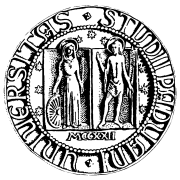

UNIVERSITY OF PADOVA

Department of Information Engineering

Master's Degree Thesis in Bioengineering

Ahead of time prediction of nocturnal hypoglycemic events from Continuous Glucose Monitoring data in people with type I diabetes by Machine Learning-based approaches

Supervisor:

Prof. Andrea Facchinetti

Candidate:

Gaia Bondani

Co-Supervisor:

Prof. Giovanni Sparacino

Eng. Giacomo Cappon

16 December 2019

A.Y. 2018-2019

Abstract

Diabetes mellitus is a major and increasing global problem, as reported in the 2016 Global Report of the *World Health Organizations* (`www.who.int/diabetes/global-report/en/`). The number of people with diabetes has risen from 108 million in 1980 to 422 million in 2014 and an estimated 1.6 million deaths in 2016 were directly caused by diabetes. Diabetes can lead to short-term complications caused by glycemic levels higher (*hyperglycemia*) and lower (*hypoglycemia*) than the normal range, while prolonged hyperglycemia leads to long-term complications that affect the vascular system, the central nervous system and organs such as kidneys and eyes. These long term complications usually result in cardiovascular disease, retinopathy and nephropathy. To avoid these scenarios, a correct managing of blood glucose level is necessary to keep it inside the acceptable range.

The control of the blood glucose (BG) level inside safe limits, called *euglycemic range*, requires continuous monitoring of blood glucose level. The most common method for type 1 diabetics is the Self Monitoring Blood Glucose (SMBG), that is a finger prick test taken several times a day. These readings are used to adjust the required insulin dose, indispensable to keep the blood glucose between the euglycemic range. In the last two decades another blood glucose monitoring method has been developed: the *Continuous Glucose Monitoring (CGM) system*. This kind of sensor allows diabetic people to control their BG levels in every moment of the day, whenever they want, thanks to frequently BG acquisitions and to a minimally invasive implanta-

tion on the patient skin. The most common CGM sensors give the measurements every 1/3/5 minutes, all day long and for several days continuously, up to ten. The advantages of this type of control are many, above all the patients can use the information provided by CGMs to better adjust the injection of exogenous insulin and then achieve a more precise glycemic control.
Furthermore, the availability of all these glycemic data, collected by the CGM systems, has encouraged the development of many algorithms that can improve the performance of the BG management. For example, the past CGM data can be used to predict future glycemic levels with even 20-30 minutes of advance. These predictions, coupled with suitable alerts, can prevent or at least mitigate the hypoglycemic/hyperglycemic episodes before they occur, by warning the patients with sufficient anticipation to take some countermeasures (i.e. lowering the basal insulin in case of a predicted hypoglycemic event). Many models and algorithms have already been proposed in literature, to solve different kind of prediction tasks related to the glycemic control in diabetic people. In this work, an approach still not deeply investigated is carried out: the prediction of nocturnal hypoglycemic events using only the CGM data of the previous day employing Machine Learning approaches.
In chapters 1 the background of the problem of the glycemic control in diabetic people and the description of the task are illustrated, while in chapter 2 the datasets used and the pre-processing methodologies are explained. To try to solve the chosen task many analyses and models have been carried out and they are presented in chapter 3 and 4. The obtained results are shown and discussed in chapter 5. As consequent of the results, another analysis has been investigated and it is described in chapter 6. Finally, in chapter 7 there are a conclusion about the work and the suggested further analyses.

Keywords: biological signal processing, diabetes, data analysis, machine-learning approaches, CGM data

Contents

CHAPTER 1

Diabetes and Continuous Glucose Monitoring

In this first chapter, the aim and the motivations of the thesis are illustrated. An overview on the problem of diabetes management is presented to help to understand the rationale behind the aim of this thesis work. Then, a panoramic on the literature methods is made, to get a complete overview of the problem to solve and of how it is tried to solve.

