, X_{16,j,i}, X_{17,j,i}]$, with $t \in \{14, ..., 17\}$, $i = 1, ..., m$, and $j = 1, ..., n$. On the right side the scalar target variable Y_{18}

properly learn can be complex. Finding the right combination so that a trained model is able to generalize the concepts is not taken for granted. Different parameters tunings are needed, as well as multiple runs of the algorithms, to analyze the general behavior of the algorithm on the provided data, and different subdivisions of the dataset, to keep a balanced distribution of instances when assigning them to the learning and the testing set.

Chapter 6

A GP Approach to Precision Farming

6.1 Introduction

In this Chapter the first approach adopted with ML techniques is presented. In particular, GP was adopted among the different techniques. Thanks to its structural characteristics, presented in Chapter 4, GP is suitable for addressing the search towards simple and intelligible predictive models. The yearly number of calves produced per cow (Model 3.1) is the current method to estimate the ongoing farm performance, exploiting the calving interval and perinatal mortality. Differently from this model, the estimate of the future trend is more propaedeutic to evaluate the breeding performance. By appropriately processing the data through techniques capable of finding patterns that link the representative variables and the actual number of calves weaned per cow recorded in the following year, it is possible to propose a literally predictive measure. Without making a priori assumptions about the relationship between the response and the independent variables, ML techniques may provide interesting feature selection characteristics, representing a flexible and robust alternative in predictors identification. Specifically, the potential of GP is investigated, to create and to analyze predictive models for the number of weaned calves in Piemontese cattle breedings, which could improve the analysis of Piemontese breeding performance. Inside the ML arena, GP has a set of interesting characteristics that distinguish it from many other methods. Other than assuming no hypotheses about the shape of the final model, characteristic that is intrinsic to all ML methods, after setting appropriate parameters, GP can generate readable and interpretable models, which is crucial for our application. Moreover, GP is able to perform an automatic feature selection, thus relieving us from any pre-processing task. To investigate the efficiency of GP, a dataset composed by observations on representative Piemontese breedings was used. The results show that the algorithm is appropriate and can perform an implicit feature selection, highlighting important variables and leading to simple and interpretable models. Considering that the algorithm can address all the defined objectives, GP represents the research baseline.

6.2 Matherials and Methods

6.2.1 Propaedeutic Preparation of the Dataset

The restrictions listed in Section 5.2 were applied to the main dataset, that originally consisted of 633063 records, each corresponding to a visit performed by technicians in each farm. At this stage, multiple records corresponded to each farm. Filters listed in Section 5.2 were imposed to obtain a solid representative subset: breeding located in Piemonte with at least 30 cows and percentage of artificial insemination between 90% and 100% were selected, with updated visits during 2017 and 2018. Thereby, a total of 725 breedings were taken into account and the most recent visit was extracted for all farms for both years, i.e., the visits occurring performed between November and December. Each breeding was then represented with 6 instances. Among them, the instances referring to 2018 were used to derive the target through Equation 5.1. The dataset obtained make it possible to perform two types of analysis: a first one in which, for each breeding, the subset referred to 2017 can be used to predict the target in 2018, and a second one in which the whole available series recorded previously to the target can be analyzed. As a first general approach performed to test the effectiveness of ML techniques, the first outlined benchmark was therefore initially inspected. The second kind of approach, presented in Chapter 8, that is the vectorial one, required data editing to apply techniques specific to manage vectors representing time series. Since the performance of the farm mainly focuses on fertility, the data concerning multiparae cows were considered to elaborate the number of deliveries and the calving intervals. In the same way, data referred to bulls used for artificial insemination were maintained (i.e., selection indices, representing namely estimations of the additive genetic effect of a subject for specific traits). Information referred to inbreeding levels between animals were not incorporated into the study at this stage, since they required more investigations. However, they were included in the subsequent development of the study, for a more accurate inspection on the consanguinity of unborn calves. Among the filtered farms, two main groups were identified: a smaller one, containing 330 unique breedings, and a larger one, consisting of 395 breedings, that differs from the first group in being characterized by the use of many different bulls, pursuing natural impregnations. In the first group, instead, there were farms in which only artificial insemination is performed, in some cases combined for impregnation with the single owned bull. The main difference between the two sets results in a wider use of owned bulls: this means that, instead of recording the date on which the insemination took place, breedings belonging to the second group usually set a period of several days, followed by the diagnosis of the pregnancy. As both datasets are representative for the Piemontese breeding reality, where the second dataset features a more diffused situation and the first one depicts the most accurate one, both groups were incorporated in this first study, as propaedeutic to the objective. Since the aim is the inference of the target by means of models obtained with ML techniques,

the first set of farms was designated as a learning set, since the algorithm can learn from precise recordings, while the second set was designated as a test set. After all, each record of the final datasets stood for a single farm with variables 1 – 19 referred to year 2017, whereas Y represents the actual number for weaned calves recorded in 2018 (Table 6.1). All variables can only assume positive values and the target variable assumed values in the range $[0.26; 1.24]$.

	Reference Year	**Variable Name**	**Reference to Table 5.1**
1	2017	$COWS$	4
2	2017	C_{AGE}	14
3	2017	$INTP$	obtained with 5.2
4	2017	C_{PAR}	16
5	2017	N_{PAR}	17
6	2017	C_{EASE}	45
7	2017	C_{GRAVID}	119
8	2017	C_{INS}	120
9	2017	$BIRTHW_M$	73
10	2017	$BIRTHW_F$	74
11	2017	IND_{PAR}	49
12	2017	TFA_{BIRTH}	92
13	2017	TFA_{PAR}	93
14	2017	N_{ELIM}	210
15	2017	N_{TOT}	211
16	2017	N_{BALIVE}	217
17	2017	$N_{CORRECT}$	35
18	2017	$ABORT$	71
19	2017	$MORT$	101
20	2018	Y	Target Variable 5.1

Table 6.1: Final set of variables used for the first benchmarked problem. The bottom line represent the dependent variable Y, i.e., the target for the predicted models generated by GP based on the set of independent variables.

6.2.2 Application of GP

As mentioned in the previous section, the first group of farms (size 330) was used as a learning set, while the second one (size 395) as a test set. To obtain the best performance from the algorithm, it is necessary to identify two parts of the learning set, namely the training and the validation set (Section 4.2). Several runs of GP are performed. Therefore, for each of them, the dataset is split again, in order to work on different portions of the dataset each time. We considered the possibility of dividing the learning set through a k-fold cross validation approach, in order to obtain the training and validation subsets. However, the reduced set of data did not allow the identification of a suitable value for k: for instance, if we had chosen k smaller than

10, we would have obtained a small number of subsets, leading to a small number of runs (i.e., fewer than 10). On the contrary, with k greater than 10, we would have had a restrained number of records within the test sets (i.e., less than 33 validation instances for each run). Indicatively, it is advisable to have a number of runs that allow one to obtain statistically significant results, i.e., at least 30. Hence, the learning set was split into 30 different subsets, with constant training-validation partitioning (75%-25%), in order to perform 30 runs of GP. At each run, divisions were carried out with a random choice of records without repetition, keeping training and validation separate. In other words, among the total 330 learning records, 83 records were chosen to form the validation set, and the remaining 247 were labeled as training ones, reiterating the process with different sets for all 30 runs. A final check on the selected instances was performed in order to ensure that all instances had been included, that is the union of all the training sets had to be equal to the whole learning set. The population of individual obtained for each run on the training set was evaluated on the validation set, in order to select the best ones (i.e., models with the lowest error on the validation set). Finally, the generalization ability of the latter was checked, by analyzing the respective error achieved on the test set. Individuals were generated with GP using a tree-based representation, where the trees were built using a set of terminal symbols T and a set of primitive functional symbols F. The set T was composed by the previously described variables (Table 6.1). The set F was equal to {`plus; minus; times; mydivide`}, where `plus, minus and times` indicate the usual operators of binary addition, subtraction and multiplication, respectively, while `mydivide` represents the protected division, that returns the numerator when the denominator is equal to zero. In order to limit overfitting and maintain the models as simple as possible, besides crossover and mutation, operators such as shrinkmutation and swapmutation (predefined in GPLab) were used. These two operators, respectively, exchange a subtree with a terminal node and permutate non-commutative functions' elements. Table 6.2 reports the experimental settings.

Parameter	**Description**
ST-GP	
Maximum number of generations	20
Population size	500
Selection Method	Lexicographic Parsimony Pressure
Elitism	Keepbest
Initialization Method	Ramped half and half
Tournament Size	2
Subtree Crossover Rate	0.8
Subtree Mutation Rate	0.1
Subtree Shrinkmutation Rate	0.05
Subtree Swapmutation Rate	0.05

Table 6.2: Parameters used for GP in the former experimental study

6.3 Results

6.3.1 Fitness of Models and Overall GP Performance

The performance resulting from the simulations is reported in Figure 6.1, where the fitness among the 30 runs on the training, the validation and the test sets are presented. The Lilliefors test, performed with significance level α=0.05, showed that a normal distribution can be assumed only on the training set. Hence, we applied a Kruskal-Wallis test (α=0.05), under the alternative hypothesis that, at the end of the runs, the RMSEs do not have equal medians. Results entailed that there is no significant difference between the three distributions: given a p-value p=0.17, the null hypothesis was not rejected, that is the median values of the errors committed on the three sets are not different. The median value obtained on the test set allows us to affirm that the obtained models are able to generalize well on unseen data.

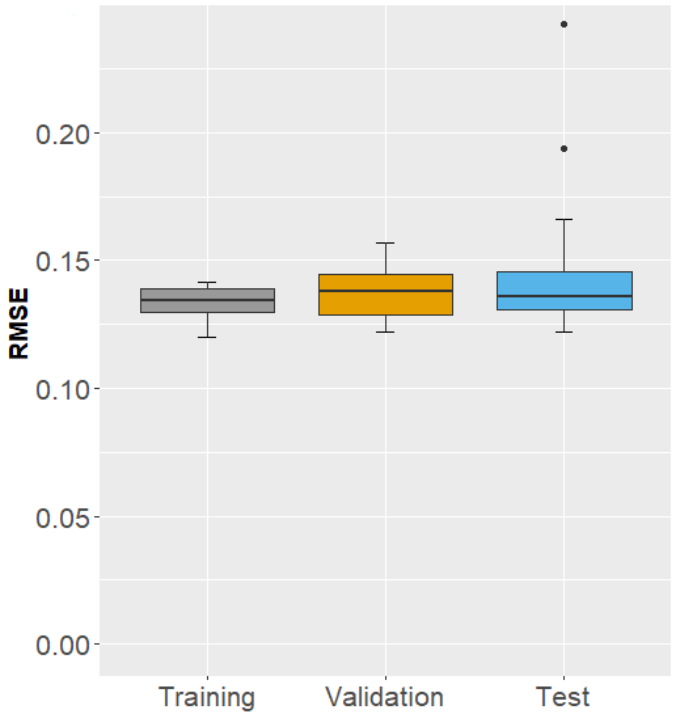


Figure 6.1: Performance of the best 30 selected models, respectively, on the training, validation and test sets. There is no significant difference between the results (Kruskal-Wallis test: p = 0.17, with α=0.05), i.e., the median values of the errors committed on the three phases are not different.

The frequency with which the variables appear across the different models indicates the possible key features and the negligible ones. Otherwise stated, considering the presence and absence of a feature within the model, it is possible to deduce its overall importance by analyzing the median of the feature presence, i.e., a binary variable. Predictors included by at least half of the best solutions on all the runs result in non-

null median frequencies, whereas negligible correspond to null median frequencies: values greater than zero suggest that the corresponding variables were used in over 50% of the final solutions, namely the number of cows ($COWS$), the number of occurred deliveries in the farm during the year (N_{PAR}), and the number of calves that were born alive (N_{BALIVE}). The information was confirmed also by the equivalent percentage, reported in Table 6.3.

Variable	**% of use on 30 run**
X1 – $COWS$	73%
X2 – C_{AGE}	27%
X3 – $INTP$	43%
X4 – C_{PAR}	27%
X5 – N_{PAR}	53%
X6 – C_{EASE}	40%
X7 – C_{GRAVID}	23%
X8 – C_{INS}	17%
X9 – $BIRTHW_M$	13%
X10 – $BIRTHW_F$	10%
X11 – IND_{PAR}	37%
X12 – TFA_{BIRTH}	13%
X13 – TFA_{PAR}	23%
X14 – N_{ELIM}	37%
X15 – N_{TOT}	43%
X16 – N_{BALIVE}	50%
X17 – $N_{CORRECT}$	37%
X18 – $ABORT$	23%
X19 – $MORT$	13%

Table 6.3: Median frequencies (percentage) of each variable among the best 30 individuals found by GP

Finally, the interpretability of the expressions was investigated, considering the number of variables involved in each of the best final models and the corresponding fitness. In order to compare the performance of GP models, the number of parameters encapsulated in each one was examined, paying attention to the corresponding fitness obtained on the test set (Table 6.4). Observing Table 6.4, a general trend can be identified: models that use fewer variables tend to have a worse fitness (i.e., a larger error) on the test set than those that use more variables. Among the 19 variables in the dataset, the obtained models include from a minimum of 3 to a maximum of 10 variables. An intermediate situation is represented by models involving 4 variables, since, in this case, the error is small and, as shown later, the expression is more interpretable. Two models were selected in order to make comparisons: the one showing the best fitness among all the evolved

expressions (GP_3 in Figure 6.2), and the one with the best fitness among the models that use 4 variables (GP_8 in Figure 6.2), chosen as straightforward interpretable.

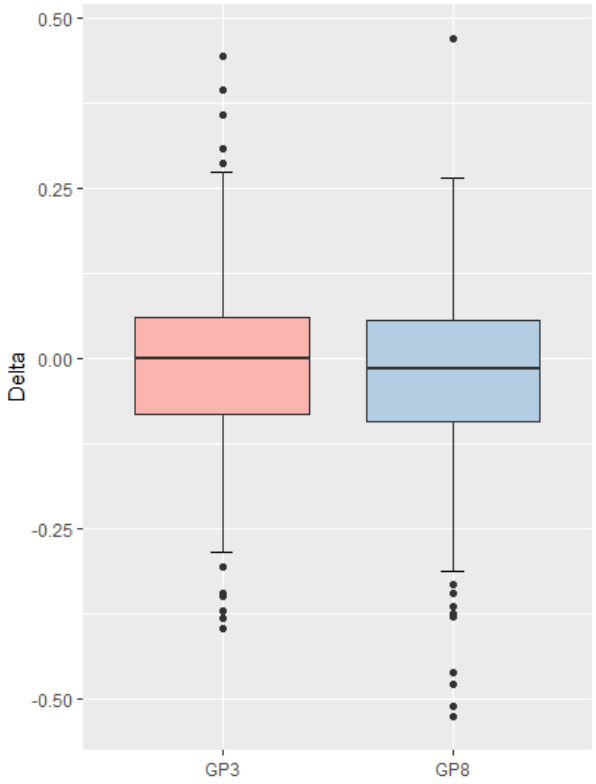


Figure 6.2: Comparisons between GP models on the test set. Distributions of the differences between predicted and real values are plotted. Both GP predicted values are not significantly different (Kruskal-Wallis: p - value = 0.2372). GP_3 shows a median value equal to -0.0005928782, smaller than the median value obtained with GP_8 (-0.0146762341)

For both Models GP_3 and GP_8, the distance values between predictions based on 2017 and target values Y_i recorded in 2018 are represented through boxplots, that is:

$$\Delta_{model,i} = Y_{model,i} - Y_i \tag{6.1}$$

for each record $i = 1, \ldots, 395$ in the test set. Predictions obtained with the two models GP_3 and GP_8 are not significantly different (Kruskal Wallis: p-value = 0.2372).

Prediction model	Fitness on test	N. of variables	% of used variables
model 1	0.1379	5	26%
model 2	0.1418	3	16%
model 3	0.1218	9	47%
model 4	0.1354	8	42%
model 5	0.1660	3	16%
model 6	0.1290	8	42%
model 7	0.1370	4	21%
model 8	0.1321	4	21%
model 9	0.1258	8	42%
model 10	0.1357	3	16%
model 11	0.2422	9	47%
model 12	0.1461	3	16%
model 13	0.1286	7	37%
model 14	0.1548	4	21%
model 15	0.1320	9	47%
model 16	0.1261	7	37%
model 17	0.1285	8	42%
model 18	0.1371	9	47%
model 19	0.1610	3	16%
model 20	0.1571	4	21%
model 21	0.1355	9	47%
model 22	0.1450	3	16%
model 23	0.1291	7	37%
model 24	0.1426	4	21%
model 25	0.1935	5	26%
model 26	0.1330	10	53%
model 27	0.1305	6	32%
model 28	0.1543	3	16%
model 29	0.1308	7	37%
model 30	0.1361	9	47%

Table 6.4: Fitness on the test set, number of involved variables and corresponding percentage are reported for each model evolved by GP in each of the 30 performed runs

6.3.2 Models Expression

The two selected models, whose expression is provided in Equations 6.2 and 6.3, perform likewise, incorporating different variables with respect to Y_p (see Equation 3.1). Parameters such as $MORT$ and N_{ELIM} used in Equation 3.1 were included also in GP expressions, i.e., mortality at 60 days in GP_8, and number

of calves born alive in both GP_3 and GP_8. Regarding GP_3, the expression in infix notation to obtain the predictions is:

$$Y_{GP_3} = \frac{X_{11}}{X_{17} + \frac{X_3}{X_{16}} + \frac{X_3}{X_6 \cdot \frac{2 \cdot X_{18} + X_{16}}{\frac{X_9}{X_{19}} + X_1}}}, \tag{6.2}$$

where

1- $COWS$,
3- $INTP$,
6- C_{EASE},
9- $BIRTHW_M$,
11- IND_{PAR},
16- N_BALIVE,
17- $N_{CORRECT}$,
18- $ABORT$,
19 – $MORT$.

In model GP_3 , the denominators of *mydivide* operator do not meet existence conditions, that is they can assume null values (e.g. perinatal mortality $X19$ is null for some records). It is not possible to assert that the *mydivide* operator is actually a division and the previous expression 6.2 cannot be further simplified. Contrary to GP_3 , the model for GP_8 is comprehensible:

$$Y_{GP_8} = \frac{X_5}{\frac{(X_5 \cdot X_{14} + X_{16})}{X_1} + X_1}. \tag{6.3}$$

Since we previously set the constraint in the dataset on farms with more than 30 cows, and the other variables can also assume only positive values, the denominators of *mydivide* in the latter model (6.3) are also positive. Indeed, the denominator cannot reach null levels, since the number of cows is added to a quantity greater than zero. Existence conditions are in this case always verified, and therefore the function *mydivide* is a division, leading to a simplified version of Model 6.3:

$$Y_{GP_8} = \frac{X_1 \cdot X_5}{X_1^2 + X_5 \cdot X_{14} + X_{16}} \tag{6.4}$$

where

X1 – $COWS$
X5 – N_{PAR}
X14 – N_{ELIM}
X16 – N_{BALIVE}

Model 6.3 can further be rewritten as

$$Y_{GP_8} = \left(\left(\frac{N_{PAR}}{COWS} \right)^{-1} + \frac{N_{ELIM}}{COWS} + \left(\frac{N_{BALIVE}}{COWS} \cdot \frac{1}{N_{PAR}} \right) \right)^{-1}. \tag{6.5}$$

The first term can be expressed as the inverse of the mean number of the yearly deliveries occurred in the farm, since the number of all deliveries is divided by the total number of cows ($\overline{N_{PAR}}$). Likewise, the second and third terms contain, respectively, the yearly number of calves per cow that did not survive during the weaning period ($\overline{N_{ELIM}}$) and the yearly number per cow of calves born alive ($\overline{N_{BALIVE}}$), that is:

$$Y_{GP_8} = \left(\frac{1}{\overline{N_{PAR}}} + \overline{N_{ELIM}} + \frac{\overline{N_{BALIVE}}}{N_{PAR}} \right)^{-1}. \tag{6.6}$$

Stated otherwise, by renaming the terms and performing basic operations, we obtained the following:

$$1 = n_j v_{1,j} + n_j v_{2,j} + n_j v_{3,j} \tag{6.7}$$

for $j = 1, \ldots, 725$, since we considered the complete dataset with all the selected farms (see Section 5), where:

n_j=$Y_{GP8,j}$
$v_{1,j} = \left(\overline{N_{PAR_j}}\right)^{-1}$
$v_{2,j} = \overline{N_{ELIM_j}}$
$v_{3,j} = \frac{\overline{N_{BALIVE_j}}}{N_{PAR_j}}$

It is straightforward that Equation 6.7 can be formulated as the sum of rescaled variables

$$1 = \widetilde{v}_{1,j} + \widetilde{v}_{2,j} + \widetilde{v}_{3,j} \tag{6.8}$$

where $\widetilde{v}_{i,j} = n_j v_{i,j}$ for $i = 1, 2, 3$. Thereby, it was possible to measure the contribution of each term in the sum expressed in Equation 6.8. The distributions of each $\widetilde{v}_{i,j}$ was statistically analyzed and the three boxplots were displayed (Figure 6.3). An extremely significant difference is verified to exist between all variables (Wilcoxon test with Bonferroni correction: $\alpha = 0.017$, $p \ll 0.001$). Moreover, we inspected how far the mean value of each variable is from the unit. We compared, one by one, the three distributions via a single sample Wilcoxon test. We set the alternative hypothesis that the distribution shows a mean value $\mu \neq 1$, with $\alpha = 0.05$. Once again, we found an extremely significant difference between the mean value of $\widetilde{v}_{i,j}$ from the value 1. Similarly, we compared the distributions with respect to 0. The results of the test were analogous to the previous ones: with extremely significant p-values ($p \ll 0.001$), we could deduce that $\widetilde{v}_{2,j}$ and $\widetilde{v}_{3,j}$ remain relevant parameters, even assuming values close to zero, providing hence a minimal contribute in Equation 6.7. In other words, we could assert that all the variables in Equation 6.8 are influent: in particular, $\widetilde{v}_{1,j}$ is the most important one, since its mean value was $\mu_1 = 0.951$, whereas $\widetilde{v}_{2,j}$ and $\widetilde{v}_{3,j}$ respectively showed $\mu_2 = 0.032$ and $\mu_3 = 0.021$. Model 6.6 can be simplified, to the point of being expressed as the sum of three parameters. We verified that these three parameters are the average number of parts occurred during the year in the herd $\left(\left(\overline{N_{PAR}}\right)^{-1}\right)$, the number of calves per cow that have not passed the weaning phase ($\overline{N_{ELIM}}$) and finally the number of calves per cow live births compared to the total number of parts of the herd during the year ($\overline{N_{BALIVE}}/N_{PAR}$). From a zootechical point of view, these are actually the main parameters that intuitively can give an idea of the economic performance of the farm. All of them play a significant role with respect to the response variable: more importance is given to the parameter $\left(\overline{N_{PAR}}\right)^{-1}$, that can be associated to the inverse of the mean calving interval (days between two deliveries) of the farm, whereas $\overline{N_{ELIM}}$ and $\overline{N_{BALIVE}}/N_{PAR}$ give a smaller contribute.

Summing up, the most frequent variables in the models, obtained with GP, are the number of cows in the farm ($COWS$), the number of deliveries occurred in the breeding (N_{PAR}) and the number of calves born alive (N_{BALIVE}). The calving interval ($INTP$) and the number of dead calves at 60 days (N_{ELIM}) are slightly less frequent. Perinatal mortality is not so frequent, meaning that it could play a minor role in the prediction. The most frequent variables included in the expression 6.2 are $COWS$ and N_{BALIVE}, followed by $INTP$ and $MORT$. Then there are 5 less frequent additional parameters that could, therefore, be relevant in the refinement of the prediction. The median error of predictions obtained with model GP_3 is slightly smaller than the one obtained with model GP_8. The latter, however, processes fewer variables, exploiting exactly the three most frequent ones, listed in Table 6.3.

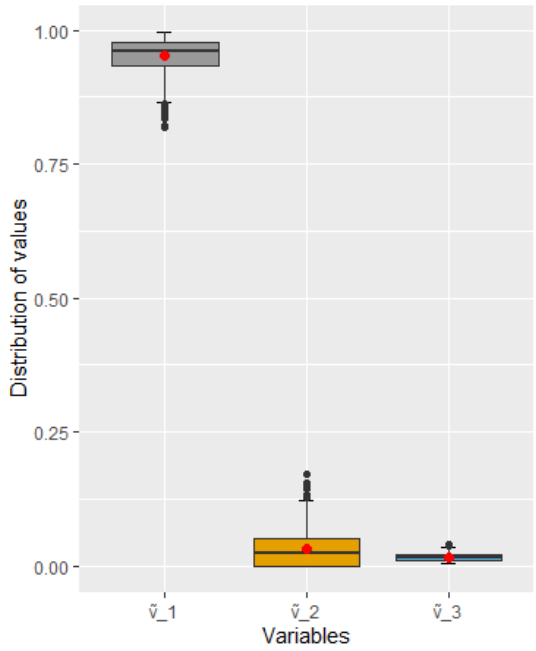


Figure 6.3: Boxplots of the distributions of the variables in Equation 6.8. Wilcoxon test with Bonferroni correction at α=0.017 reported p $\ll$ 0.001. Hence, the variables are significantly different. The single sample Wilcoxon test, with α=0.05, showed for each distribution a mean value $\mu \neq 1$(p $\ll$ 0.001). Therefore, all the variables are extremely significant in Equation 6.8. Mean values are respectively μ_1=0.951, μ_2=0.032 and μ_3=0.021 (red dots).

6.4 Conclusions

In this study, we investigated the performance of medium to large Piemontese cattle farms located in Piemonte. The currently used model (reported in Equation 3.1, Chapter 3) is employed to evaluate the current breeding performance, by rescaling the average calving interval referred to all the cows in the farm among the previous year, and to predict the number of calves per cow for the next year. The model is not completely suitable for representing the performance of the farms. Indeed, during the weaning period, many calves do not survive, entailing great losses to the economic revenues of the breedings (see Section 3). The reasons for those deaths are various and difficult to identify objectively, as they can be traced back to environmental conditions, to factors intrinsic to the animal itself, or connected to the feeding and breeding management. It is necessary to take into account crucial parameters, that encompass the calf's weaning in the output, as the number of calves born alive and those dead during the weaning period, i.e., among the 60 days following the birth. Variables featuring these aspects become very informative, but are not combined in the formulation of the previous model. The pursued goal is the construction of a predictive model that can exhaustively incorporate the mentioned information. Therefore, in light of these premises, an automatic learning method was applied, which can meet the requirements of pattern identification and informative

variables encapsulation. In addition, it is necessary to research and propose a simple model, which can be easily interpreted by the breeder. The expression to target should be a simplification and an added value to the management of the farm. The breeder should be able to easily read the information, in order to identify the critical points and strengths in production.

Given its ability to perform an automatic feature selection, a GP approach was applied to build predictive models, trained, validated and tested on data recorded in 2017 and 2018. Accurate models were achieved, and this means that GP can learn from a smaller dataset composed by representative farms and predict good results on the selected test set. Moreover, the algorithm was able to select and process important variables, without previous assumptions on the zootechnical aspect. The variety of expressions obtained by GP is composed of well-performing models that involve more parameters, resulting in a more complex expression, hardly reducible to a simpler one. However, other predictive models that encapsulate fewer variables were also achieved. Although these expressions have a slightly larger error, their formula can be extremely simple and possibly easier to interpret from the zootechnical point of view. The preliminary results obtained were presented at the 2020 international virtual conference WCCI, appearing in the conference proceedings [2].

It is therefore worth investigating further the application of GP to a larger dataset. In this first study, we focused on data directly referred to parturitions and artificial insemination, in order to process sound and solid data. The dataset was filtered and resized, and 19 variables were kept out of the whole 210 available: many were duplicate fields, aggregates of several variables, and even incomplete ones, as they were introduced later in the database of ANABORAPI. Parameters such as those related to heifers, i.e., bovines that did not give birth yet, were not considered, since we focused on data directly referred to cows, i.e., bovines that gave birth at least once. In breeding farms, heifers are mostly intended to the production of calves and are going to contribute to the restock of the herd. It appears necessary to investigate the behavior of GP and its features selection ability among these variables, as well as parameters referred to the bulls used for natural impregnation. To this purpose, their genetic indexes will be added to the analysis, as well as the levels of consanguinity of calves that will be born from ongoing pregnancies. The main intent consists however in the comparisons with other ML methods, to inspect better the potential of GP in the zootechnical field, and to explore possibly better models. Another aspect that emerges from the analysis of the model and, above all, the dataset is the need to represent information related to the management of environmental and feeding conditions on the farm. Such data are not available and hence require an appropriate collection of information such as for example the size of the boxes and the surface available to the animals, air and water quality and the composition of the food ration. These factors are usually considered as marginal. It is common to think that cow-calf problems are almost exclusively induced by genetic and pathological factors associated to pregnancy and childbirth. Indeed, not enough importance is given to the period after birth, in

which the cow and the calf need feeding and environmental conditions suitable for the respective postpartum and weaning phases. In this context, once again, the ability of GP to automatically select features could be very important to understand if and which of these variables are influential.

Chapter 7

Inside The Machine Learning Arena: Genetic Programming vs Other ML Methods

7.1 Introduction

The effectiveness of the type of management of a farm, i.e., the overall performance of the breeding, allows the breeder to consolidate the ongoing processes or, on the contrary, to adopt new management strategies. Among the Italian Piemontese Beef Breedings such a measure was identified as the yearly production of calves weaned per cow (Chapter 3). Modelling farm dynamics in order to predict the value of this parameter is a possible solution to investigate and highlight breeding strengths, and to find alternatives to penalizing factors. To solve this problem, a GP approach was proposed and described in Chapter 6 and presented in [2], consisting in a white-box technique suitable for big data management and with an intrinsic ability to select important variables, providing simple models. In the preliminary study, the dataset was investigated and a GP approach applied, in order to explore the possibility to address this task with GP. The method performed well, entailing that the ML horizon should be investigated further and that comparisons with other techniques should be carried out, even on larger datasets containing more features. In the previous experiment we extracted and processed 19 variables and we kept stricter filters on data: to perform GP, we selected the farms, based on the date of visit recorded between 2017 and 2018.

Considering the promising results, we further investigate the effectiveness of GP by including a greater number of variables, and evaluate its performance comparing the results with other ML techniques, usually applied to this kind of tasks. The dataset, besides being expanded vertically by including other variables that may be useful in predicting the target, is filtered by breedings presenting data updated over all years

between 2014 and 2019. In this way, breedings with a consolidated management are processed. Besides this, the creation of a solid pool of farms will make it possible to manage vectorial variables referred to the same breedings (Chapter 8). Emphasis is placed on the division of the dataset into different partitions, illustrating the need for the techniques to learn on a portion of the dataset and to test the prediction models on new instances. Afterwards, the GP baseline and other ML methods are applied. Once again the most frequent variables included in the models built by GP are highlighted, and their zootechnical significance is investigated a posteriori, evaluating the performance of the prediction models. The expressions are analyzed in order to propose a zootechnical interpretation of the equations. Comparisons with other common techniques, including also black-box methods, are performed in order to evaluate the performance of different type of methods in terms of accuracy and generalization ability. Among other ML techniques, some common methods were selected, including black-box ones as NN and RF to compare their results with those obtained with GP. Black-box models generally perform better, since their structure is able to capture the high non-linearity underlying data. However, as their definition suggests, they can be very unclear and do not explain the links between input and output variables, as well as the internal mechanisms leading to the results. The approach entailed constructive and helpful considerations on the addressed task, confirming its keyrole in the zootechnical field, especially in the beef breeding management.

7.2 Matherials and Methods

7.2.1 The Dataset

In this study, only the farms that show constant visits between 2014 and 2019 were considered. In this way, the effects related to farm management are solid and only the breedings with substantial data were kept. Indeed, even if the investigation is based on farms with data from 2017 as input and from 2018 as target, as a change in the type of management stabilizes over time, we considered breedings with historical records updated between 2014-2019, in order to focus on farms with a solid management. A newly started company does not have completely representative data. Moreover, the summary produced by ANABORAPI elaborates the average values of recordings related to the 365 days previous to the reference date. To avoid data from farms not yet fully operational, with gaps in registrations or close to resigning at the end of 2018, we set the restriction to companies active in the previous 5 years. As in our pilot study (Chapter 6, [2]), filters were imposed on breedings located in Piedmont with at least 30 cows and a percentage of artificial insemination between 90% and 100%. For each breeding, data recorded in 2017 and 2018 were considered and the record for the last check in the corresponding year was considered. Subsequently, for each farm, input and target variables were extracted, respectively from 2017 to 2018. However, two further conditions were added for

this research. As already stated, in order to keep in the pool of currently active breedings those with stable and consolidated situations, inspections had to be constantly carried out for at least more than two years. Moreover, the fact that breeders carry out between 90% and 100% of artificial insemination means that a part of the considered farms own bulls and carry out also natural impregnations. Most often, instead of recording the date on which the insemination took place, a period of several days followed by the pregnancy diagnosis is set. These farms were therefore excluded from the analysis. A main group of 304 representative Piemontese cattle farms results from the selection. Since the performance of the farm mainly focuses on fertility, data concerning multiparae cows were considered to elaborate the number of deliveries and the calving intervals. In the same way, data on bulls used for artificial insemination were maintained (i.e., EBVs, that represent namely estimations of the additive genetic effect of a subject). Parameters on heifers were included in the dataset, since these are bovines that did not give birth but, in breeding farms, are mostly intended for the production of calves. Moreover, since many breeders carry out also natural impregnation besides artificial insemination, data related to the bulls used for natural insemination were added to the analysis, as well as the levels of consanguinity of calves that would be born from ongoing pregnancies. The only strictly environmental measure available in the dataset, that was hence kept, is the Livestock Unit (LU or LSU): it has the purpose of synthetically expressing the zootechnical load, to easily compare the environmental impact of different farms. Based on the age of the animals, appropriate coefficients are applied to the number of animals for each age category in the breeding: cattle over 2 years old (1 * LSU), cattle aged between 6 months and 2 years (0.6 * LSU) and cattle less than 4 months old (0.4 * LSU) [61]. The final dataset counts 304 records, each standing for a single farm, and a total of 48 input attributes (referring to year 2017) and one target variable, that is the actual number for weaned calves recorded in 2018. All variables represent positive quantities and are described in Table 7.1.

	Attribute	**Reference in Table 5.1**
1	$CATTLE\_SIZE$	3
2	$COWS$	4
3	$HEIFERS$	5
4	F_{CALVES}	6
5	$BULLS$	7
6	M_{CALVES}	8
7	$PERCENT\_FA$	11
8	C_{AGE}	14
9	C_{PAR}	16
10	N_{PAR}	17
11	$SALXGRAV$	24
12	$N_{CORRECT}$	35

Continued on next page

Table 7.1 – continued from previous page		
	Attribute	**Reference in Table 5.1**
13	H_{EASE}	42
14	$H_{DIFFICULT}$	43
15	$H_{CESAREAN}$	44
16	C_{EASE}	45
17	$C_{DIFFICULT}$	46
18	$C_{CESAREAN}$	47
19	C_{N_IND}	48
20	C_{PART_IND}	49
21	H_{PART_IND}	80
22	N_{TFA}	87
23	TFA_{BIRTH}	92
24	TFA_{PAR}	93
25	N_{TFN}	100
26	TFN_{BIRTH}	98
27	TFN_{PAR}	99
28	C_{GRAVID}	119
29	C_{INS}	120
30	$C_{POSTPARTUM}$	121
31	C_{EMPTY}	122
32	LSU	131
33	$LSU1$	132
34	$LSU06$	133
35	$LSU04$	134
36	$INTP$	obtained with 5.2
37	$CONSANG\_NEW$	60
38	$N\_CONSANGNEW$	67
39	$BIRTHW_M$	73
40	$BIRTHW_F$	74
41	$MORT$	101
42	$ABORT$	71
43	N_{ABORT}	72
44	N_{ELIM}	210
45	N_{TOT}	211
46	N_{BALIVE}	217
47	$BORN_{FA}$	214
48	$BORN_{FN}$	215
49	Y	Target Variable 5.1

Table 7.1: Final attributes used in the studied dataset. The last line (variable Y) represents the dependent variable, target attribute for the predictive models generated by ML techniques.

Among the new set of breedings, we compared the reported number of calves that died at birth and the sixtieth day after, as we previously did with the first inspected dataset. (Section 3.2, Figure 3.2).

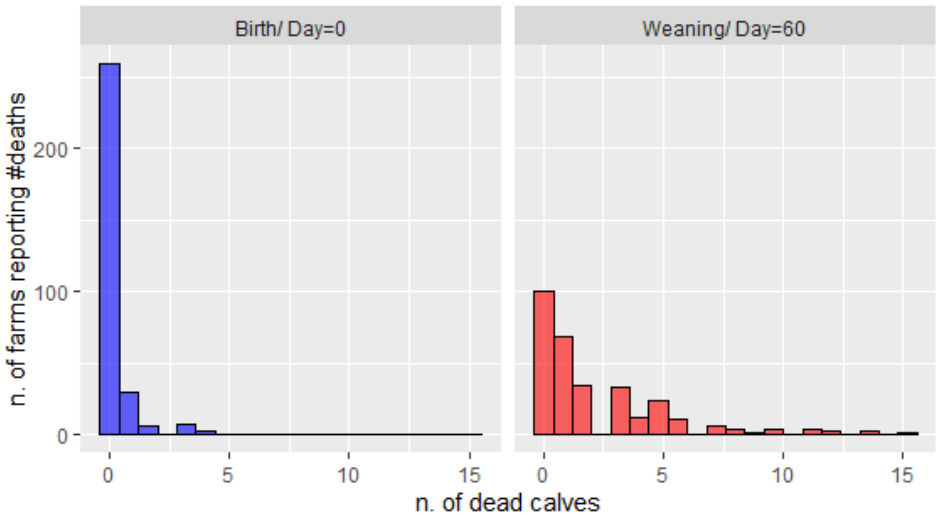


Figure 7.1: Distribution of reported deaths for 304 farms during 2017, respectively at birth and after 60 days. All the breedings show extremely different values between the dead calves at birth (left) and after 60 days (right) (Kruskal-Wallis test: p-value $\ll 0.001$).

As straighforward from Figure 7.1, during birth almost all the farms did not report any deaths, while at the end of weaning the number of farms with zero deaths drops drastically. Data show a high number of dead calves at 60 days, confirming the outcome already observed for all 725 farms inspected in Section 3.2.

7.2.2 The Dataset Preparation and Methods Enrolled in the Study

As previously described, one of the basic steps for applying ML techniques is the subdivision of the dataset into two disjoint parts: the learning set and the test set. Therefore, we split 30 times the dataset in order to obtain 30 different sets, each with constant learning-test partitioning (75%-25%) and randomly selected instances, so as to cover the entire main dataset and avoid retrieving the same instances many times. The learning set was further split, in a way exactly similar to the division into a learning and test set. Each of the 30 learning sets was randomly divided, with a constant partitioning (75% -25%), into a training set and a validation set. The choice of this methodology, i.e., a 75%-25% split repeated for both partitions, is due to the size of the dataset. The division between learning and test entailed a learning set of size equal to 228 instances. We initially considered partitioning the learning set in training-validation through a k-fold cross validation, but the reduced size did not allow us to find a suitable value of k: for example, k smaller than 10 led to a restrained number of training-validation subsets. On the contrary, a value of k greater

than 10, led to a restrained number of records within the validation sets (i.e., fewer than 21 farms). Using a 30-fold cross-validation would imply a validation of size 7, not representative at all. For this reason, we repeated further the subdivision 75%-25% to obtain disjoint training and validation sets, finally checking that the union of all the training sets was equal to the initial learning set (i.e., that no instance had been excluded). Finally, the sizes for testing and learning were respectively 76 and 228, of which 171 and 57 for training and validation. The GPLab package written in Matlab was used. GP is a stochastic algorithm, so the evolved population needs to be evaluated over a portion of the learning set, i.e., a validation set, in order to extract one model. The parameters available in GPLab were set depending on the median error produced by the 30 best models over the validation set. Comparisons with other techniques, listed below, were made on the benchmark problem with the R library caret (Table 7.2). According with the tuning performed for GP, optimal parameters were chosen also in this case by tuning different values specified in a grid for each algorithm. Although differently, all techniques produce one model for each experimental phase on the corresponding training dataset. The solution must thereafter undergo the testing phase to be evaluated for its generalization capabilities. Through the subdivision undertaken, 30 prediction models were obtained for each technique. Parameters were set according to Table 7.3 and 7.4 and, once the models were obtained, the error was evaluated.

Method	**Description**	**Package**
'GP'	Genetic Programming based algoritm (GP)	GPLab library built in Matlab
'knn'	k-Nearest Neighbour algorithm (kNN)	R library caret
'nnet'	Neural Network algorithm (NN)	R library caret
'lm'	Linear regression algorithm (LM)	R library caret
'ranger'	Random Forest Tree-based algorithm (RF)	R library caret

Table 7.2: ML techniques adopted and the respective used package.

Application of ML techniques: Genetic Programming

The parameters set for GP in our study are summarized in Table 7.3. The initial population was generated with the *Ramped half and half method*: half the initial population is constructed using the full method, that generates trees where all the leaves, i.e., the variables, are placed at the same depth. The second half is constructed using the grow method, by creating trees of different sizes and shapes. Among other parameters, it is possible to guarantee the survival of the best individual at each run (*Elitism*) and set the selection method: we decided to set the *lexicographic parsimony pressure*, since this strategy optimizes both fitness and tree size, as fitness is treated as the primary objective and tree size as a secondary objective in a lexicographic ordering. This peculiarity leads to the conservation of the most influential variables over

generations. The algorithm performs, hence, an implicit feature selection and, among all the input variables, only the most relevant are encapsulated in the solutions. In this study, at the end of the evolution process on the training set the population size consisted of 500 members (population size), whereas a single model was then extracted at the end of the run on the validation set. It is necessary to evaluate on the validation dataset the 500 individuals obtained on the training set, and finally evaluate the models on the test set, in order to measure their generalization ability on unseen data. The set F was equal to {`plus; minus; times; mydivide`}.

Parameter	**Description**
ST-GP	
Maximum number of generations	40
Population size	500
Selection Method	Lexicographic Parsimony Pressure
Elitism	Keepbest
Initialization Method	Ramped half and half
Tournament Size	2
Subtree Crossover Rate	0.8
Subtree Mutation Rate	0.1
Subtree Shrinkmutation Rate	0.05
Subtree Swapmutation Rate	0.05

Table 7.3: Parameters used to perform GP.

Application of ML techniques: Linear Model, k-Nearest Neighbour, Neural Network and Random Forest

We compared GP performance with other classical ML approaches used for regression tasks: kNN, NN, LM, and RF. Differently from GP, these methods do not carry out an automatic feature selection and, by the end of the learning process for each run, the final solution already consists of one model (Chapter 4). The corresponding main parameters for all ML approaches are listed in Table 7.4. The unmentioned parameters were kept as default values, since during tuning no tangible improvements in terms of loss function were achieved.

kNN (Subsection 4.3.2) requires a proper value of k: if too small, it could lead to results highly influenced by noise, whereas a large value could be computationally expensive. We used the `knn` package and configured k equal to 15. Greater values generated overfitting models. Concerning NN (Subsection 4.3.5) we set a size of 15 hidden units, fitting a single hidden layer, using the `nnet` package. RF (Subsection 4.3.3) was exploited in its implementation in the `ranger` package, as it disposes of additional useful hyper-parameters. When dealing with a large number of features, it is common to reach greater bias. For each node, the standard

algorithm selects a subset of features as candidates and thereafter splits optimally the nodes and the one yielding the highest score is chosen. The number of features (`mtry`) can be set by the user, whereas the split on the nodes is automatically performed by the algorithm. Due to multiple testing, the standard method rather tends to choose strongly influential variables, masking the effects of moderately influential ones, especially if `mtry` is set to small values. However, irrelevant variables could be selected instead by the algorithm for very small values of `mtry`, resulting in a forest of poorly performing trees. If the dataset contains many relevant predictors, `mtry` should hence be set to larger values. By setting larger values, however, the forest could be weaker, as the single decision trees are more likely to be similar, as the set of candidates could not vary much. Moreover, the splitting rule computes the optimal cut with respect to the variables. If the same variables are constantly chosen, it is possible with high probability that also the cut is performed mostly similarly, leading to overfitting trees. In order to include all variables as candidates for the forest at each node, we set `mtry` equal to the number of variables, whereas the chosen cut-points were fully randomized by setting `extraTrees` parameter. The latter, available in the *ranger* package, drops the attempt to find an optimal cut-point at each node, by determining its value completely randomly. Among all the randomly generated splits, the one yielding the highest score is chosen to split the node. Injecting a higher degree of randomness has the effect of providing the tree with a more efficient generalization ability. To take full advantage of *extremely randomized trees*, the number of random splits (`num.random.splits`) was kept at the default value, i.e., 1. Besides this, instead of learning on bootstrap copies, it is possible to directly grow the trees using the whole training samples, specifying `sample.fraction = 1`. This option was chosen in order to use all instances of the training set (that are already repeated in different training sets, due to the dataset splitting rule illustrated in 7.2.2), and to avoid a decrease in the performance of the model by learning on a lower number of samples. Furthermore, the performance is evaluated over the validation sets, equally for all the applied ML methods, instead of using the *OOB* error.

ML technique	**Parameters**
`knn`	`k = 15`
`nnet`	`size = 15; decay = 0.2`
`lm`	`Intercept = TRUE`
`ranger`	`mtry = 48; splitrule = extratrees; sample.fraction = 1`

Table 7.4: Parameters used to perform ML techniques with caret package in R.

7.3 Results

7.3.1 Interpretability of GP models

The section is dedicated to the discussion of the models obtained with GP. As done in Chapter 6, we analyzed the features selected and the obtained expressions, by considering their interpretability. By repeating the steps exposed in Section 6.3.1, the frequency with which the variables are used by the 30 best models, that is those that have the best fitness on the validation set and that have been evaluated on the test set, were examined. Results are reported in Table 7.5.

Variable	**% of use on 30 runs**	**Variable**	**% of use on 30 runs**
X1 $CATTLE\_SIZE$	27%	X25 N_{TFN}	17%
X2 $COWS$	57%	X26 TFN_{BIRTH}	13%
X3 $HEIFERS$	7%	X27 TFN_{PAR}	20%
X4 F_{CALVES}	3%	X28 C_{GRAVID}	3%
X5 $BULLS$	17%	X29C_{INS}	10%
X6 M_{CALVES}	13%	X30 $C_{POSTPARTUM}$	20%
X7 $PERCENT\_FA$	23%	X31 C_{EMPTY}	17%
X8 C_{AGE}	10%	X32 LSU	7%
X9 C_{PAR}	7%	X33 $LSU1$	20%
X10 N_{PAR}	43%	X34 $LSU06$	7%
X11 $SALXGRAV$	13%	X35 $LSU04$	23%
X12 $N_{CORRECT}$	33%	X36 $INTP$	13%
X13 H_{EASE}	10%	X37 $CONSANG\_NEW$	27%
X14 $H_{DIFFICULT}$	7%	X38 $N\_CONSANGNEW$	17%
X15 $H_{CESAREAN}$	7%	X39 $BIRTHW_M$	7%
X16 C_{EASE}	33%	X40 $BIRTHW_F$	27%
X17 $C_{DIFFICULT}$	7%	X41$MORT$	17%
X18 $C_{CESAREAN}$	0%	X42 $ABORT$	7%
X19 C_{N_IND}	40%	X43 N_{ABORT}	10%
X20 C_{PART_IND}	40%	X44 N_{ELIM}	57%
X21 H_{PART_IND}	50%	X45 N_{TOT}	57%
X22 N_{TFA}	30%	X46 N_{BALIVE}	20%
X23 TFA_{BIRTH}	0%	X47 $BORN_{FA}$	17%
X24 TFA_{PAR}	17%	X48 $BORN_{FN}$	60%

Table 7.5: Percentage of use of each variable among the best 30 individuals found by GP.

Namely, the most frequent variable is the number of calves born from natural inseminations ($BORN_{FN}$), followed by the number of cows ($COWS$), the total number of born calves (N_{TOT}) and the number of calves dead in the first 60 days after birth (N_{ELIM}). In exactly half of the models the EBV referred to calving ease of the heifers was used (H_{PART_IND}). It is straightforward that GP models detected the

majority of information in the aforementioned features. On the contrary, it should be noted that none of the final prediction models included the number of deliveries that required caesarean section for multiparae ($C_{CESAREAN}$) and the mean value of EBV referred to ease of birth of the bulls, which semen has been used on artificial inseminations (TFA_{BIRTH}). The emphasis placed by GP on the listed features entails that the prediction of yearly weaned calves per cow for 2018 depends above all on the quantity of natural inseminations in the farm that is accomplished. It is also dependent on the total number of newborns and calves not weaned during 2017. The result suggests that these variables could be the main features involved in this task. It does not imply, however, that the other parameters are not important in the management of the farm. We thereafter investigated the interpretability of the expressions, considering the fitness obtained in each of the best final models, and taking into account also the number of variables involved in the formula. Considering the results reported in Table 7.6, it is possible to deduce that the model entailing the best predictions on the test set included only three variables (Model 13 in Table 7.6). The expression in infix notation is:

$$Y = \frac{X_{10} + \frac{X_2}{X_{45}}}{X_2 + \frac{X_{45}}{X_2} + \frac{X_{10}}{X_2} + \frac{X_{45}}{X_2 + \frac{X_{10}}{\frac{X_2}{X_{45}} + X_{10}}}}, \tag{7.1}$$

where X2 is the number of cows ($COWS$), X10 is the total number of deliveries occurred during the year in the farm (N_{PAR}) and X45 is the total number of born calves (N_{TOT}). Since these quantities are always summed to and divided by positive quantities in Equation 7.1, the denominators are never null. The *mydivide* operator is actually a division and the model can be reformulated as

$$Y = \left(\frac{X_2 + \frac{X_{45}}{X_2}}{X_{10} + \frac{X_2}{X_{45}}} + \frac{\frac{X_{10}}{X_2}}{X_{10} + \frac{X_2}{X_{45}}} + \frac{\frac{X_{45}(X_2 + X_{45} \cdot X_{10})}{X_2(X_2 + X_{45} \cdot X_{10}) + X_{10} \cdot X_{45}}}{X_{10} + \frac{X_2}{X_{45}}} \right)^{-1}. \tag{7.2}$$

In Equation 7.2 it is possible to notice that the simplification led to an expression containing a sum of three terms. Whenever such a result is reached, the following considerations can be made:

- the obtained expression is given by

$$y = (x_1 + x_2 + \cdots + x_n)^{-1} \tag{7.3}$$

  where y is the result (i.e., the prediction) obtained for the values x_i, $i = 1, \ldots, n$ of the input variables, that is equivalent to $\frac{1}{y} = x_1 + x_2 + \cdots + x_n$.

- By multiplying each term on both sides in the previous expression by y, we complete the standardization process and reach the final expression

$$1 = (y \cdots x_1 + y \cdots x_2 + \cdots + y \cdots x_n) \tag{7.4}$$

  or, equivalently, the more compact expression

$$1 = (\widetilde{x}_1 + \widetilde{x}_2 + \cdots + \widetilde{x}_n). \tag{7.5}$$

- The previous standardization process allows an analysis of the contribution of each component of the linear combination. The boxplots of each component for $i = 1, \ldots, n$ give a visual idea of the distribution of data in the interval [0;1] and statistical tests highlight any difference between them and with respect to the range boundaries.

We hence standardized Equation 7.2, in order to evaluate the contribution of each of the three components isolated in the expression. Following the previous step and renaming n_j the predictions Y obtained for all the instances $j = 1, \ldots, 304$, and $v_{i,j}$ the three components in parenthesis ($i = 1, 2, 3$), Equation 7.2 can be expressed as

$$1 = n_j v_{1,j} + n_j v_{2,j} + n_j v_{3,j}, \tag{7.6}$$

or equivalently

$$1 = \widetilde{v}_{1,j} + \widetilde{v}_{2,j} + \widetilde{v}_{3,j}. \tag{7.7}$$

referring to the rescaled values $n_j \cdot v_{i,j}$ as $\widetilde{v}_{i,j}$.
Since the distributions of $\widetilde{v}_{i,j}$ are not normal (Lilliefors test: $p < 0.05$), the statistical significance was checked with the non-parametric Wilcoxon test with Bonferroni correction (α=0.017) for paired data: all components are significantly different ($p < 0.001$), that is the difference of the mean values is not zero, in particular comparing $\widetilde{v}_{2,j}$ and $\widetilde{v}_{3,j}$. The boxplots for each of them (Figure 7.2) show that the predictions obtained with Equation 7.7 are mainly due to the first addend, that is most of the information is contained in $\widetilde{v}_{1,j}$. Stated otherwise, in Equation 7.2 the corresponding value

$$\frac{X_2 + \frac{X_{45}}{X_2}}{X_{10} + \frac{X_2}{X_{45}}} \tag{7.8}$$

is the part of the individual almost completely defining the value of the prediction. The remaining components play a minor role, with a minimal effect on the performance of the individual obtained, corresponding

to a refinement of the value gained with the main component 7.8.

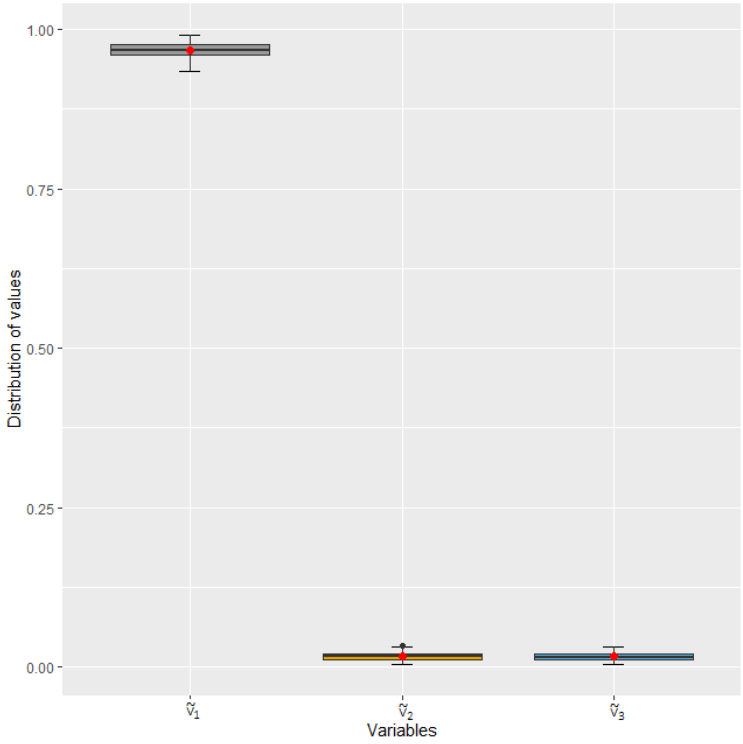


Figure 7.2: Boxplots of the distributions of the variables in Equation 7.7. Wilcoxon test with Bonferroni correction at α=0.017 reported significant differences between the median of the three distributions ($p < 0.001$). The single sample Wilcoxon test, with α =0.05, finally showed for each distribution mean values $\mu \neq 1$ and $\mu \neq 0$ ($p < 0.001$). Mean values are, respectively, μ_1 =0.9671, μ_2 =0.0166 and μ_3 =0.0163 (red dots).

In order to further investigate the mentioned concept and the interpretability of GP models, we focused on a second individual, namely Model 20 in Table 7.6. The model included 3 variables, showing a larger error than equation 7.2. Despite this, the model gains a great interpretability, since the expression released at the end of the run is given by

$$Y = \frac{X_{45}}{X_2 + X_{44}}, \tag{7.9}$$

where X44 is the number of calves that did not survive during the weaning period.
Because of the same reasons entailing the simplification of Equation 7.1 into 7.2, the previous expression leads to the following:

$$Y = \left(\frac{1}{\frac{X_{45}}{X_2}} + \frac{X_{44}}{X_{45}} \right)^{-1}, \tag{7.10}$$

otherwise stated as

$$Y = \left(Calves^{-1} + DeadCalves\right)^{-1}, \tag{7.11}$$

where $Calves$ is the yearly number of calves per cow and the number of calves per cow that do not survive during the weaning period is labelled as $DeadCalves$. As in the previous case, we investigated how the prediction is distributed between the two variables $Calves$ and $DeadCalves$. We performed again the standardization procedure, supporting the analysis with an expression equivalent to 7.10:

$$1 = \widetilde{u}_{1,j} + \widetilde{u}_{2,j}, \tag{7.12}$$

where, for $k = 1, 2$, $\widetilde{u}_{k,j}$ are the rescaled quantities $\widetilde{u}_{k,j} = m_j \cdot u_{k,j}$ and the prediction Y obtained with Model 7.9 are renamed as m_j, whereas the variables $u_{k,j}$ are respectively $Calves^{-1}$ and $DeadCalves$. Performing once again the non-parametric single sample Wilcoxon test, we obtained extremely significant p-values, supporting the hypothesis that the two components $Calves$ and $DeadCalves$ mean values are different respectively from the range boundaries 0 and 1. Both variables are crucial in predicting the output, with more relevance given by $Calves$ (Figure 7.3). As we could entail for the first inspected model, the first component of the Expression 7.10 is the most crucial one in predicting the output, since it assumes values close to the result. However, this second model is also interesting, as the two plotted distributions assume the same complementary behavior. The boxplots in Figure 7.3 visually express the concept.

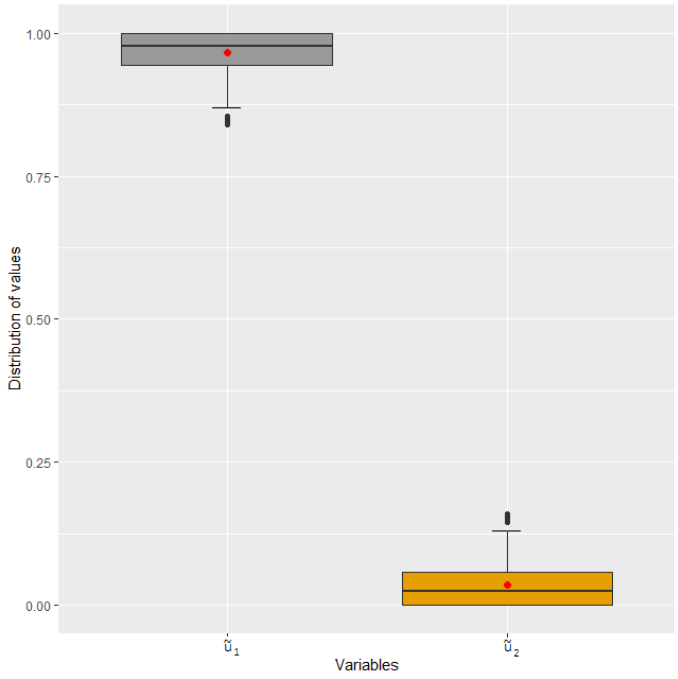


Figure 7.3: Boxplots of the distributions of the variables in Equation 7.12. The single sample Wilcoxon test, with α=0.05, showed for each variable mean values $\mu \neq 1$ and $\mu \neq 0$ ($p < 0.001$).

The lower outliers of the first distribution correspond to farms where cows produce a smaller number of calves. It seems reasonable that a higher portion of calves will not even survive during the weaning period; such cases correspond hence to the upper outliers of the second distribution.

Model	**RMSE on test**	**N. of variables**	**% of variables**	**Model**	**Fitness on test**	**N. of variables**	**% of variables**
model 1	0,1274	9	19%	model 16	0,1946	18	38%
model 2	0,1361	7	15%	model 17	0,1097	10	21%
model 3	0,1480	9	19%	model 18	0,1238	8	17%
model 4	0,0999	13	27%	model 19	0,1373	6	13%
model 5	0,1262	9	19%	model 20	0,1263	3	6%
model 6	0,1263	7	15%	model 21	0,1404	9	19%
model 7	0,1088	6	13%	model 22	0,1242	4	8%
model 8	0,1309	11	23%	model 23	0,1130	8	17%
model 9	0,1330	8	17%	model 24	0,1390	7	15%
model 10	0,1617	12	25%	model 25	0,1385	10	21%
model 11	0,1325	10	21%	model 26	0,1391	6	13%
model 12	0,1370	12	25%	model 27	0,1177	5	10%
model 13	0,0974	3	6%	model 28	0,1222	13	27%
model 14	0,1025	7	15%	model 29	0,1075	10	21%
model 15	0,1328	20	42%	model 30	0,1502	10	21%

Table 7.6: RMSE on the test set, number of involved variables and corresponding percentages are reported for each model evolved by GP in each of the 30 performed runs.

7.3.2 Comparison with other ML techniques

In this section, we compare the performance achieved with the five approaches to prediction taken into consideration. The 30 models obtained by each ML technique were first evaluated on the test set to measure capacity of generalization of each method, analyzing the median fitness. Finally, the best model (i.e., the one that presents the best fitness) was extracted for each technique. We analyzed the fitness distribution over the thirty models, to assess the ability of the models to learn and generalize. In particular, we first commented the results obtained on the learning set and, thereafter, on the test set. Figure 7.4 displays the boxplots of the fitness distribution for each technique. For all statistical tests the significance level was set at $\alpha = 0.05$. The normality of the distributions among all sets was analyzed and Lilliefors test showed a significant deviation from the normal distribution for the results of the LM method ($p = 0.006$). Therefore, in order to compare the performance of the achieved models on the learning sets, a non-parametric test was performed, to asses wheter there is a significant difference between the samples' performace medians. The median values were compared with Kruskal-Wallis test and the null hypothesis that all median values are

equal was rejected ($p < 0.001$). Indeed, all the distributions resulted significantly different according to the Wilcoxon signed-rank test with Bonferroni correction ($\alpha = 0.005$, since there are 10 comparisons), meaning that all performance distributions differ from one another on the learning set (p-values for all considered couples showed $p < 0.001$). Among all the considered methods, as is straightforward from Figure 7.4, the models obtained with RF are indeed the best performing ones in the learning phase, whereas GP produced the least accurate models.

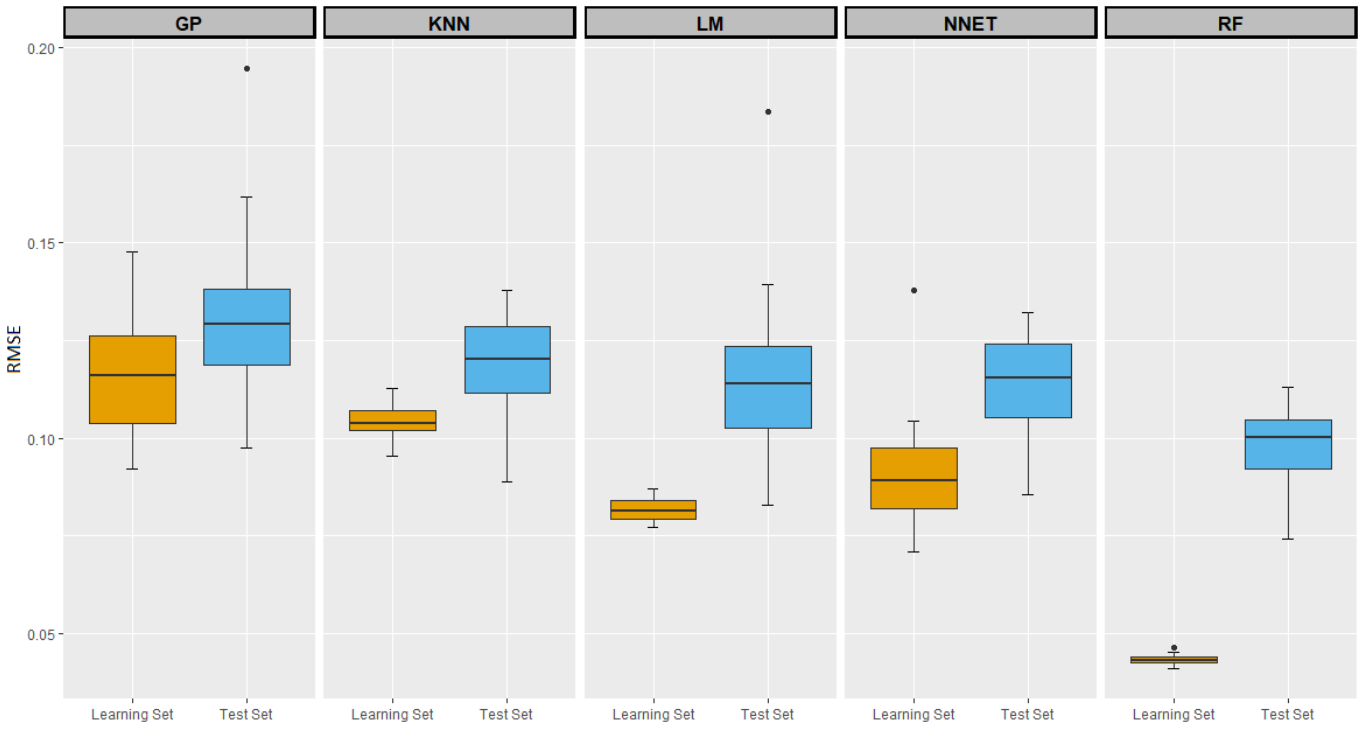


Figure 7.4: RMSE distribution for all the method applied to the 30 subsets. Respectively for each technique, the RMSE among learning set (in yellow) and test set (in blue) sets are shown in boxplots.

The results on the test set were therefore investigated. Predicted values were plotted against the observed data to check their dispersion over the 30 test sets (Figure 7.5(a)). In a supervised learning problem, a predictive model is more accurate as the predicted values are close to the observed ones. In order for the model to be very accurate, the regression line of the scatterplot should tend to overlap the bisector of the plane. For each technique we hence plotted the regression line of all the predicted values versus the observed values on the test sets and compared the coefficients of the line: intercepts and slopes are reported in Figure 7.5(a). All the techniques overestimated target values smaller than ∼0.85 (i.e., the coordinate value of the intersection between the bisector and the regression lines). For values larger than ∼0.85, the models underestimated the target. That means that, indicating with $\overline{x}$ the abscissa of the intersection, the observed values $x < \overline{x}$ were estimated with greater prediction values. On the contrary, for $x > \overline{x}$ the predicted

values are lower than the observed data. The slope of the fitting line obtained with LM is the closest to 1 (β_1=0.613): the predictions follow a linear distribution on each test set by construction and therefore the this results could be expected. Among the other techniques, NNET, GP and kNN, reported slopes β_1=0.417, $\beta_1 = 0.391$ and $\beta_1 = 0.248$ respectively. Finally RF showed a slope equal to 0.002 (β_1) and a corresponding larger value for the intercept ($\beta_0 = 0.856$). Although the latter showed a lower median RMSE compared to the other techniques, two almost symmetrical regions with respect to the bisector were identified, entailing that predictions vary within a fixed interval (0.63;1), also for values outside the previous range. It is clear that the models are not able to generalize. Regarding the corresponding achieved errors, all fitness samples showed normal distributions of the variables (conclusion supported also by the representation of q-q plot in Figure 7.5(b)), and parametric tests were performed.

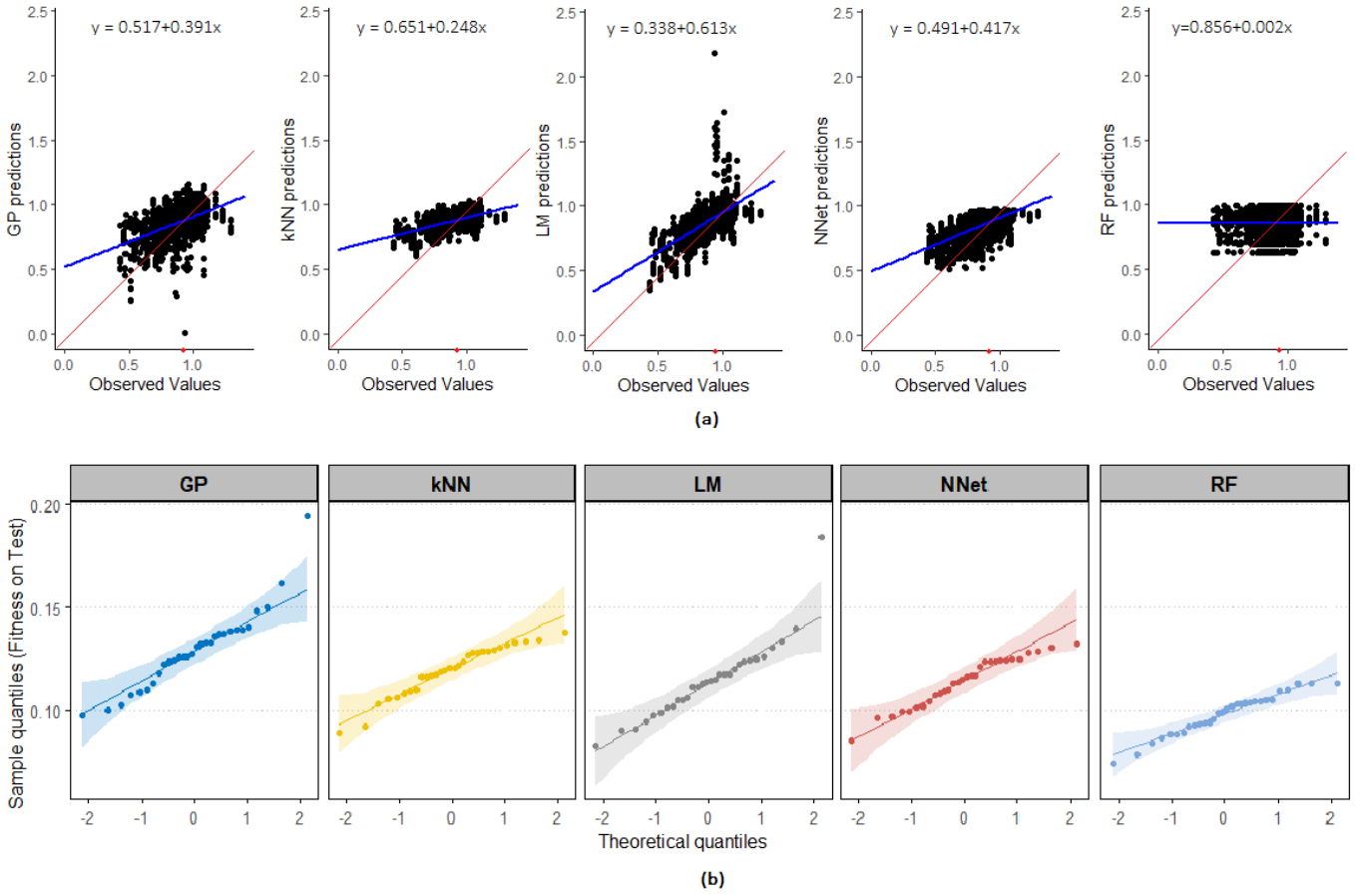


Figure 7.5: (a)Scatterplots for predictions over the test sets. Predicted values over the test set are plotted against the corresponding observed data, for each method on all the 30 test sets. The blue line represents the linear regression fitting line, whereas the red line is the bisector. Corresponding slopes and intercepts are reported for each plot, as well as the corresponding x coordinate (red dot on the abscissa axis).

(b)Q-Q plots for the fitness among the test set. Normality of the of the RMSEs obtained is visually inspected. The quantiles obtained with all the performed techniques of the fitness on the test versus the theoretical ones are plotted. The joint distribution in each case follows the diagonal and is almost entirely contained in the 95% confidence bands.

Since the Levène test did not show any difference between the variance of the distributions ($p = 0.139$), we carried out the one-way ANOVA test: the result was extremely significant, entailing that at least one sample had performace different from the others on average. Finally, the Tukey test with Bonferroni correction was performed, in order to highlight which samples' average performance is actually significantly differentfrom the others. As tangible from the previous boxplots, similar results were achieved on the test set ($p > 0.005$), that is the techniques that showed a lower median fitness on the learning set, revealed a lower median error also on the test set, compared to the other techniques. Moreover, the following pairs of techniques did not show significantly different fitness distributions over the 30 runs: GP-kNN, kNN-NNet and kNN-LM, NNet-LM, stating that the pairs of considered methods performed likewise among the test. What is clear is that, once again, models obtained with RF are the best performing models also on the test, with respect to all other techniques. It is crucial to assess the robustness of the model with respect to its ability to generalize over unseen data. To this purpose, we finally compared the RMSE of each technique on the learning and test sets. Apart from LM results, analyzed with the non-parametric Kruskal-Wallis and Wilcoxon signed-rank tests, all pairs of results for each technique were tested with the Student's t-test. All techniques showed a significant difference between the learning and test results, extremely remarkable for kNN, NNet, LM, and RF ($p < 0.001$). Regarding the results achieved with GP, high significance was also detected comparing the learning and test results ($p = 0.006$).

The statistical tests entail that all the models can achieve good results on unseen instances, in particular the RF algorithm, since it outperformed all other techniques on both the learning and test sets. It is followed by LM, NNet, kNN, with similar results as stated in the previous paragraph, and finally by GP. However, by analyzing the results on the test sets, their ability to generalize tends not to be as accurate as that obtained during the learning phase. In fact, only the application of RF led to significantly better results. It must be considered that, among all methods, GP is the technique that actually produced models that show a median error on the test set that is not too different from the one obtained on the learning set. kNN, NNet, LM and RF can easily perform better and show a lower RMSE (Figure 7.4), since the predictions receive the contributions of all variables. Feature selection is not an intrinsic operation performed by the latter algorithms, unrelated to their structure, and is usually carried out in advance in ML approaches. GP however accomplishes this task. Considering the best model obtained with GP, i.e., the one showing the lowest RMSE analyzed in the previous section (Model 13 in Table 7.6, i.e., Expression 7.2), its performance is comparable to the median behavior obtained with RF models, even incorporating only three variables among the 48 in input, without imposing a priori hypotheses. We also managed to provide a zootechnical interpretation, which would not be possible with black-box techniques. This fact outlines that the different architecture of the evolutionary algorithm can be a good alternative, balancing overfitting issues, whereas

other techniques could slightly be affected. The characteristics of GP outline models that combine few variables, leading to a great interpretability of the formula and allowing further speculations on influential parameters to be made.

7.4 Conclusions

In this study, we investigated more deeply the performance of Piemontese cattle breedings, namely the number of weaned calves per cow produced per year. The sought prediction model had to include relevant factors that describe the weaning period, that is the 60 days after birth. Medium to large farms located in Piedmont were considered. The dataset provided by the ANABORAPI was accurately filtered, imposing some conditions: since the number of involved variables was much greater (we processed 19 variable in the previous study), we extracted records from the biennium 2017-2018, among the most representative farms, i.e., with solid data during all the time lapse between 2014 and 2019. The final dataset consisted of 304 farms and 48 variables, referring to information on cows and artificial inseminations, as well as heifers, natural inseminations and levels of consanguinity of calves resulting from ongoing pregnancies. ML techniques can provide prediction models without making any kind of a priori assumptions. To this purpose, the dataset was divided into a learning and a test set, and the GP approach was inspected further. The GP characteristic, i.e., the provision of well-performing models, that automatically select significant features, let us confirm the considerations exposed in Chapter 6 about the achieved expressions. Whenever a GP model can be expressed as a sum of terms, it is possible to perform an analysis of the standardized equation. We could reconfirm that the first term of the considered sum is the most important one, assuming values close to the output, whereas the other components concurred minimally in the prediction. GP models detected the majority of information in five features, outlining their possibly crucial role in the prediction of the performance of the breeding farm. The number of calves born from natural inseminations is the most significant variable, followed by the number of cows, the total number of born calves, and the number of calves dead in the first 60 days after birth. In exactly half of the models the EBV referred to the calving ease for the primiparae. Comparisons with other classic methods, such as k-Nearest Neighbor, Neural Network, Linear Regression, and Random Forest were made. Compared to other techniques, GP is not the best performing one, considering the median RMSE among 30 runs. On the contrary, RF produces models with the best fitness on the test set. This could be mainly due to the different architecture of the algorithms. On the one side, we handle with classic techniques, producing models that, on average, outperform GP, showing better fitness but complex expressions. On the other side, only GP led to less accurate models in terms of performance, since their error was slightly greater, even if easy to read and interpret. GP can model straightforward expressions, which combine few variables, selected during the evolution process. At the end of the procedure, the best models performed as well as those

obtained with other commonly used techniques, that are however characterized by non-dynamic algorithms as the evolutionary ones. In conclusion, we could assert that GP could represent the most suitable technique, considering all the results in relation to the kind of task dealt with, i.e., the provision of a accessible and interpreatable model, in which variables are automatically selected and combined together. Evolutionary algorithms can be applied on zootechnical data, achieving well-performing models, able to learn from the available data. The results, published in [1], encouraged further investigations, in order to explore the role of other variables in predicting the considered output. In this sector it is common to associate cow-calf problems to genetic and pathological factors, related to pregnancy and childbirth. However, many factors usually considered as marginal: difficult to detect and assert as critical points, the quality of water and air, illumination, the available space and surface, the composition of the food ration could influence the weaning period, being key information about the environment of the farms. Furthermore, comparisons over other time frames are requested. The management of the farm and the choices made by the farmer drag on over time and have delayed effects. It is necessary to analyze the problem, taking into account the data over several years as the learning set, to investigate whether ML techniques could detect crucial factors that did not emerge in this study.

Chapter 8

Towards A Vectorial Approach to predict Beef Farm Performance

8.1 Introduction to Standard vs Vectorial Approaches

In the previous chapters, two general approaches were assumed, in order to test the performance of different ML approaches in the zootechnical field. Attempting to build predictive models for measuring the breeding performance (Chapter 3), GP was chosen inside the ML arena for its interesting characteristics, useful to address the considered problem (Chapter 4). Its behaviour was investigated first among a representative set of farms (Chapters 6) and thereafter among a subset extracted from the latter (Chapters 7), both partitioned between training, validation, and test sets. Furthermore, a distinct number of variables was used among the two performed studies. In both cases, it was possible to simplify the candidate models, to obtain clear and intelligible expressions and analyse the features extracted by the algorithm. On the other hand, from the comparisons illustrated in Chapter 7.8, it is straightforward that other applied techniques, structurally different from each other and from GP, performed better among both the learning and test sets. The reported fitness indicated that a lower error was committed in predicting the target with comparative methods, as different types of structure define the corresponding algorithms. According to this point of view, RF could be classified as the most promising technique. However, the major features offered by GP, i.e., implicit selection of informative predictors and access to the models, do not suggest to discard the method, but to investigate further. In particular, although RF proved to perform better in terms of RMSE minimization, a discrepancy between the error distributions recorded over the learning and test sets was detected, entailing that this method could overfit more easily. On the other hand, GP clearly showed similar fitness distribution for the learning and test sets, suggesting that the method could be able to adapt better to new data.

Since we are interested in these aspects of GP, we investigated deeper the scenario offered by this family of algorithms, to search for possible ways to improve the predictive capacity of the generated models. One of the factors that might be useful to work on consists in splitting the dataset into partitions as independent as possible. To this end, the constraints on the number of breedings we focused on could be set on less stringent thresholds, in order to train models on a larger number of instances. However, this approach would require new investigations about the possible noise introduced with them into the dataset. Reasonably, it would be more beneficial to exhaust the available information, left unexploited up to this point of the study, that is data recorded in the years prior to 2017. These data were only used to determine a pool of representative farms and remained mostly unused. None of the basic methods investigated in the previous Chapters 6 and 7 takes into account a temporal component. Due to their structures, they could only exploit *punctual data* extracted from one year, targeting the following year. To clarify, it is not impossible to deal with past data. The sequences can be split into different observations, in order to maintain the structure of a panel dataset, but the algorithms can not detect the temporal patterns, as in this case the observations would be treated as distinct instances [11, 12]. Of necessity, the strategy entails the loss of valuable information, useful to predict the corresponding target. So far, data from 2017 was used exclusively with targets on 2018. In order to tackle properly the prediction, instead of incorporating the data into a standard panel (see Table 5.3), we encapsulated all the values recorded over the years, for each variable, into vectors (see Table 5.4). Stated otherwise, we introduced the vectorial variables containing data from 2014 to 2017 as input, while targeting the same value among 2018. We opted for this approach since GP was recently developed as VE-GP, offering indeed the possibility to exploit vectors as well as scalars, looking promising as its flexibility allows to tackle many different tasks [11, 12]. VE-GP approach was hence investigated among the breeding farms used in Chapter 7. ST-GP and classic techniques were compared once again, as a different splitting strategy among the dataset was adopted. The outcomes were analysed with respect to VE-GP and LSTM recurrent neural network results, presented in this chapter.

8.2 Matherials and Methods

8.2.1 The Dataset

In Chapter 5 we exposed the main differences among the structure of the standard and vectorial datasets. In Section 5.3 the configuration needed to perform the relevant methods to handle vectorial variables, i.e., VE-GP and LSTM recurrent neural network, was illustrated. Taking into consideration the features that were selected by GP in the two previous chapters, we made some considerations about the number of variables to keep in the new benchmark. Since the results by GP did not improve by incorporating more features, it

was more appropriate to focus on a smaller number of predictors, that can actually be reconducted to the target. In Table 8.1 the final variables are provided for the benchmark.

	Variable Name	**Reference to Table 5.1**
1	$COWS$	4
2	$HEIFERS$	5
3	$INTP$	obtained with 5.2
4	C_{PAR}	16
5	ETA_PART_1	20
6	C_{EASE}	45
7	H_{EASE}	42
8	C_{PART_IND}	49
9	H_{PART_IND}	80
10	TFA_{BIRTH}	92
11	TFA_{PAR}	93
12	$UBA06$	133
13	$UBA04$	134
14	N_{ELIM}	210
15	N_{TOT}	211
16	N_{BALIVE}	217
17	$CORRECT$	35
18	$CONSANG\_NEW$	60
19	Y	Target Variable 5.1

Table 8.1: Final set of variables used for the benchmarked problem. The bottom line represents the dependent variable Y, i.e., the target for the predicted models generated by GP by processing the set of independent variables.

As a greater number of features could become a source of noise, some variables that are actually less informative in predicting the target from an *a posteriori* zootechnical point of view were omitted at this stage, as well as variables partially contained into other similar features. For example, in Chapter 7, both the total number of calves born and the number of births following natural impregnation were used by most GP models. The number of calves born from natural impregnation is already contained in the total number of newborns. Although it was the most frequently used variable, it may be more appropriate to keep only the total number of newborns, by forcing the algorithm to use the latter variable, as informative over all the considered farms (natural impregnation is not performed by all the selected breedings). Prediction of target can be simpler for the algorithms if the useful information is directly provided, resulting easier to be detected. However, ML methods are able to find the necessary source of information also if it is more

complex to extract. Clearly, the task can be easily tackled if some patterns are evident over data. If the information is distributed among other features, the algorithm can detect it anyhow. On the contrary, if no hint is available, the method can not guess the patterns as by magic. The variables 1-19 were stored into two datasets: one containing the data referred to 2017-2018 for the standard approach, and the second one containing the data referred to 2014-2017 for the vectorial approach. In both cases the different partitions intended for training, validation and test refers to the same records, sampled equally on both datasets. A different splitting strategy was adopted with respect to the one previously considered. Indeed, the main idea was to extract a sufficient number of learning instances, in order to perform a k-fold cross validation among it, maintaining at the same time a balanced percentage between learning and test sets (70%-30%). Thereafter, 94 records were extracted to form the test set and the remaining 210 formed the learning set. Among the latter a 7-fold cross validation was imposed, obtaining 7 pairs of training-validation sets, consisting respectively in 180-30 instances. In order to perform a sufficient number of runs of GP and to compare models, the technique was repeated 10 times by selecting the test instances sequentially from the main dataset, restarting from the beginning each time the last record was reached during the selection phase. The learning instances was randomly shuffled before performing the 7-fold sampling.

8.2.2 Standard vs Vectorial Approaches: Experimental Settings

We refer to the standard GP approach with the abbreviation ST-GP to distinguish it from VE-GP. ST-GP and other classic ML approaches, already presented in Chapter 7, were re-performed using the GPLab package built in Matlab and the R library caret. Correspondingly, besides GP , KNN (4.3.2) and NNET (4.3.5) were also tuned, based on the average performance over the validation sets. Concerning linear regression (4.3.1), a *generalized linear model with elastic net regularization* approach `glmnet` was preferred over standard `lm` (Chapter 7). The algorithm fits generalized linear models by means of penalized maximum likelihood, combining the Lasso and Ridge regularizations, using the cyclical coordinate descent (Section 4.3.4). These techniques allow one to accommodate correlation among the predictors, by penalizing less informative variables: Ridge penalty shrinks the coefficients of correlated predictors towards each other, while Lasso tends to pick the most informative ones and discard the others. Compared to standard linear regression, more accurate results are usually expected from its application, as it combines feature elimination from Lasso and feature coefficient reduction from Ridge. The elastic-net penalty is controlled by the parameter α: $\alpha = 0$ is pure Ridge, whereas $\alpha = 1$ is pure Lasso. The overall strength of the penalty for both Ridge and Lasso is controlled by the parameter λ: the coefficients are not regularized if $\lambda = 0$. As λ increases, variables are shrinked towards zero and they are discarded by Lasso regularization, whereas Ridge regularization includes all the variables. Among the vectorial issue, LSTM (4.3.5) was run using the deep learning toolbox imple-

mented in Matlab, whereas the developed GPLab package was used to implement VE-GP [11]. In Table 8.2 and 8.3 the final optimal parameters are summarized.

Parameter	**Description**
ST-GP	
Maximum number of generations	40
Population size	250
Selection Method	Lexicographic Parsimony Pressure
Elitism	Keepbest
Initialization Method	Ramped half and half
Tournament Size	2
Subtree Crossover Rate	0.7
Subtree Mutation Rate	0.1
Subtree Shrinkmutation Rate	0.1
Subtree Swapmutation Rate	0.1
Maxtreedepth	17
VE-GP	
Maximum number of generations	40
Population size	250
Selection Method	Lexicographic Parsimony Pressure
Elitism	Keepbest
Initialization Method	Ramped half and half
Tournament Size	2
Subtree Crossover Rate	0.7
Subtree Mutation Rate	0.3
Mutation of aggregate function parameters	0.2
Maxtreedepth	17

Table 8.2: Parameters used to perform GP.

Regarding ST-GP, we provided the algorithm with a set of primitives F composed of {`plus; minus; times; mydivide`}, as already performed in Chapter 6 and 7. Similarly, we chose proper functions for VE-GP. Differently from ST-GP, suitable functions are indeed provided to manage scalar and vectors [11]. For the considered problem we used {`VSUMW; V_W; VprW; VdivW; V_mean; V_min; V_meanpq; V_minpq`}. The first four operators represent respectively the elementwise sum, difference, product, and the protected division between

two vectors or between a scalar and a vector, e.g. `VSUMW([2,3.5,4,1],[1,0,1,2.5]=[3,3.5,5,3.5])`. The mean and minimum of a vector return the corresponding value for the whole vector (standard aggregate functions `V_mean` and `V_min`), or for a selected range $[p, q]$ inside the vector, where p and q are positive integers with $0 < p \leq q$ (parametric aggregate functions `V_meanpq` and `V_minpq`), e.g. `V_mean([2,3.5,4,1])=2.6`, $\texttt{V\_mean}_{3,4}\texttt{([2,3.5,4,1])=2.5}$. The fact that standard and parametric aggregate functions collapse respectively the whole vectorial variable or a window defined by the parameters into a single value allows to handle all the information contained in the vector or part of it. Besides crossover and mutation, the algoritm is provided with an operator reserved for the mutation of the aggregate function parameters. It allows p and q to evolve in order to detect the most informative window, where to apply thereafter the aggregate function. The set of terminals was composed of the predictors in Table 8.1 for both ST- and VE-GP.

ML technique	**Parameters**
`knn`	`k = 15`
`nnet`	`size = 7; decay = 0.2`
`glmnet`	α `= 0.8,` λ `= 0.85`
LSTM	`hidden units=200; epochs=50; batchsize=1; learning algorithm=adam`

Table 8.3: Parameters used to perform ML techniques with caret package in R and the Deep Learning Toolbox in Matlab

8.3 Results

8.3.1 ST-GP vs VE-GP

Performance of ST-GP and VE-GP were first compared, in order to analyze the behavior of the two algorithms. In Figure 8.3 the median fitness evolution is plotted, based on the following procedure. For each fold within the learning set, a model was selected according to its performance over the validation set. Hence, after 7 runs of GP, 7 models were available, i.e., the ones showing the lowest fitness among the validation. All the 7 best drawn models were evaluated on the whole learning set and the test set, and the median of the 7 models was stored. As the 7-fold was repeated 10 times, 10 median trends were available at the end of the entire evolutionary process. The plot shows the median behaviour of the 10 median fitness achieved for each generation. We initially decided to run the two algorithms for 100 generations. The choice of stopping the evolution after 40 generations was dictated by the overfitting trend recorded among ST-GP. On the contrary, VE-GP proved to be more stable than ST-GP, at least as far as we ran 100 generations. Moreover, the median fitness was overall lower, showing that GP is affected by a remarkable improvement of such a

problem, if temporal information is added, together with proper functions. VE-GP models outperformed the ST-GP ones, stabilizing at lower errors.

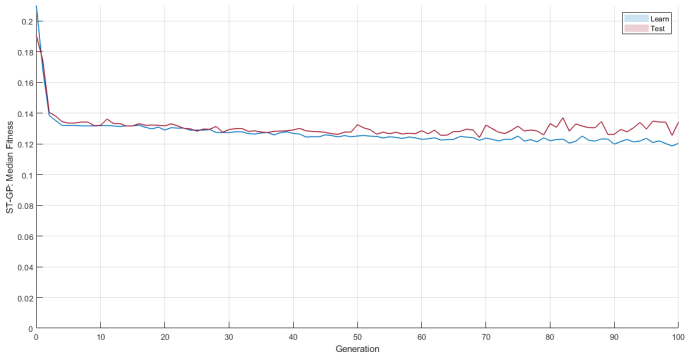


Figure 8.1: ST-GP fitness evolution

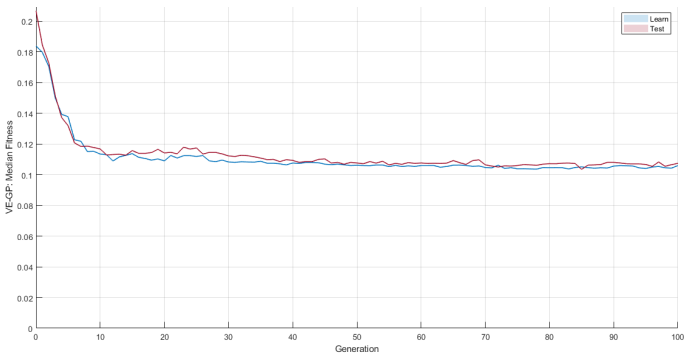


Figure 8.2: VE-GP fitness evolution

Figure 8.3: ST-GP and VE-GP fitness evolution plots. For each generation, the graph plots the median of the 10 median fitness achieved by the best 7 models on the validation sets and correspondingly the performance achieved on the learning and test sets.

We analyzed the predictors encapsulated in the final models by both ST- and VE-GP, selected with respect to the performance achieved on the test sets by running the algorithms for 40 generations. Table 8.4 shows that both methods used 9 variables to tackle the target. However, not the same predictors were used and, above all, not with the same frequency. The number of $COWS$, for example, was highly exploited by both GP algorithms, but all the VE-GP models based the prediction on this feature, whereas only the 70% of ST-GP models found it to be informative. C_{PAR}, on the other hand was used only by ST-GP and in the 50% of solutions, as well as N_{BALIVE} was involved in 60% of them. N_{TOT} was rather exploited only by VE-GP and in 80% of the models. It is evident that, as long as GP is run to predict the target based on the information of a single year, patterns are more difficult to be found and the algorithm (ST-GP) tries to solve the problem by extracting as much information as possible from as many features as possible (7 variables out of 18 were used in more than 20% and at most in 70% of the solutions). When providing temporal information, the search was easier for GP, whose models achieved better fitness, detecting mainly the information based on few predictors (4 out of 18 were exploited in more than the 30% of solutions, and among the four features, one was handled by all the models). Even considering the variables used by each model (Table 8.5), on average 8.4 predictors were used by ST-GP (from 6 to 15), whereas VE-GP built models exploiting 5.5 features on average (from 3 to 9).

Variable	**% of use on 10 runs of ST-GP**	**% of use on 10 runs of VE-GP**
X1 $COWS$	70%	100%
X2 $HEIFERS$	10%	10%
X3 $INTP$	0%	10%
X4 C_PAR	50%	0%
X5 ETA_PART_1	0%	10%
X6 C_{EASE}	0%	10%
X7 H_{EASE}	0%	10%
X8 C_{PART_IND}	0%	0%
X9 H_{PART_IND}	0%	0%
X10 TFA_{BIRTH}	10%	0%
X11 TFA_{PAR}	0%	0%
X12 $UBA06$	0%	0%
X13 $UBA04$	20%	0%
X14 N_{ELIM}	70%	40%
X15 N_{TOT}	0%	80%
X16 N_{BALIVE}	60%	0%
X17 $CORRECT$	30%	0%
X18 $CONSANG\_NEW$	20%	30%

Table 8.4: Frequency of use of each variable among the best 10 individuals found by ST-GP (left column) and VE-GP (right column).

Prediction model	Fitness on test	N. of variables	% of variables
ST-GP			
model 1	0.1335	9	50%
model 2	0.1207	6	33%
model 3	0.1143	11	61%
model 4	0.1383	8	44%
model 5	0.1392	7	39%
model 6	0.1439	7	39%
model 7	0.1395	8	44%
model 8	0.1370	6	33%
model 9	0.1285	15	83%
model 10	0.1184	7	39%
VE-GP			
model 1	0.1117	5	26%
model 2	0.1016	3	16%
model 3	0.1044	9	47%
model 4	0.1085	8	42%
model 5	0.1134	3	16%
model 6	0.0998	8	42%
model 7	0.1018	4	21%
model 8	0.1149	4	21%
model 9	0.0999	8	42%
model 10	0.1121	3	16%

Table 8.5: Fitness on the test set, number of involved variables and corresponding percentage for each model evolved by ST-GP (upper table) and VE-GP (lower table) in each of the 10 runs

8.3.2 General Comparisons With Other ML Methods

We compared GP behavior with the other methods listed in the previous section. As already explained, besides ST-GP, also KNN, NNET, and GLMNET exploited the information on 2017 with target in 2018, whereas LSTM was involved as VE-GP to process vectorial variables 2014-2017 and target 2018. Results reported in 8.3.1 about ST-GP compared to VE-GP are supported also by the corresponding boxplots in Figure 8.4. The median and mean RMSEs values are reported in Table 8.6.

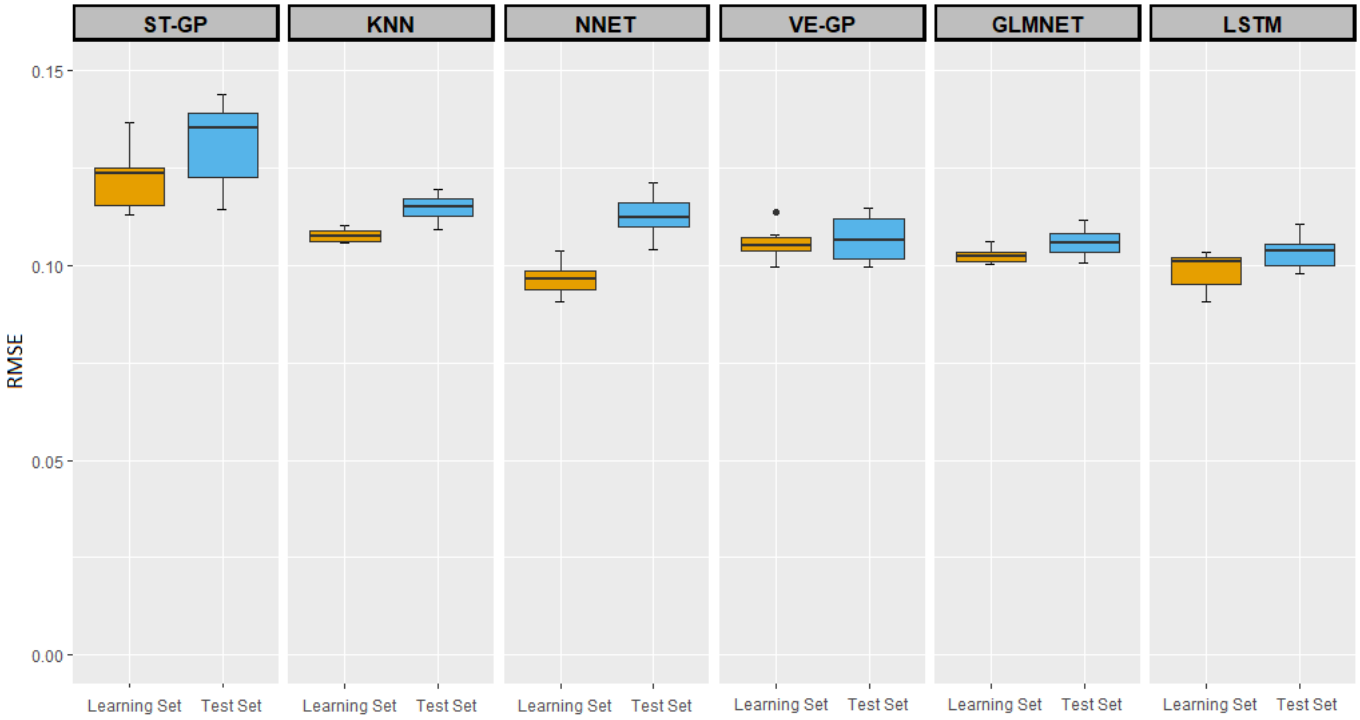


Figure 8.4: RMSE on both the learning and test set for the different algorithms. Learning results are plotted in yellow (left) and test results are plotted in blue (right) for each technique

	STGP	**KNN**	**NNET**	**VEGP**	**GLMNET**	**LSTM**
Results among the learning sets						
Median	0.1238	0.1074	0.09671	0.10516	0.1025	0.10112
Mean	0.1220	0.1077	0.09666	0.10535	0.1025	0.09884
Results among the test sets						
Median	0.1353	0.1151	0.1122	0.1065	0.1057	0.10372
Mean	0.1314	0.1147	0.1128	0.1068	0.1056	0.10337

Table 8.6: Median and mean RMSE of the different techniques among the learning and test sets

The Kruskal-Wallis nonparametric test, performed for all the considered methods with a significance level of α= 0.05, was applied to investigate the RMSE achieved among the learning set and the test sets separately. The resulting p-values ($p \ll 0.001$) showed extremely significant differences in median performance between the methods, considering both stages. The pairwise Wilcoxon tests provided with Bonferroni correction $\alpha = 0.05/15 = 0.0033$ was hence performed among all compared techniques. Among the learning set, STGP was significantly different from all other methods, resulting in a poor performance. Similarly, KNN resulted significantly different with respect to both NNET and LSTM, as well as the comparison between VEGP and LSTM. Concerning the RMSE achieved on the test sets, STGP performed poorly with respect to other

methods, showing greater, significantly different, values for the RMSE on average. On the contrary, GLMNET performance was significantly better than KNN and NNET, as well as LSTM compared respectively to KNN and VEGP. As a consequence, the following pairs of methods did not show significantly different performance: VEGP-KNN, VEGP-NNET, VEGP-GLMNET, VEGP-LSTM, likewise the pair LSTM and GLMNET.

In order to discuss the quality of predictions, the quality of learning should also be analyzed for the different techniques. Indeed, we compared learning and test fitness distributions obtained by the single methods, in order to determine the occurrence of overfitting. Wilcoxon signed rank test showed that the two distributions for KNN and the two obtained with NNET were different, since the corresponding p-values were extremely significant (Wilkoxon: $p \ll 0.001$), as well as the median RMSE (Kruskal-Wallis test: $p \ll 0.001$). Concerning ST-GP, the two distributions and the median error were slightly different (Wilcoxon and Kruskal Wallis: p-values equal to 0.048 and 0.034 respectively). GLMNET showed the same learning and test fitness distributions, but different median RMSE (Wilcoxon: $p > 0.05$; Kruskal Wallis: $p = 0.041$), whereas LSTM achieved different distributions with similar median. VE-GP was the only method that produced the same fitness distributions with the same median among the learning and test set.

In short, considering all the results from the statistical tests, ST-GP produced less accurate models and all the other methods outperformed ST-GP. However, among the different techniques, KNN and NNET clearly overfitted, generating good results while training, but losing their ability to generalize on the test set. On the contrary, VE-GP, GLMNET, and LSTM produced better and statistically similar results, as the RMSEs that were achieved in the attempt of predicting the target among the test sets were not significantly different across the methods. In particular, LSTM produced the best fitness considering both learning and test sets results. However, VE-GP was the only method showing the same distribution among learning and test sets, highlighting its ability to generalize better over unseen data. From these outcomes it was possible to recall the importance of introducing the temporal information in the form of vectors, to improve the accuracy of predictions among the considered problem.

8.4 Conclusions

Chapter 8 was dedicated to the inspection of GP behaviour when predicting a target starting from datasets that, in one case, were exclusively formed by scalar values (treated hence with ST-GP) and, in the other, assumed a vector representation (handled with VE-GP). This representation is quite useful for incorporating temporal patterns or, in general, successive collections of data for single variables among the same candi-

date. Indeed, with the common representation through standard data frames such patterns are usually not recognizable and the performance of the models do not improve. On the contrary, if the data are organized into a vectorial dataset the algorithm receives temporal information in input. Thereby, by means of proper functions able to manage vectors, it can produce more accurate predictions. First of all, the dataset was prepared to deal with the vector-based representation. The datasets, sharing the same scalar target from 2018 (i.e., the quota of weaned calves per cow per year), were prepared extracting the data among 2017 and among the whole period 2014-2017, based on a previously defined set of farms. In this study, a different splitting rule was defined among the datasets with respect to previous chapters. Learning and test sets were selected respecting the proportion 70%-30%, and thereafter learning sets, randomly reshuffled, were split according to a 7-fold cross validation technique. Prediction models were constructed with different GP algorithms, ST- and VE-GP first, that were thereafter compared with other ML methods. In particular, VE-GP was compared with LSTM, that considers the time relationship among the data.

The main goal was hence to inspect the ability of VE-GP with respect to ST-GP in predicting the target. The developed algorithm could produce better results, by achieving lower RMSEs among both learning and test sets. We analyzed first the evolution of the median fitness observed on the learning and test sets, and clearly VE-GP proved to be more stable, evolving a population through more generations without giving sign of overfitting, whereas ST-GP showed the "symptom" quite soon, considering similar experimental settings. Besides, VE-GP reached better results by encapsulating fewer variables in each extracted candidate model, and detecting the information to greater extent mostly from specific features. VE-GP still yelds a good interpretability of the solutions, by giving access to the formula and to the features implicitly selected, providing meaningful information about the tackled issue. Being able to extract important features among the predictors in form of vectors, the algorithm improved the target forecast. VE-GP turned out to outperform not only ST-GP, but also other techniques used in the field of ML. Although VE-GP performance is similar to LSTM and GLMNET (the latter exploiting the standard data representation), it was the only method that did not show a significantly different behaviour on the learning and test sets. The two distributions and their median are similar, entailing that VE-GP provides a good response in terms of generalization ability on unseen data. Improvements can be expected by feeding the algorithm with larger datasets, by providing more candidates and longer vectors.

Chapter 9

Exploring New Features to Predict Beef Farm Performance

9.1 Introduction

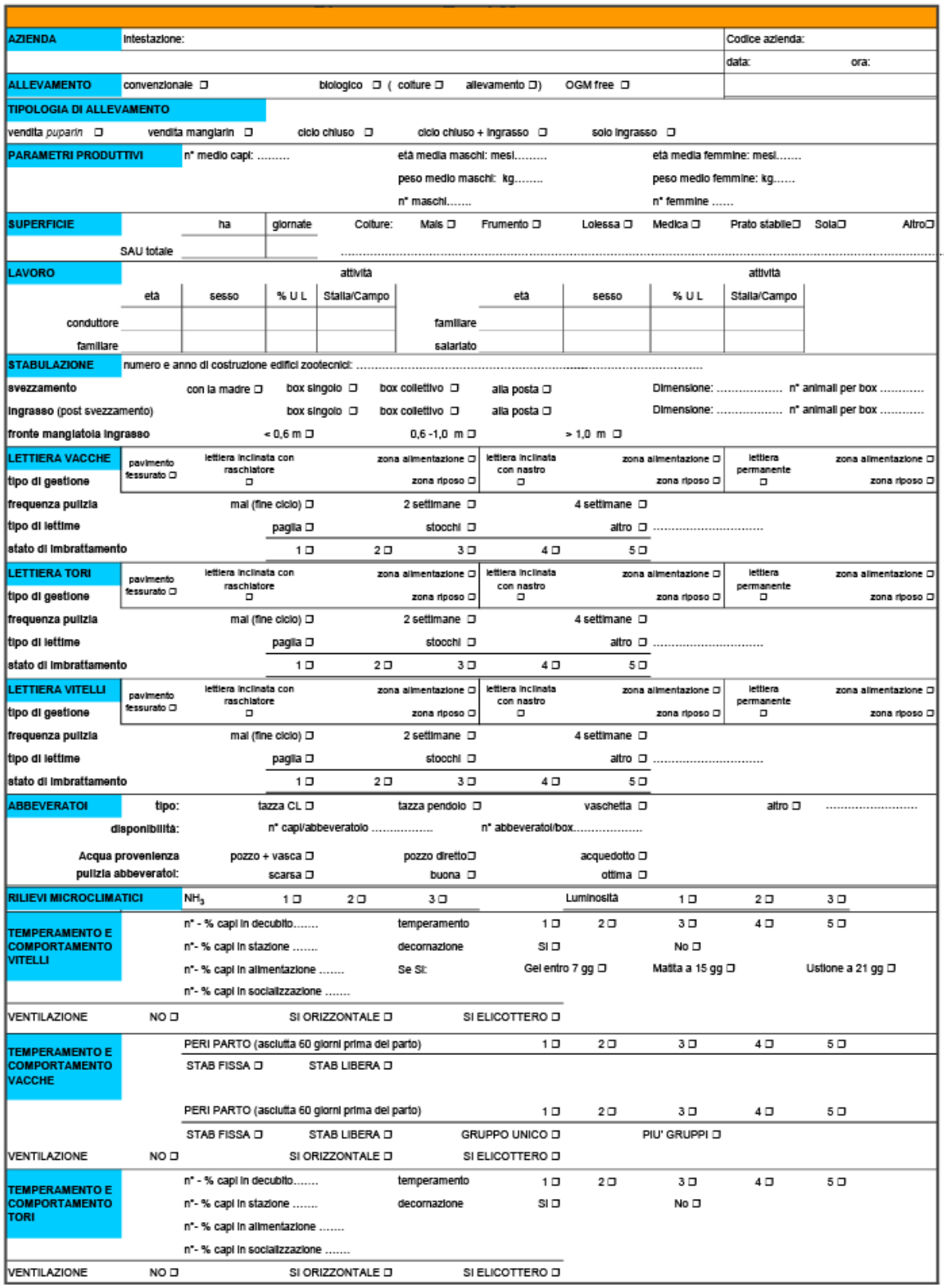
In Chapter 3 the fact that many factors related to the farm management affect the performance of the breeding was highlighted, in order to define the performance of the farm, in the context of Piemontese cattle breedings. The availability of informative tools makes it possible to streamline technical issues and to improve the economic aspects. All the stages of the productive process can indeed be planned in details at each stage. By keeping updated data, it is possible to monitor variations that could cause undesired effects or, on the contrary, that could promote improvements. The summary data used so far in the study provided useful information to predict the target identified to forecast the breeding performance. The predictors selected, handled with different ML techniques in order to build the models, were extrapolated from the Herd-Book, as this is the available source of information. Investigations on the problem were performed exploiting two representations: in the first, summary data referring to a single year were used, and subsequently, in the other, the entire data collection from the previous years was incorporated in the benchmark. The approach produced several promising results, highlighting the potential of ML techniques. However, many aspects of farming are not yet directly represented by data. Certain aspects such as nutrition, environmental conditions, animal welfare, are measured indirectly by monitoring the report data in 3.1. Changes in the management or problems affecting some aspects entail variations which can become evident only after a long time. From a technical point of view, it is essential to operate in such a way as to ensure maximum health and well-being to animals, criteria increasingly linked to the commercial and economic aspect. The predictive methods investigated have a notable strength, namely the possibility of using any type of variable. It may therefore be appropriate to obtain a more detailed representation of parameters that may negatively or positively

influence production, in order to be subsequently used for the forecast of the breeding performance. Indeed, this information is not recorded in the databases. The critical points on which the efficiency of the breeding depends are mainly the average number of calves per year per cow and mortality. Besides this, the weaning of calves contribute substantially to these parameters, depending on multiple factors such as the environment, the quality of air and environment, feeding type, and it is necessary to focus on many issues such as the body conditions, the supply of essential nutrients, the space available for the animal, which influence the costs. Therefore we advanced the idea of collecting new data, which could contribute to better results in terms of prediction, as well as being able to provide a more precise representation of the farm anyway. The chapter was dedicated to the description of the tools provided to collect the additional data, i.e., with an on field survey. Thereafter the information was coded and stored into a database, subsequently analyzed, in order to inspect their role in the prediction of weaned calves per cow per year. Predictors were selected and ML techniques, already applied in the previous benchmarks, were involved once more, to extrapolate patterns and considerations were presented.

9.2 Materials and Methods

9.2.1 The Draft of the Questionnaire

The main goal consisted in drawing a survey to collect and codify useful factors, in order to find further important parameters to model farm management conditions. Indeed visits to farms were planned, in order to fill on field the questionnaire and collect the additional data. No repeated visits were scheduled, for logistical and temporal reasons. First of all it was necessary to highlight all the possible aspects to be recorded, trying not to neglect any useful information. Therefore, we took a cue from a previous version of the questionnaire, designed for fattening farms of Piemontese cattle [56]. The form had been organized to focus on fattening breeding farms and corresponding items were related to the size of the farm, the administered ration, and some technopathies, all referred to fattening calves. Given that the farms we selected for data collection are breeding farms, and that only part of them fatten, we extended the survey with additional information such as the type of farm, vaccination plan, and description for litter, feeding type, and ration. Besides this, we set the codes to describe the type of ventilation, the cleaning of the water trough, the micro-climatic reliefs (light and presence of ammonia), and the temperament of the animals. The time devoted to understanding the questionnaire and the involved variables was fundamental to focus on the aspects of breeding to be acquired. In particular, the visits to farms represented an important training phase, quite a useful step to better understand the dynamics related to farm management.



AZIENDA | Intestazione: | Codice azienda:
data: ora:

ALLEVAMENTO | convenzionale ❐ | biologico ❐ (colture ❐ allevamento ❐) | OGM free ❐

TIPOLOGIA DI ALLEVAMENTO
vendita *puparin* ❐ | vendita manglarin ❐ | ciclo chiuso ❐ | ciclo chiuso + ingrasso ❐ | solo ingrasso ❐

PARAMETRI PRODUTTIVI | n° medio capi: ..... | età media maschi: mesi..... | età media femmine: mesi.....
peso medio maschi: kg..... | peso medio femmine: kg.....
n° maschi..... | n° femmine .....

SUPERFICIE | ha | giornate | Colture: Mais ❐ Frumento ❐ Loiessa ❐ Medica ❐ Prato stabile❐ Sola❐ Altro❐
SAU totale

LAVORO

	età	sesso	% U L	Stalla/Campo		età	sesso	% U L	Stalla/Campo
conduttore					familiare				
familiare					salariato				

STABULAZIONE | numero e anno di costruzione edifici zootecnici: .....
svezzamento | con la madre ❐ | box singolo ❐ | box collettivo ❐ | alla posta ❐ | Dimensione: ..... n° animali per box .....
ingrasso (post svezzamento) | box singolo ❐ | box collettivo ❐ | alla posta ❐ | Dimensione: ..... n° animali per box .....
fronte mangiatoia ingrasso | < 0,6 m ❐ | 0,6 -1,0 m ❐ | > 1,0 m ❐

LETTIERA VACCHE	pavimento fessurato ❐	lettiera inclinata con raschiatore ❐	zona alimentazione ❐	zona riposo ❐	lettiera inclinata con nastro ❐	zona alimentazione ❐	zona riposo ❐	lettiera permanente ❐	zona alimentazione ❐	zona riposo ❐
tipo di gestione										

frequenza pulizia | mai (fine ciclo) ❐ | 2 settimane ❐ | 4 settimane ❐
tipo di lettime | paglia ❐ | stocchi ❐ | altro ❐ .....
stato di imbrattamento | 1 ❐ | 2 ❐ | 3 ❐ | 4 ❐ | 5 ❐

LETTIERA TORI	pavimento fessurato ❐	lettiera inclinata con raschiatore ❐	zona alimentazione ❐	zona riposo ❐	lettiera inclinata con nastro ❐	zona alimentazione ❐	zona riposo ❐	lettiera permanente ❐	zona alimentazione ❐	zona riposo ❐
tipo di gestione										

frequenza pulizia | mai (fine ciclo) ❐ | 2 settimane ❐ | 4 settimane ❐
tipo di lettime | paglia ❐ | stocchi ❐ | altro ❐ .....
stato di imbrattamento | 1 ❐ | 2 ❐ | 3 ❐ | 4 ❐ | 5 ❐

LETTIERA VITELLI	pavimento fessurato ❐	lettiera inclinata con raschiatore ❐	zona alimentazione ❐	zona riposo ❐	lettiera inclinata con nastro ❐	zona alimentazione ❐	zona riposo ❐	lettiera permanente ❐	zona alimentazione ❐	zona riposo ❐
tipo di gestione										

frequenza pulizia | mai (fine ciclo) ❐ | 2 settimane ❐ | 4 settimane ❐
tipo di lettime | paglia ❐ | stocchi ❐ | altro ❐ .....
stato di imbrattamento | 1 ❐ | 2 ❐ | 3 ❐ | 4 ❐ | 5 ❐

ABBEVERATOI | tipo: | tazza CL ❐ | tazza pendolo ❐ | vaschetta ❐ | altro ❐ .....
disponibilità: | n° capi/abbeveratoio ..... | n° abbeveratoi/box.....
Acqua provenienza | pozzo + vasca ❐ | pozzo diretto❐ | acquedotto ❐
pulizia abbeveratoi: | scarsa ❐ | buona ❐ | ottima ❐

RILIEVI MICROCLIMATICI | NH_3 | 1 ❐ | 2 ❐ | 3 ❐ | Luminosità | 1 ❐ | 2 ❐ | 3 ❐

TEMPERAMENTO E COMPORTAMENTO VITELLI
n° - % capi in decubito..... | temperamento | 1 ❐ | 2 ❐ | 3 ❐ | 4 ❐ | 5 ❐
n°- % capi in stazione ..... | decornazione | SI ❐ | No ❐
n°- % capi in alimentazione ..... | Se SI: | Gel entro 7 gg ❐ | Matita a 15 gg ❐ | Ustione a 21 gg ❐
n°- % capi in socializzazione .....

VENTILAZIONE | NO ❐ | SI ORIZZONTALE ❐ | SI ELICOTTERO ❐

TEMPERAMENTO E COMPORTAMENTO VACCHE
PERI PARTO (asciutta 60 giorni prima del parto) | 1 ❐ | 2 ❐ | 3 ❐ | 4 ❐ | 5 ❐
STAB FISSA ❐ | STAB LIBERA ❐

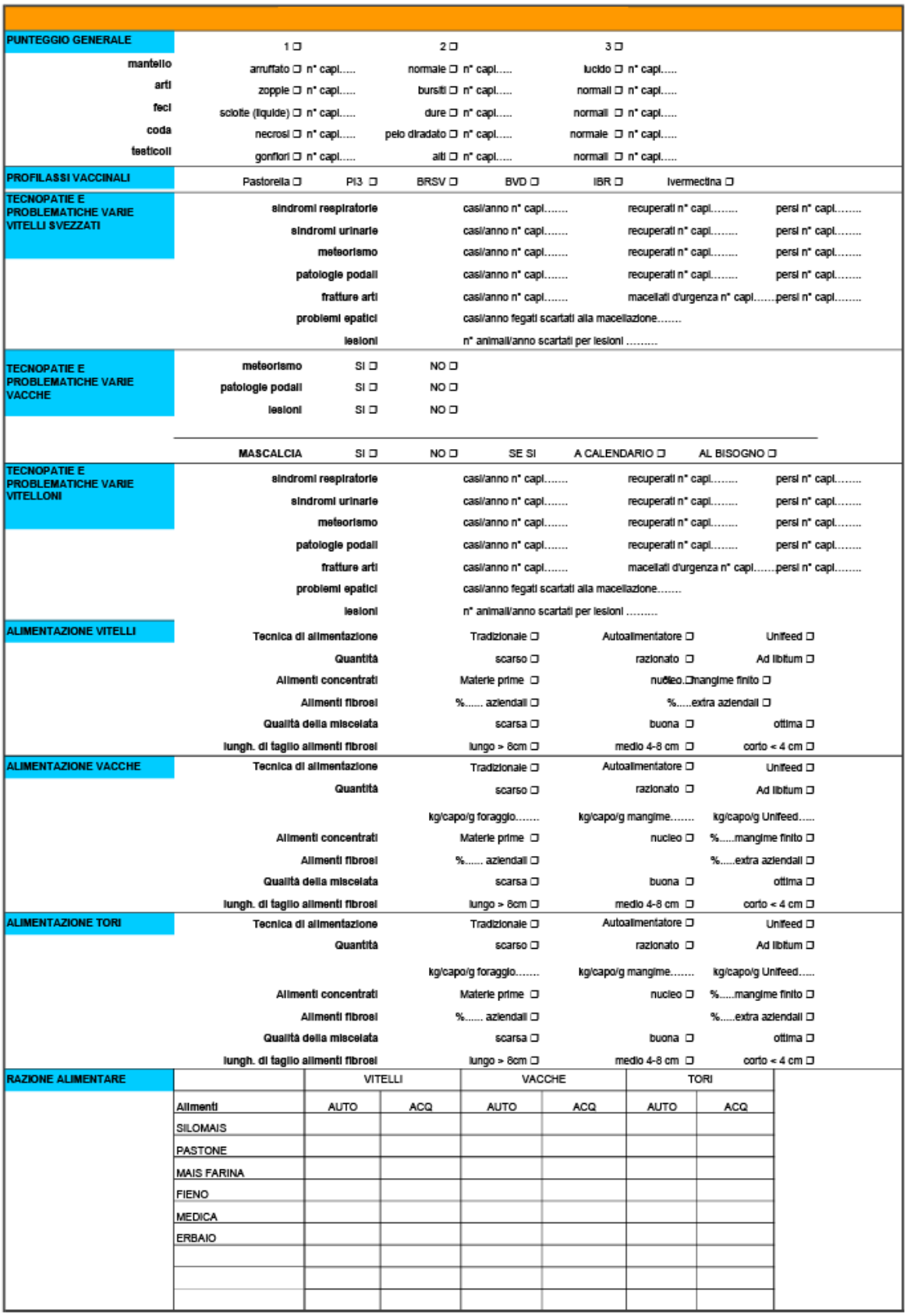
PERI PARTO (asciutta 60 giorni prima del parto) | 1 ❐ | 2 ❐ | 3 ❐ | 4 ❐ | 5 ❐
STAB FISSA ❐ | STAB LIBERA ❐ | GRUPPO UNICO ❐ | PIU' GRUPPI ❐

VENTILAZIONE | NO ❐ | SI ORIZZONTALE ❐ | SI ELICOTTERO ❐

TEMPERAMENTO E COMPORTAMENTO TORI
n° - % capi in decubito..... | temperamento | 1 ❐ | 2 ❐ | 3 ❐ | 4 ❐ | 5 ❐
n°- % capi in stazione ..... | decornazione | SI ❐ | No ❐
n°- % capi in alimentazione .....
n°- % capi in socializzazione .....

VENTILAZIONE | NO ❐ | SI ORIZZONTALE ❐ | SI ELICOTTERO ❐

Figure 9.1: On field survey, designed to collect data related to environment, feeding type, ration, litter, vaccination and technopaties, and temperament.



PUNTEGGIO GENERALE	1 ❐		2 ❐		3 ❐					
mantello	arruffato ❐	n° capi.....	normale ❐	n° capi.....	lucido ❐	n° capi.....				
arti	zoppie ❐	n° capi.....	bursiti ❐	n° capi.....	normali ❐	n° capi.....				
feci	sciolte (liquide) ❐	n° capi.....	dure ❐	n° capi.....	normali ❐	n° capi.....				
coda	necrosi ❐	n° capi.....	pelo diradato ❐	n° capi.....	normale ❐	n° capi.....				
testicoli	gonfiori ❐	n° capi.....	alti ❐	n° capi.....	normali ❐	n° capi.....				
PROFILASSI VACCINALI	Pastorella ❐	PI3 ❐	BRSV ❐	BVD ❐	IBR ❐	Ivermectina ❐				
TECNOPATIE E PROBLEMATICHE VARIE VITELLI SVEZZATI	sindromi respiratorie		casi/anno n° capi.....		recuperati n° capi.....		persi n° capi.....			
	sindromi urinarie		casi/anno n° capi.....		recuperati n° capi.....		persi n° capi.....			
	meteorismo		casi/anno n° capi.....		recuperati n° capi.....		persi n° capi.....			
	patologie podali		casi/anno n° capi.....		recuperati n° capi.....		persi n° capi.....			
	fratture arti		casi/anno n° capi.....		macellati d'urgenza n° capi.....		persi n° capi.....			
	problemi epatici		casi/anno fegati scartati alla macellazione.....							
	lesioni		n° animali/anno scartati per lesioni							
TECNOPATIE E PROBLEMATICHE VARIE VACCHE	meteorismo	SI ❐	NO ❐							
	patologie podali	SI ❐	NO ❐							
	lesioni	SI ❐	NO ❐							
	MASCALCIA	SI ❐	NO ❐	SE SI	A CALENDARIO ❐	AL BISOGNO ❐				
TECNOPATIE E PROBLEMATICHE VARIE VITELLONI	sindromi respiratorie		casi/anno n° capi.....		recuperati n° capi.....		persi n° capi.....			
	sindromi urinarie		casi/anno n° capi.....		recuperati n° capi.....		persi n° capi.....			
	meteorismo		casi/anno n° capi.....		recuperati n° capi.....		persi n° capi.....			
	patologie podali		casi/anno n° capi.....		recuperati n° capi.....		persi n° capi.....			
	fratture arti		casi/anno n° capi.....		macellati d'urgenza n° capi.....		persi n° capi.....			
	problemi epatici		casi/anno fegati scartati alla macellazione.....							
	lesioni		n° animali/anno scartati per lesioni							
ALIMENTAZIONE VITELLI	Tecnica di alimentazione		Tradizionale ❐		Autoalimentatore ❐		Unifeed ❐			
	Quantità		scarso ❐		razionato ❐		Ad libitum ❐			
	Alimenti concentrati		Materie prime ❐		nucleo.❐mangime finito ❐					
	Alimenti fibrosi		%..... aziendali ❐		%.....extra aziendali ❐					
	Qualità della miscelata		scarsa ❐		buona ❐		ottima ❐			
	lungh. di taglio alimenti fibrosi		lungo > 8cm ❐		medio 4-8 cm ❐		corto < 4 cm ❐			
ALIMENTAZIONE VACCHE	Tecnica di alimentazione		Tradizionale ❐		Autoalimentatore ❐		Unifeed ❐			
	Quantità		scarso ❐		razionato ❐		Ad libitum ❐			
			kg/capo/g foraggio.....		kg/capo/g mangime.....		kg/capo/g Unifeed.....			
	Alimenti concentrati		Materie prime ❐		nucleo ❐		%.....mangime finito ❐			
	Alimenti fibrosi		%..... aziendali ❐				%.....extra aziendali ❐			
	Qualità della miscelata		scarsa ❐		buona ❐		ottima ❐			
	lungh. di taglio alimenti fibrosi		lungo > 8cm ❐		medio 4-8 cm ❐		corto < 4 cm ❐			
ALIMENTAZIONE TORI	Tecnica di alimentazione		Tradizionale ❐		Autoalimentatore ❐		Unifeed ❐			
	Quantità		scarso ❐		razionato ❐		Ad libitum ❐			
			kg/capo/g foraggio.....		kg/capo/g mangime.....		kg/capo/g Unifeed.....			
	Alimenti concentrati		Materie prime ❐		nucleo ❐		%.....mangime finito ❐			
	Alimenti fibrosi		%..... aziendali ❐				%.....extra aziendali ❐			
	Qualità della miscelata		scarsa ❐		buona ❐		ottima ❐			
	lungh. di taglio alimenti fibrosi		lungo > 8cm ❐		medio 4-8 cm ❐		corto < 4 cm ❐			

RAZIONE ALIMENTARE	VITELLI		VACCHE		TORI	
Alimenti	AUTO	ACQ	AUTO	ACQ	AUTO	ACQ
SILOMAIS						
PASTONE						
MAIS FARINA						
FIENO						
MEDICA						
ERBAIO						

Figure 9.2: On field survey, designed to collect data related to environment, feeding type, ration, litter, vaccination and technopaties, and temperament.

9.2.2 The Dataset and Experimental Settings

The farms were selected on the basis of their location, the size (i.e., number of cows larger than 30), and the willingness of the farmer to dedicate the time necessary to provide the information, through interviews and farm visits. The survey was filled between March and April in 2019. Although collected in 2019, this information well represents the breeding for the previous year, i.e., 2018, as no changes were made to the management in any breeding during the first months of 2019. The dataset was hence provided first with the Herd-Book variables, as already done in the previous chapters. However, the farms were not selected based on the percentage of performed artificial insemination. The final pool consisted of breedings, showing heterogeneous situations. All of them were representative in terms of number of heads and constant data recordings. The corresponding Herd-Book variables are listed in Table 9.1, and were extracted from year 2018. The target was hence drawn from 2019.

	Variable Name	**Reference to Table 5.1**
1	$CATTLE\_SIZE$	3
2	$COWS$	4
3	$HEIFERS$	5
4	$INTP$	obtained with 5.2
5	C_{PAR}	16
6	$SALXGRAV$	24
7	H_{EASE}	42
8	C_{PART_IND}	49
9	$UBA04$	134
10	N_{ELIM}	210
11	N_{TOT}	211
12	N_{BALIVE}	217
13	$CORRECT$	35
14	$CONSANG\_NEW$	60
15	$PERCENT\_FA$	11
16	Y	Target Variable 5.1

Table 9.1: Final set of variables used for the benchmarked problem. The bottom line represents the dependent variable Y, i.e., the target for the predicted models generated by GP based on the set of independent variables.

From the questionnaire, a total number of 201 features was defined. Given this high number of variables, we opted for a preventive corresponding feature selection. To understand their possible link with the target, taking into account the relevance of the variables contained in the survey from a zootechnical point of view and the actual correlation with the target (quota of weaned calves per cow recorded for 2019), selected

22 predictors, in order to contribute additional data to the Herd-Book ones. The correlation between the variables and the target was considered. Features correlated more than 30%, positively and negatively, were finally selected (Table 9.2).

Variable Name	**Correlation with target**
Cows temperament after delivery	0.456
Manger front <0.6m	0.454
Grain	0.394
Self-produced flour (cows)	0.387
Slatted floor (cows)	0.368
Maximum number of calves per box	0.362
Slatted floor (calves)	0.358
Multiple groups (calves)	0.338
Number of animals per watering place	0.317
Purchased soy (cows)	0.312
On farm fibrous food (calves)	0.305
Extra fibrous food (calves)	-0.305
Number of animals stationing	-0.314
On farm fodder (cows)	-0.322
On farm fodder (bulls)	-0.322
Vertical fanning (cows)	-0.337
Permanent meadow	-0.368
Single box (calves)	-0.392
Finished feed	-0.400
Purchased floor (cows)	-0.433
Manger front >0.6m	-0.463
On farm herbage (bulls)	-0.542

Table 9.2: The 22 variables extracted form the questionnaire (total of 201 features), based on their correlation with the target. Features with positive and negatives correlation lower than 30% were omitted from the study and were not listed

A splitting strategy similar to the one used in Chapter 8 was adopted. Unfortunately, the lack of time available to visit a significant number of candidate farms limited the total number of available breedings to 33. In this way, the study had to deal with a poor number of instances. However, a k-fold cross validation could be performed among the learning samples, maintaining at the same time a balanced percentage between learning and test sets (70%-30%). Thereafter, 9 records were extracted to form the test set and the remaining 24 formed the learning set. Among the latter, an 8-fold cross validation was imposed, obtaining 8 pairs of training-validation sets, consisting respectively in 21-3 instances. In order to perform a sufficient number

of runs of GP and to compare models, the technique was repeated 10 times by selecting the test instances sequentially among the main dataset. The learning instances were randomly shuffled before performing the 8-fold sampling.

Two stages were defined within the study: first, a ST-GP approach was performed in order to investigate its behaviour on three benchmarks, i.e., the Herd-Book data, the survey data, and both sets of data in input, with respect to the same target. Thereafter, standard ML methods were applied to compare performance on the data, i.e., KNN (4.3.2), NNET (4.3.5), RF (4.7), and extreme gradient boosting (4.3.4). The algorithms were tuned based on the average performance on the validation sets. Concerning the latter, the size of trees was set equal to 1, that is stumps were used as learners. The subsample ratio of features to be selected when constructing each tree, occurring once for every stump, was set equal to 0.8.
Final parameters are listed in Tables 9.3 and 9.4.

ML technique	**Parameters**
`knn`	`k = 9`
`nnet`	`size = 5; decay = 0.1`
`ranger`	`mtry = 2; splitrule = extratrees; sample.fraction = 1`
`xgboost`	`nrounds = 150; max_depth = 1, colsample_bytree=0.8`

Table 9.3: Parameters used to run ML techniques with the caret package in R

Parameter	**Description**
ST-GP	
Maximum number of generations	100
Population size	500
Selection Method	Lexicographic Parsimony Pressure
Elitism	Keepbest
Initialization Method	Ramped half and half
Tournament Size	2
Subtree Crossover Rate	0.7
Subtree Mutation Rate	0.1
Subtree Shrinkmutation Rate	0.1
Subtree Swapmutation Rate	0.1
Maxtreedepth	17

Table 9.4: Parameters used to perform ST-GP.

9.3 Results

The first goal was the investigation of GP's ability to predict the quota of weaned calves per cow in 2019, based on data collected in 2018. GP was thereafter compared with other methods. Due to the splitting rule in different subsets previously exposed, 10 models were finally obtained over three different benchmark problems:

- Benchmark A: variables in Table 9.1 were investigated, 1-15 were set as input, feature 16 as target.
- Benchmark B: variables in Table 9.2 were investigated, with feature 16 from Table 9.1 as target.
- Benchmark C: variable from both Tables 9.1 and 9.2 were investigated, with feature 16 from Table 9.1 as target.

We compared the RMSE between predicted and real valued on the learning and test data, to measure the accuracy of prediction. In order to compare the performance of the models achieved with GP on the learning sets, a non-parametric test was performed, to establish whether there is a significant difference between the RMSEs' medians. The median values were compared with Kruskal-Wallis test and the null hypothesis that all values are equal was not rejected ($p > 0.05$). All the distributions also did not result significantly different, according to Wilcoxon signed-rank test with Bonferroni correction ($\alpha = 0.05/3 = 0.016$), meaning that the three performance distributions on the learning sets are similar (p-values for all considered couples showed $p > 0.016$).
Similar remarks could be made on the test, as the three distributions showed p-values greater than the previously set significant thresholds. Indeed, Kruskal-Wallis test p-value among the medians was reported to be larger than 0.05, and Wilcoxon signed-rank test with Bonferroni correction produced p-values larger than 0.016. The corresponding box-plots are shown in Figure 9.3, the median and mean values in Table 9.5.

ST-GP	**A**	**B**	**C**
Results on the learning sets			
Median	0.08495	0.07723	0.07877
Mean	0.09049	0.07955	0.07701
Results on the test sets			
Median	0.1519	0.13829	0.14436
Mean	0.1606	0.14075	0.14484

Table 9.5: Median and mean RMSE of the different techniques among the learning and test sets

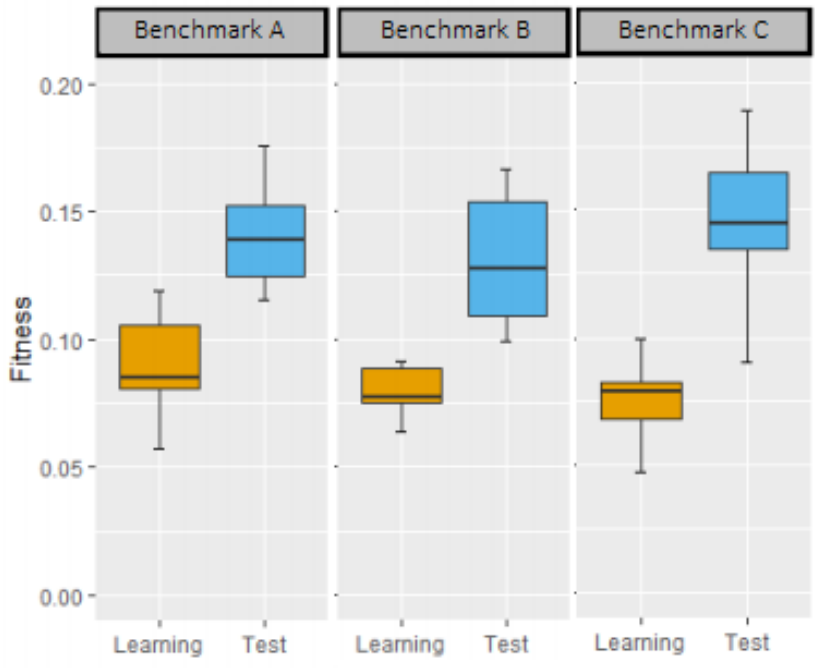


Figure 9.3: Boxplots showing RMSEs distribution achieved by ST-GP among the three different benchmaks

Within each benchmark, the distributions were compared to inspect possible overfitting issues. In all cases, the three pairs of learning and test sets were significantly different, as the corresponding p-values were reported to be $\sim$0.001.

The reported results were similar for the three benchmarks investigated with GP. As we are mainly interested in the variables extracted through the survey, supported by the fact that GP produced a better median fitness on the corresponding dataset, we focused on the second benchmark to perform comparisons with other ML methods. Statistical significance of the null hypothesis of no difference in medians between the learning fitness between GP and each of the other methods is based on the Kruskal-Wallis test, with α set equal to 0.05. The resulting p-value is extremely significant ($p = 0.1317 \cdot 10^{-4}$). Indeed, among the five involved methods, `xgbtree` performed better with statistically significant smaller fitness values (α previously set to 0.005 after Wilcoxon test with Bonferroni correction). The other techniques performed similarly on the learning set.

Concerning the results among the test sets, RMSEs produced by all the methods show the same distribution, as the Wilcoxon test reported non-significant outcomes. For further comparison, we measured overfitting as the difference between test and learning set RMSEs for each of the compared methods: all the learning-test pairs of distributions overfitted, as p-values obtained with Kruskal-Wallis test were $\sim$0.001 in all cases. In Figure 9.4 the corresponding boxplots were plotted.

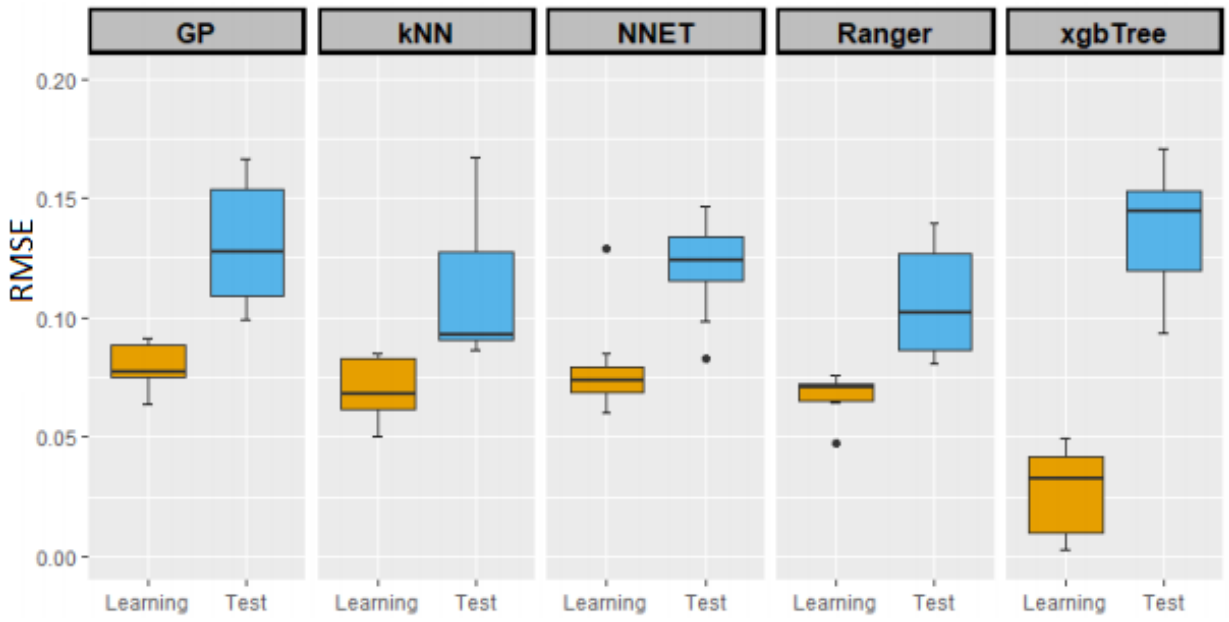


Figure 9.4: RMSE on both the learning and test set for the different algorithms. Learning sets results are plotted in yellow (left) and test results are plotted in blue (right) for each technique

In this study it was not possible to find methods that outperformed the others. First of all, the number of breeding farms was very small and the models could not actually learn properly. However, the performance obtained on the new set of variables did not differ from the performance recorded in previous studies, considering the variables of the Herd-Book. This aspects suggest a possible important role of the new information in predicting the target. ST-GP did not prove itself as the best performing method, but it can neither be classified as worse than the other techniques, since the reported fitness was comparable to that produced by KNN, NNET, and RF. A different behaviour was displayed by extreme gradient boosting. Thanks to its structure, it was adopted as suitable for dealing with the task. The results proved that the good predictive models built in the learning phase failed to generalize, as the lower RMSE was not confirmed in the testing phase, despite proper parameters tuning. This method turned out to be therefore unsuitable for that type of data, showing overfitting issues and inability to generalize.

9.4 Conclusions

In this chapter, we explored the use of GP in the field of zoothecnical modelling on a different dataset. The prediction of the number of weaned calves per cow in 2019 in the context of Piemontese cattle breedings was based on input data referred from 2018. The problem was previously studied using the data extracted from the Herd-Book. As widely illustrated, there are many aspects that influence the weaning period of the calf, that are substantially linked to its well-being. Most of them are not available in the archives. Hence, this

information required to design a new specific questionnaire to collect such information. Designed specifically on the basis of the characteristics of the breedings of interest, the form was used to describe and gather data regarding the environment on the farm, the rations and food supplied to the animals, the quality of air and water, available space, temperament of the animals, and technopathies and vaccination plans. All the information is related to animal welfare, which can lead to improvements in management, as well as to malfunctions, highlighting critical points and helping the farmer to evaluate the breeding performance. We then extracted the variables most correlated with the target under consideration. Three benchmarks were defined to evaluate ST-GP ability on additional data, not previously investigated, with respect to the studies carried out using the Herd-Book variables. Detecting a significantly different behavior between the three benchmarks, we focused on the second benchmark, containing only data extracted from the new form, as it produced better fitness. We assessed GP performance by means of experimental comparisons with other ML techniques. Based on the Root Mean Squared Error on the learning and test sets, the analysis revealed that GP and other common methods performed similarly, showing an analogous response on both the learning and test sets. Among the chosen algorithms, we concluded that extreme gradient boosting is not appropriate for the considered problem, as it showed evident overfitting. The preliminary results should be further investigated, as a very small number of breedings was taken into account, weakening predictive capabilities and statistical tests. Data collection through an on-field survey is a time consuming process, but from the first results it appears to bring interesting developments to the problem under analysis. For this reason, the current work is oriented towards the extension of farm visits, in order to collect data from as many farms as possible.

Chapter 10

Conclusion

The work carried out in this thesis aimed to perform investigations on the possible improvements in the modeling of the performance of beef farms, specifically on the basis of the situation identified within the Piemontese cattle breedings. Considering it as a case study, the measure of cattle breeding performance was investigated. The *Piemontese* breed outlined in Chapter 2 is known as the prevailing beef breed mainly raised in Piemonte, in the north-west of Italy. Differently from traditional beef breedings, the extensiveness conditions for the management are not met. Indeed, it is raised in intensive farms. However, it is suitable for being raised in the most diverse climates, as it is an excellent food processor and adaptable to more diverse conditions. The National Association of Piemontese Cattle Breeders (ANABORAPI) is responsible for the enhancement of the breed. The Association keeps the Herd-Book, runs a Genetic Station, where performance tests and progeny tests are carried out, and an Artificial Insemination Station where semen from A.I. bulls is produced. Therefore, it establishes the selection criteria. Among the various tools provided to the breeders, summary records are supplied in order to monitor of the breeding trend. The updated situation can be kept under control and variation in the overall performance of the farm can be inspected. Besides, working within the Association gave me the opportunity to investigate the information systems developed by ANABORAPI, becoming familiar with the zootechnical field and the crucial aspects in animal husbandry. In particular, I focused on the measure of breeding production efficiency, that revolves around the cows fertility and production, i.e., the calf quota generated yearly. With a particular focus on the weaning period, approximately two months after birth, it emerged that losses related to calf management are consistent, and the need for a methodology towards the construction of a more appropriate model was outlined.

10.1 The Definition of the Problem: Modeling Beef Farm Performance

The first aspect to tackle was the comprehension of the aspects that influence the performance of a farm of the type considered. To maximize revenues, it is essential that each mare produces as many calves as possible during her productive career, in full respect of her physiology. If well managed, the current Piemontese cow is able to produce and raise almost one calf per year. Indeed, the *"calf quota"* that each cow generates is the indicator of a cow's reproductive efficiency. Such a measure is derived from the *calving interval*, that is the number of days between two deliveries. The smaller the calf quota, the lower the fertility of the mare. By making the calf quota converge to 1, the breeding can be considered economically profitable. Clearly, the reproductive capacity affects the farmer's income, as the failure to give birth to calves and the cost of feeding the cows can become economically consistent. There can be several causes that lead to losses: one is represented by the period following the birth of the calf, as the necessary immediate interventions and calf conditions regulate this phase. In Chapter 3, all the issues related to the calving phase were illustrated and we concluded that it is reductive to measure breeding performance by observing only fertility and maternal condition, as it is currently being done. The calf goes through development stages that depend on its own condition, reaching the physiological development in 60 days after birth. Calf mortality was reported to be an important cause of economic damages in Piemontese cattle farms. Therefore, the breeding performance should be modelled considering also other factors such as neonatal mortality, outlining the calf's ability to survive, and the source of stress such as congenital calf's defects, compromising eventually the immune response and the growth rate, environmental and food conditions, that affect the quality of life of the newborn.

10.2 Improving Beef Farm Performance with Machine Learning

What are the limits of the model used to estimate farm performance? The estimate obtained with equation 3.1 is based on a classical statistical approach. It is a model formulated on a priori knowledge of the field, summarized in two parameters, that receives in input the annual corresponding average values and returns the estimate for that same year as output. If we consider the predictive aspect of the model, the estimate for the immediate future is provided considering the average calving interval among the cows pregnant at the moment of the query. The main outlined purpose was to supply the farmer with a more accurate indicator, to highlight effects that becomes evident over a few years from their introduction. Therefore, the need to identify a methodology that can respond appropriately arose. The identification of influential variables within big databases can be extremely difficult. The huge digitization of data collections streamlines procedures, for the registration and consequent processing of many additional data. Livestock is also in-

creasingly managed by continuous automated real-time monitoring, defining the field of Precision Livestock Farming, and contributing to the increase of the amount of information and complexity of databases. It can supplement the skills of the farmer, the veterinarian, and the technician, with the support of information technologies. Besides, it requires an active involvement of all subjects, in a Citizen Science perspective, in order to establish knowledge exchange, contributing to collection of programs and proper data monitoring.

"Big Data" can be introspected, providing an adequate answer to farmers, technicians, veterinarians, and all the subjects involved. Considering the huge size of data, visual inspection is not adequate. The increase in the amount of data requires proper data management and prediction techniques, to offer the possibility to process intrinsic information. To meet the two needs, that is to find an appropriate prediction model and deal with the available big data, we chose a Machine Learning (**ML**) approach. Indeed, as it is necessary to examine which variables available in the dataset impact the performance of a breeding farm, and to avoid a priori assumptions about model formulation, ML answers with great computing power and flexible algorithms, able to exploit the intrinsic information. Useful considerations about data should be limited to the preparation of the dataset and the analyses on the produced models, i.e., a posteriori interpretation. The model currently applied (Equation 3.1) is formulated on a priori zootechnical knowledge. Traditional statistical forecast analysis is preprogrammed, based on the hypothesis that past data is a good forecasting indicator for the future. ML methods are not pure magic for predicting the future. They are based also on past information, but their structure allow them to search for patterns directly among data. A dataset can be divided in such a way as to build the models based on a slice of it, by minimizing an error function, and to test the corresponding predictive model capacity on another slice of the dataset, applying it to data never seen before. Rather than making a priori assumptions, rather than following preprogrammed algorithms, ML allows the system to learn from data.

ML Methods respond to many different issues depending on the problem to deal with: in our case it was a supervised learning task, since the dataset contains the values of the variable we want to predict. Algorithms are regulated by a series of parameters, which, according to the data and the kind of problem, lead to better or worse learning and generalization ability among unseen data. Therefore, it is very important to do a search for the best values of these parameters. Part of this depends on learning, which is evaluated on the basis of selected functions. In our case we chose to measure the fitness of the algorithms by the Root Mean Square Error function. Chapter 4 was dedicated to the description of the whole process and to the architecture of the different applied techniques. The different techniques were selected on the basis of their widespread use in many different fields and their useful characteristics that are highlighted in the literature.

Regarding the latter, this type of approaches is lacking in the beef breeding sector and, in particular, in the Piemontese breed sector. ML is widely applied in the dairy sector, as evidenced by the hundreds of studies, partially listed in Chapter 3. The absence of direct comparisons with other research in the same sector was only partially a limiting issue, as the study presented in this thesis adds value to the ML approach, as it lays the foundation for possible future developments.

10.3 Datasets, Experimental settings, and Comparison of Different Techniques

The main pursued objective was to perform a comprehensive investigation of the possibilities for the improvement of modelling farm performance, in order to work in the perspective of subsequently integrate information systems. So far, each technique adapts differently to the data, producing different results on different datasets, depending on the prediction task, and on the split rules between training, validation, and test partitions. The result may be better, worse, or similar to others in terms of accuracy of the result. However, each of them exhibits characteristics that make a technique more interesting and appropriate than others.

10.3.1 The Dataset

Prediction models were built starting from a determined dataset. The farms and the variables handled through the entire study produced different benchmark problems. I inspected the main database, containing the summary data described in Chapter 6. I analyzed the meaning of all variables, the structure of the database, and the type of features contained. Globally, the database contains the last twenty years of data, including farms that are no longer active, for a total of 219 descriptive features reported in Table 5.1, including the number of calves alive per cow by equation 3.1, i.e., the defined target to predict. The criteria according to which the farms were thereafter selected were described, in order to define a pool of representative farms. Data recorded during a specific historical period, i.e., one year, were extracted to form the input set of predictors, as well as the target for the same subsequent time period, i.e., the next year. The structure of the database allowed me to distinguish two types of analysis. As the dataset is a historical archive, considering the target for a determined year, it was possible to isolate the summary data and use scalar predictors from the previous one in input or, on the other hand, to fully exploit the sequential information contained in several years. In order to perform the second approach, the multiple values for the predictors were collapsed into vectors. The editing defined two kinds of data frames, referred to as standard and vectorial.

10.3.2 Genetic Programming vs Common State-of-the-Art Methods

Different ML methods produce different kind of models that, depending on the architecture, generally are not available at the end of the whole process, besides resulting quite complex to understand. The sought models should be available for further analysis, in an attempt to understand which links were detected independently of the algorithm. Furthermore, a simple and legible expression is usually also simple for the user to interpret, when available. Genetic Programming (**GP**) is a method that comes with many desirable characteristics. First of all, it performs an implicit feature selection, by extracting informative attributes, and produces models that are resumed in intelligible expressions, potentially reducible to simpler forms. Besides this, it offers the possibility to handle vectorial predictors. A GP approach was first adopted to exploit the standard data panel. I applied the methodology, referred to as *Standard Genetic Programming* (**ST-GP**) in the thesis, using the standard data panel to solve different benchmark problems.

Predictive models were trained, validated and tested on a pool of 725 farms, with data recorded in 2017 and 2018, for a total of 19 predictors and one target. Among the supplied instances, 330 were reserved as a learning set, and 395 as a test set. Accurate models were achieved, showing that GP can learn from a smaller dataset composed by representative farms and predict good results on the selected test set. The algorithm was able to select and process important variables, without previous assumptions on the zootechnical aspects. The final candidate models performed well, exploiting more predictors and resulting in a more complex expression, hardly reducible to a simpler one. However, other predictive models encapsulating fewer variables were also achieved. Considering their expression, extremely simple and possibly easier to interpret from the zootechnical point of view despite a slightly higher error, GP proved to be flexible, allowing us to argue that accuracy is not the only criterion to measure the usefulness of the results. Preliminary results were obtained with ST-GP and were illustrated in Chapter 6.

Additional investigations on ST-GP, including the corresponding comparative methods results, were exposed in Chapter 7. The datasets was re-formed, based partly on the results of Chapter 6, but mainly considering the direction we wanted to take in the following, that is to investigate the performances on a vectorial data panel. For this reasons, a smaller pool was considered, i.e., composed of 304 farms, selected on the basis of the corresponding data recorded in the period 2014-2017. At the same time the number of predictors was increased to 48 to test the feature selection ability of the algorithm. The results were compared with other techniques: some common methods were selected to compare the results obtained with GP, known to be able to capture the high non-linearity underlying the data. Due to their structures, the methods encapsulated all the features into the prediction models and, differently from GP, did not perform an implicit feature selection.

If 48 variables are given in input, methods as Linear Regression, Random Forests, or Neural Networks end up using all features, with different degrees of importance. GP, on the other hand, begins the evolutionary process (of the population of models) using all the variables, but it manages to pass on the most informative ones from one generation to the next. The experimental results confirmed the considerations anticipated by the preliminary results, enforced by the comparison with other algorithms. On the one side, we handled classic techniques, producing on average models performing better than GP, showing lower fitness but complex expressions. On the other side, GP led to less accurate models in terms of performance, since the error was slightly greater, but easy to read and interpret. GP can combine a few variables, selected during the evolution process, into straightforward expressions. At the end of the procedure, the best models performed as well as those obtained with other commonly used techniques.

Subsequently, I investigated the results obtained with vectorial variables representing time series, increasing the amount of information available as input for the different techniques. Exploring the vectorial approach (Chapter 8) required, as already stated, a different input data structure. However, the target did not change. To this purpose, the farms considered in the pool of instances were the same as in Chapter 7. However, since the results showed that GP exploited only certain variables, I reduced the number of predictors to 18. In this way, possible noise due to extra variables, not very informative, was reduced. The main goal was to inspect the ability of *Vectorial Genetic Programming* (**VE-GP**) with respect to ST-GP, to predict the target. The recently developed VE-GP algorithm could produce better results, by achieving better fitness on both the learning and test sets. VE-GP proved to be more stable, evolving a population through more generations without showing overfitting, while ST-GP, was affected by overfitting already in the early generations, under similar experimental settings. VE-GP still favored the interpretability of the solutions, by giving access to the formula and to the features implicitly selected, providing meaningful information about the tackled issue. Moreover, better results were obtained by encapsulating fewer variables in each extracted candidate model, detecting almost all the information among specific features. The algorithm improved the target forecast, proving to outperform not only ST-GP, but also other techniques used in the field of ML. The algorithm, in particular, was compared to Long Short Term Memory Recurrent Neural Network (**LSTM**), suitable for handling vectorial predictors. Although VE-GP performance was similar to LSTM and Generalized Linear Models (the latter exploiting the standard data panel representation), it was the only method showing similar learning and test fitness distribution, entailing greater ability in generalization among unseen data.

While performing the vectorial approach, I ran a parallel research on an enriched dataset, built by adding to the Herd-Book predictors a series of very informative zootechnical information. Indeed, while vectorial pre-

dictors are very informative and helped to gain more accurate results, we wondered whether other variables, not present in the original database, could also be. In Chapter 3 it was highlighted that many other factors influence the performance of the farm. Since they are currently absent in the registries as they are not coded nor usually collected, it was necessary to design a specific questionnaire (Chapter 9), to be filled in through farm visits. In this regard, after planning the methodology to be pursued, I was assisted to learn the necessary animal husbandry notions to better understand the problem, and thereafter fill directly the form on the farms. The survey was organized based on a previous version, supplied for fattening breedings. The farms that we selected to this purpose are breeding farms. Hence, we enriched the survey with additional information about vaccination plan, technopathies, description of animals, focusing on temperament, litter, and the corresponding feeding type and ration. Besides this, we set the codes to describe the type of ventilation, the cleaning of the trough, the micro-climatic reliefs (light and presence of ammonia), and the temperament of the animals. The farms were selected based on location, size, and the availability of farmers to the visit and the interview. From a total number of 201 variables, I extracted the features mostly correlated to the target, the latter referred to 2019 as the visits were performed between March and April 2019. The selected predictors were finally 22, combined with Herd-Book data referred to 2018. GP performance was assessed by means of three benchmarks to evaluate a possible improvement of ST-GP on additional data, compared to the basic study carried out on the Herd-Book variables. Thereafter, experimental comparisons with other ML techniques in the field were made possible. The analysis revealed that GP did not show different behaviors between the three benchmarks, that focused respectivly on Herd-Book data, on field survey data, and on the combination of both sets. Comparisons were hence conducted among the second benchmark, as GP produced better results. Techniques performed almost all similarly and it was not possible to determine the presence of the best-performing one. Nevertheless, ST-GP did not produce worse results with respect to the other techniques, since reported fitness was comparable to that produced by the others.

10.4 Further Considerations

The objective envisaged a first approach to find a methodology to build predictive models for the measurement of breeding performance. As it is not appropriate to impose a priori assumptions about the variables and the models, given the size of the datasets and the large number of involved features, ML methods addressed properly the task. In general it is possible for us to state that ML is suitable for predicting the defined target, i.e., the number of calves weaned per cow per year. The chosen algorithms proved to be suitable to solve the issue on different subsets of the main dataset. Above all, they provided interesting comparisons with GP, that, among all the techniques, offered a wider range of characteristics that responded appropriately to the problem under analysis. We can assert that GP could represent the baseline along all the study,

proving to be the most suitable method, considering its usefulness in providing accessible and interpreatable models. Variables were automatically selected and combined together, offering the possibility of additional investigations among the features selected by candidate models. Indeed, the models produced by GP are mathematical expressions in plain text, showing the relationships found between the variables. After all, the collection of new data, exploited with an on-field survey, provided also some insights that promise good developments. We saw that, despite having very few reference farms, ML and, in particular, GP produced encouraging results, entailing the proper usefulness of both the undertaken data collection and the approach. One of the major issues that arises from this study is the need to apply the methodology to a larger dataset. Relaxing the thresholds imposed on the selection of breedings samples would determine more instances in the pool, likely yielding better results. ML is designed to handle thousands of data. Most importantly, by extending the number of instances, the algorithms could learn from more examples and be tested on more different cases as well, helping building more accurate predictions. However, not only the size of the dataset is to be considered useful in achieving performing models. The most interesting aspect about using GP is rather the vectorial development. The introduction of vectorial variables produced a significant improvement among the accuracy of the result. Evolution was also much more stable and the fact that the algorithm can handle any type of variable, both scalar and vectorial, makes it quite a flexible tool. These considerations open the possibility of providing more complex datasets, containing different types of sequential features. The possibility of managing vectorial variables, whose values can be of different types and have no fixed length among the whole dataset, push the analysis beyond the basic research presented in this thesis. On the one side, both categorical and continuous variables can be treated simultaneously, without specifying it explixitly to the algorithm: the latter is indeed able to process them during the evolution without hints given by the user. On the other hand, when dealing with vectors, some data may be not available, i.e., the vector variables may not have the same length. Moreover, it is admissible to handle also scalars, which do not show a temporal trend. VE-GP is suitable to manage all kinds of features dynamically, combining them in the prediction of the target. Evolutionary algorithms can be applied to zootechnical data, achieving performing models, able to learn from all the available data. In this case study, the breeding report variables extracted at the end of the year were used. In one case they were managed for only one year, in the other four the average values, corresponding to four years, were used, proving to be more suitable for reducing the prediction and generalization errors. Instead of limiting the analysis to the year-end average, it might be more useful to incorporate the data collected from each farm visit into a vector representation. As a result, all variations, even small ones, would be available and the algorithm could identify temporal patterns that were not visible by directly processing the average value for the whole year.

Bibliography

[1] F. Abbona, L. Vanneschi, M. Bona, and M. Giacobini. Towards modelling beef cattle management with genetic programming. *Livestock Science*, 241:104205, 2020.

[2] F. Abbona, L. Vanneschi, M. Bona, and M. Giacobini. A GP approach for precision farming. *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2020.

[3] A. Abraham, N. Nedjah, and L.M. Mourelle. Evolutionary computation: from genetic algorithms to genetic programming. In *Genetic Systems Programming*, 2006.

[4] A. Albera, R. Mantovani, G. Bittante, A. Groen, and P. Carnier. Genetic parameters for daily live-weight gain, live fleshiness and bone thinness in station-tested piemontese young bulls. *Animal Science*, 72(3):449–456, 2001. doi: 10.1017/S1357729800051961.

[5] A. Albera, P. Carnier, and A.F. Groen. Definition of a breeding goal for the piemontese breed: economic and biological values and their sensitivity to production circumstances. *Livestock Production Science*, 89(1):66–77, 2004. doi: 10.1016/j.livprodsci.2003.12.004.

[6] A. Albera, A.F. Groen, and P. Carnier. Genetic relationships between calving performance and beef production traits in piemontese cattle. *Journal of Animal Science*, 82(12):3440–3446, 2004. doi: 10.2527/2004.82123440x.

[7] E. Alfaro-Cid, K. Sharman, and A. Esparcia-Alcázar. Genetic programming and serial processing for time series classification. *Evolutionary Computation*, 22:265–285, 2014.

[8] J. Alonso, Á.R. Castañón, and A. Bahamonde. Support vector regression to predict carcass weight in beef cattle in advance of the slaughter. *Computers and Electronics in Agriculture*, 91:116–120, 2013.

[9] B. J. Amrine, D.E.and White and R. Larson. Comparison of classification algorithms to predict outcomes of feedlot cattle identified and treated for bovine respiratory disease. *Computers and Electronics in Agriculture*, 105:9–19, 2014.

[10] ANABORAPI. Associazione nazionale allevatori bovini razza piemontese. `http://www.anaborapi.it`.

[11] I. Azzali, L. Vanneschi, S. Silva, I. Bakurov, and M. Giacobini. A vectorial approach to genetic programming. In *EuroGP*, 2019.

[12] I. Azzali, L. Vanneschi, I. Bakurov, S. Silva, M. Ivaldi, and M. Giacobini. Towards the use of vector based GP to predict physiological time series. *Appl. Soft Comput.*, 89:106097, 2020.

[13] P. Bartashevich, I. Bakurov, S. Mostaghim, and L. Vanneschi. Evolving pso algorithm design in vector fields using geometric semantic GP. *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2018.

[14] D. Berckmans. Precision livestock farming technologies for welfare management in intensive livestock systems. *Revue scientifique et technique*, 33 1:189–96, 2014.

[15] D. Berckmans. General introduction to precision livestock farming. *Animal Frontiers*, 7(1):6–11, 2017. doi: 10.2527/af.2017.0102.

[16] D. Berckmans and M. Guarino. Precision livestock farming for the global livestock sector. *Animal Frontiers*, 7(1):4–5, 2017. doi: 10.2527/af.2017.0101.

[17] G. Bittante, M. Ramanzin, and I. Andrighetto. *Tecniche di produzione animale*. Liviana, 2005.

[18] M. Bona, A. Albera, G. Bittante, A. Moretta, and G. Franco. L'allevamento della manza e della vacca piemontese. *Supplemento al n. 44 dei Quaderni della Regione Piemonte-Agricoltura*, pages 65–129, 2005.

[19] T. Bylander. Estimating generalization error on two-class datasets using out-of-bag estimates. *Machine Learning*, 48:287–297, 2004.

[20] P. Carnier, A. Albera, R. Dal Zotto, A.F. Groen, M. Bona, and G. Bittante. Genetic parameters for direct and maternal calving ability over parities in piedmontese cattle. *Journal of Animal Science*, 78 (10):2532–2539, 2000. doi: 10.2527/2000.78102532x.

[21] E. Casas, J. Keele, S. Shackelford, M. Koohmaraie, T. Sonstegard, T. Smith, S. Kappes, and R. Stone. Association of the muscle hypertrophy locus with carcass traits in beef cattle. *Journal of animal science*, 76 2:468–73, 1998.

[22] T. Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[23] J.B. Cole, S. Newman, F. Foertter, I. Aguilar, and M. Coffey. Breeding and genetics symposium: Really big data: Processing and analysis of very large data sets. *Journal of Animal Science*, 90(3):723–733, 2012. doi: 10.2527/jas.2011-4584.

[24] M. Craninx, V. Fievez, B. Vlaeminck, and B. Baets. Artificial neural network models of the rumen fermentation pattern in dairy cattle. *Computers and Electronics in Agriculture*, 60:226–238, 2008.

[25] A. De Vries and M. I. Marcondes. Review: Overview of factors affecting productive lifespan of dairy cows. *Animal : an international journal of animal bioscience*, 14 S1:s155–s164, 2020.

[26] H. Drucker. Improving regressors using boosting techniques. In *ICML*, 1997.

[27] S. Dunner, C. Charlier, F. Farnir, B. Brouwers, J. Canon, and M. Georges. Towards interbreed IBD fine mapping of the mh locus: Double-muscling in the asturiana de los valles breed involves the same locus as in the belgian blue cattle breed. *Mammalian Genome*, 8:430–435, 1997. doi: 10.1007/s003359900462.

[28] R. Dutta, D.V. Smith, R. Rawnsley, G. Bishop-Hurley, J. Hills, G. Timms, and D. Henry. Dynamic cattle behavioural classification using supervised ensemble classifiers. *Comput. Electron. Agric.*, 111: 18–28, 2015.

[29] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119 – 139, 1997. doi: 10.1006/jcss.1997.1504.

[30] M. Gahegan. Is inductive machine learning just another wild goose (or might it lay the golden egg)? *International Journal of Geographical Information Science*, 17:69 – 92, 2003.

[31] D. Gianola, H. Okut, K. Weigel, and G. Rosa. Predicting complex quantitative traits with bayesian neural networks: a case study with jersey cows and wheat. *BMC Genetics*, 12:87 – 87, 2011.

[32] O. Gonzalez-Recio, G.J.M. Rosa, and D. Gianola. Machine learning methods and predictive ability metrics for genome-wide prediction of complex traits $. *Livestock Science*, 166:217–231, 2014.

[33] L. Grobet, L.J. Martin, D. Poncelet, D. Pirottin, B. Brouwers, J. Riquet, A. Schoeberlein, S. Dunner, F. Ménissier, J. Massabanda, R. Fries, R. Hanset, and M. Georges. A deletion in the bovine myostatin gene causes the double–muscled phenotype in cattle. *Nat Genet*, 17:71–74, 1997. doi: 10.1038/ng0997-71.

[34] O. Guzhva, H. Ardö, A. Herlin, M.G. Nilsson, K. Åström, and C. Bergsten. Feasibility study for the implementation of an automatic system for the detection of social interactions in the waiting area of automatic milking stations by using a video surveillance system. *Comput. Electron. Agric.*, 127:506–509, 2016.

[35] I. Halachmi and M. Guarino. Editorial: Precision livestock farming: a 'per animal' approach using advanced monitoring technologies. *Animal : an international journal of animal bioscience*, 10 9:1482–3, 2016.

[36] T. Hastie, R. Tibshirani, and J. Friedman. The elements of statistical learning: Data mining, inference, and prediction, 2nd edition. In *Springer Series in Statistics*, 2009.

[37] A. Hessle, M. Therkildsen, and K. Arvidsson-Segerkvist. Beef production systems with steers of dairy and dairy × beef breeds based on forage and semi-natural pastures. *Animals*, 9, 2019.

[38] K. Holladay and K.A. Robbins. Evolution of signal processing algorithms using vector based genetic programming. *2007 15th International Conference on Digital Signal Processing*, pages 503–506, 2007.

[39] F.B. Hutt. A hereditary lethal muscle contracture in cattle. *Journal of Heredity*, 25(1):41–46, 1934. doi: 10.1093/oxfordjournals.jhered.a103843.

[40] H. Jabbar and R. Khan. Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study). 2014.

[41] Silke Janitza and R. Hornung. On the overestimation of random forest's out-of-bag error. *PLoS ONE*, 13, 2018.

[42] R. Kambadur, M. Sharma, TP Smith, and JJ. Bass. Mutations in myostatin (GDF8) in double-muscled belgian blue and piedmontese cattle. *Genome Res*, 7(9):910–6, 1997. doi: 10.1101/gr.7.9.910.

[43] C. Kamphuis, H. Mollenhorst, J. Heesterbeek, and H. Hogeveen. Detection of clinical mastitis with sensor data from automatic milking systems is improved by using decision-tree induction. *Journal of dairy science*, 93 8:3616–27, 2010.

[44] K. Kapitanova and S. Son. Machine learning basics. 2012.

[45] J. Koza. Genetic programming as a means for programming computers by natural selection. *Statistics and Computing*, 4:87–112, 1994.

[46] M. Kuhn. Classification and regression training [r package caret version 6.0-86]. 2020. `https://cran.r-project.org/web/packages/caret/caret.pdf`.

[47] J.J. Lauvergne, B. Vissac, and A. Perramon. Étude du CaractÈre Culard. I. Mise au Point Bibliographique. *Annales de zootechnie*, 12(2):133–156, 1963. `https://hal.archives-ouvertes.fr/hal-00886796`.

[48] H.W. Leipold, W.F. Cates, O.M. Radosits, and W.E. Howell. Arthrogryposis and associated defects in newborn calves. *American Journal of Veterinary Research*, 31:1367–1374, 1970.

[49] K. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis. Machine learning in agriculture: A review. *Sensors*, 18(8):2674, Aug 2018. ISSN 1424-8220. doi: 10.3390/s18082674.

[50] C. Lokhorst, R. M. de Mol, and C. Kamphuis. Invited review: Big data in precision dairy farming. *animal*, 13(7):1519–1528, 2019. doi: 10.1017/S1751731118003439.

[51] E. Lynch, M. McGee, and B. Earley. Weaning management of beef calves with implications for animal health and welfare. *Journal of Applied Animal Research*, 47(1):167–175, 2019. doi: 10.1080/09712119.2019.1594825.

[52] G. Machado, M. R. Mendoza, and L. G. Corbellini. What variables are important in predicting bovine viral diarrhea virus? A random forest approach. *Vet. Res.*, 46(85), 2015. doi: 10.1186/s13567-015-0219-7.

[53] R. Mantovani, M. Cassandro, B. Contiero, A. Albera, and G. Bittante. Genetic evaluation of type traits in hypertrophic piemontese cows. *Journal of Animal Science*, 88(11):3504–3512, 2010. doi: 10.2527/jas.2009-2667.

[54] A.C. McPherron and S.J. Lee. Double muscling in cattle due to mutations in the myostatin gene. 94, 1997. doi: 10.1073/pnas.94.23.12457.

[55] R.S. Michalski, J.G. Carbonell, and T.M. Mitchell. Eds. 2013. doi: 10.1007/978-3-662-12405-5.

[56] A. Mimosi, P. Cornale, and S. Gemello. Razza bovina piemontese: analisi gestionale e definizione di un modello per l'ingrasso del vitellone. 2013-2014. Tesi di laurea.

[57] M. W. Mitchell. Bias of the random forest out-of-bag (oob) error for certain input parameters. *Open Journal of Statistics*, 2011:205–211, 2011.

[58] G. Morota, R.V. Ventura, F.F. Silva, M. Koyama, and S.C. Fernando. Big data analytics and precision animal agriculture symposium: Machine learning and data mining advance predictive big data analysis in precision animal agriculture. *Journal of animal science*, 96(4):1540–1550, 2018. doi: 10.1093/jas/sky014.

[59] A. Ortiz-Pelaez and D.U. Pfeiffer. Use of data mining techniques to investigate disease risk classification as a proxy for compromised biosecurity of cattle herds in wales. *BMC Vet Res.*, 4(24), 2008. doi: 10.1186/1746-6148-4-24.

[60] M. Paganini. *L'allevamento del bovino da carne*. Le Point Veterinaire Italie, 2019.

[61] Sistema Piemonte. Uba. http://www.sistemapiemonte.it/agricoltura/dw_rpu/glossario3.shtml.

[62] R. Poli, W. Langdon, and N. McPhee. A field guide to genetic programming. 2008. doi: 10.1007/s10710-008-9073-y. Lulu Enterprises, UK Ltd.

[63] E.O. Price, J.E. Harris, R.E. Borgwardt, M.L. Sween, and J.M. Connor. Fenceline contact of beef calves with their dams at weaning reduces the negative effects of separation on behavior and growth rate. *Journal of Animal Science*, 81(1):116–121, 2003. doi: 10.2527/2003.811116x.

[64] C.J. Rutten, A. Velthuis, W. Steeneveld, and H. Hogeveen. Invited review: sensors to support health management on dairy farms. *Journal of dairy science*, 96(4):1928–1952, 2013.

[65] A.L. Samuel. Some studies in machine learning using the game of checkers. *IBM J. Res. Dev.*, 3: 210–229, 1959.

[66] S. Savoia, A. Brugiapaglia, A.and Pauciullo, L. Di Stasio, S. Schiavon, G. Bittante, and A. Albera. Characterisation of beef production systems and their effects on carcass and meat quality traits of piemontese young bulls. *Meat science*, 153:75–85, 2019.

[67] R. Schapire. The strength of weak learnability. *Machine Learning*, 5:197–227, 1990.

[68] CFSPH Center For Food Security and Public Health. Bovine diseases and resources. http://www.cfsph.iastate.edu/Species/bovine.php.

[69] S. Silva. GPLAB - a genetic programming toolbox for matlab. 2007. http://gplab.sourceforge.net/index.html.

[70] A. Spiess and Natalie Neumeyer. An evaluation of r2 as an inadequate measure for nonlinear models in pharmacological and biochemical research: a monte carlo approach. *BMC Pharmacology*, 10(6), 2010.

[71] C. Strachey. Logical or non-mathematical programmes. In *ACM '52*, 1952.

[72] H. Tao, F. Guo, Y. Tu, B.W. Si, Y.C. Xing, D.J. Huang, and Q.Y. Diao. Effect of weaning age on growth performance, feed efficiency, nutrient digestibility and blood-biochemical parameters in droughtmaster crossbred beef calves. *Asian-Australasian journal of animal sciences*, 31(6):864–872, 2018. doi: 10.5713/ajas.17.0539.

[73] A. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of The London Mathematical Society*, 41:230–265, 1937.

[74] A. Turing. Computing machinery and intelligence (1950). *Mind*, LIX:433–460, 1950. doi: 10.1093/mind/LIX.236.433.

[75] E.M. van der Heide, R. Veerkamp, M.L. van Pelt, C. Kamphuis, I. Athanasiadis, and B. Ducro. Comparing regression, naive bayes, and random forest methods in the prediction of individual survival to second lactation in holstein cattle. *Journal of dairy science*, 2019.

[76] T.L. Wheeler, S.D. Shackelford, E. Casas, L.V. Cundiff, and M. Koohmaraie. The effects of piedmontese inheritance and myostatin genotype on the palatability of longissimus thoracis, gluteus medius, semimembranosus, and biceps femoris. *Journal of Animal Science*, 79(12):3069–3074, 2001. doi: 10.2527/2001.79123069x.

[77] M.L. Williams, N.M. Parthalin, P. Brewer, W.P.J. James, and M.T. Rose. A novel behavioral model of the pasture-based dairy cow from gps data using data mining and machine learning techniques. *J Dairy Sci.*, 99(3):2063–2075, 2016. doi: 10.3168/jds.2015-10254.

[78] D.R. Wilson and T.R. Martinez. Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38:257–286, 2000. doi: 10.1023/A:1007626913721.

[79] I. Witten, Eibe Frank, M. Hall, and Chris Pal. Data mining, fourth edition: Practical machine learning tools and techniques. 2016.

[80] C. Yao, X. Zhu, and K.A. Weigel. Semi-supervised learning for genomic prediction of novel traits with small reference populations: an application to residual feed intake in dairy cattle. *Genetics Selection Evolution*, 48(1):84, December 2016. doi: 10.1186/s12711-016-0262-5. `https://hal.archives-ouvertes.fr/hal-01479213`.

[81] H. Zheng, H. Wang, and T. Yan. Modelling enteric methane emissions from milking dairy cows with bayesian networks. *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1635–1640, 2016.

Acknowledgements

I would like to express all my gratitude to the people who contributed to this project, from both professional and human points of view, enhancing my wealth of knowledge and experience. Therefore, I would first like to thank my supervisor, Professor Mario Giacobini, for giving me all the necessary tools, support and ideas to develop the thesis. My thanks, above all, for having recognized the possibility to accomplish this project, and for having bet on it.

This doctorate was performed in collaboration with ANABORAPI, to which I am indebted for providing me with the Herdbook data and for granting the time required to develop the work, even abroad. Thus, I thank the Director of the Association Andrea Quaglino and my colleague Marco Bona. A deep thank to him for having provided, together with Professor Giacobini, the main ideas and concepts necessary to define the study, as well as for having invested in the possibility of combining my educational background, of a mathematical nature, in the zootechnical field.

A heartfelt thanks to Alessio Moretta, who assisted me in drafting the questionnaire and guided me during farm visits, allowing me to understand the different management practices.

Concerning the foreign period, I will never thank Professor Leonardo Vanneschi enough for sharing his invaluable knowledge in Machine Learning, and for welcoming me in Lisbon. There, I found an extremely stimulating environment and amazing friends.

I would like to thank the reviewers, who contributed to enhancing the thesis and provided important insights for possible developments of the work.

Last, but definitely not least, warm thanks to my colleagues and my family, who have never missed to show interest and support, as well as useful discussions.