1.1 Diabetes Mellitus

1.1.1 The problem of the diabetes mellitus

From the *World Health Organizations* report published in 2016, the number of people with diabetes has risen from 108 million in 1980 to 422 million in 2014 and an estimated 1.6 million deaths in 2016 were directly caused by diabetes (`www.who.int/diabetes/global-report/en/`). These facts indicate that diabetes impact is growing, it is no longer a disease of predominantly rich nations because the prevalence of diabetes is steadily increasing everywhere, most markedly in the world's middle-income countries. This increase leads also to important and heavier social and economic costs for society and for the nations. In fact, diabetes and its complications bring about a substantial economic loss to people with diabetes and their families,

and to health systems and national economies through direct medical costs and loss of work. Based on cost estimates from systematic reviews [NCD-RisC, 2016], [Seuring *et al.*, 2015], it has been estimated that the direct annual cost of diabetes to the world is more than US$ 827 billion. For the future the study [Bloom *et al.*, 2011] estimates that losses in *Gross Domestic Product (GDP)* worldwide from 2011 to 2030, including both direct and indirect costs of diabetes, will total US$ 1.7 trillion, comprising US$ 900 billion for high-income countries and US$ 800 billion for low- and middle-income countries.
Diabetes prevention, treatment and management are then socio-health emergencies that push also the scientist and biomedical community to explore innovative methodologies and technologies to deal with these problems.

1.1.2 Overview of the disease

Diabetes Mellitus is a chronic disease that occurs either when the pancreas does not produce enough insulin or when the body cannot effectively use the insulin it produces. Insulin is a hormone that regulates blood sugar concentration and it is produced by the beta-cells of the pancreas. Normally it is released when the glucose concentration in the blood rises, in fact the insulin-primary function is lowering it after a meal. When food is digested and enters in the bloodstream, insulin moves glucose out of the blood and into cells, where it is processed to produce energy. Insulin promotes the uptake of glucose by the muscles, suppresses the hepatic production of glucose by the liver and controls the conversion of glucose into glycogen for internal storage in the liver. So insulin has an anabolic effect, inducing the organism to store and to synthesize nutrients; furthermore it is hypoglycemic because it reduces endogenous production of glucose in the liver and it stimulates the use of glucose in insulin-dependent organs. The normal range of the blood glucose concentration in healthy subjects is $70 - 180\ mg/dl$, that is the so-called *euglycemic-range*.

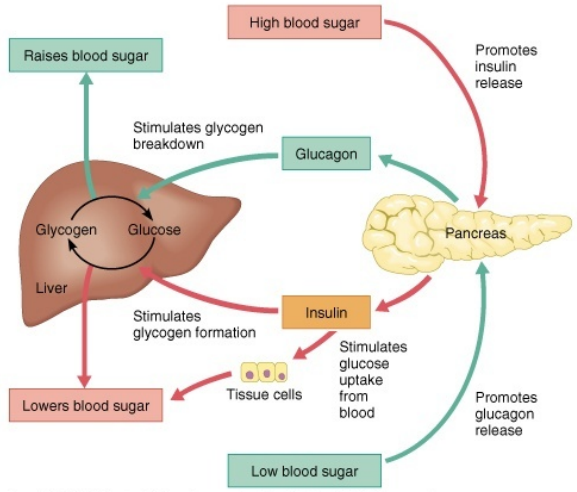

Figure 1.1: Scheme of the endocrine system to control the homeostatic feedback loop

In diabetic people, the body is unable to break down glucose into energy because there's either not enough insulin to move the glucose, or the insulin produced does not work properly. As consequence, the concentration of glucose in the blood (BG) often exceeds the euglycemic range. Hyperglycemia and hypoglycemia, which are the situation of higher and lower values of BG with respect to the glycemic-thresholds, cause many short and long-term health problems. The principal long-term complications are supposed to be caused by the chronic hyperglycemia and by the lack of insulin that protracting over time cause micro- and macro-vascular damages (due to the induced protein glycation) and then neuropathic and cardiovascular problems. The main complications are:

- *Diabetic Neuropathy*: the damage of the nerve is assessed to the reduction of the glucose in sorbitol, which can not be metabolized by many tissues such as ocular and nerves. This anomalous amount of sorbitol causes some osmotic damages and then could be at the basis of the peripheral neuropathy. This results in affected functionality of the nerves and causes many problems as the loss of feeling in parts of the body or painful, tingling, burning feeling. Furthermore, combined with reduced blood flow, neuropathy in the feet increases the chance of foot ulcers, infection and potentially the need for limb amputation.

- *Diabetic Retinopathy*: long-term accumulated damage to the small blood vessels in the retina can cause leaking of the fluid from them and cause swelling in the

macula. Swelling and fluid can cause blurry vision, making hard to see and if retinopathy gets worse, it may lead to blindness. The study [Bourne *et al.*, 2013] has assessed that 2.6% of global blindness can be attributed to diabetes.

- *Diabetic Nephropathy*: the damage of the kidney is the result of the damage to its blood vessels that can lead to kidney failure. Some people who have nephropathy will eventually need dialysis or a kidney transplant.

- *Heart disease and stroke*: the damage of the macro-vascular system can lead to coronary heart disease, strokes and peripheral vascular disease. Adults with diabetes have a two- to three-fold increased risk of heart attacks and strokes [Sarwar *et al.*, 2010].

Keeping blood sugar levels very close to the ideal can minimize, delay, and in some cases even prevent the problems that diabetes can cause. However, it is often difficult, since there are several factors that contribute to hyperglycemia in people with diabetes, including food and physical activity choices, illness, or not taking enough glucose-lowering medication.
The short-term complications instead are related to extremely out of range BG. The two scenarios are:

- *Hypoglyecemic Coma*: hypoglycemia can happen if diabetic people assume an excessive dose of insulin or if they perform an unusual intense physical activity or if they skip a meal. When BG goes down to values of $50 - 70$ mg/dl, the central nervous system becomes more excitable and a further reduction can bring to convulsions and loss of consciousness. In the extreme case, when glycemia goes below 20 mg/dl, hypoglycemic coma can occur and cause the death.

- *Hyperglyecemic Coma*: extreme hyperglycemia can happen in case of lack of insulin or illness, trauma or surgery. In this situation, the organism cannot process glucose. For this reason, a transition from carbohydrates metabolism to fat metabolism can occur, leading to the so-called *diabetic ketoacidosis* caused by the production of toxic acids (known as ketones) in the new metabolism. It can also occur that, in case of blood sugar level higher than 600 mg/dL (the *diabetic*

hyperosmolar syndrome), the excess sugar passes from the blood into the urine, which triggers a filtering process that draws an extreme amount of fluid from the body. Left untreated, these situations can lead to the coma.

It is possible to prevent diabetic coma controlling the diabetes status following the medications and meal plans, checking the blood sugar level frequently and being prepared in case of emergencies.

1.1.3 Type 1 Diabetes (T1D)

Type 1 diabetes, previously known as *insulin-dependent, juvenile* or *childhood-onset*, is characterized by deficient insulin production by the pancreas and requires daily administration of insulin. Only approximately 5% of people with diabetes are T1 diabetic. The cause of type 1 diabetes is not completely known and it is not preventable with current knowledge. In most people with type 1 diabetes, the body's immune system attacks and destroys the cells in the pancreas that make insulin. T1D typically occurs in children and young adults, although it can appear at any age. There is not a cure for this disease, the only therapy consists of exogenous injections of insulin. Insulin has to be delivered to the organism whenever the patients have a meal to compensate for the lack of the endogenous one and to allow the correct management of the blood glucose. The quantity is delivered according to tables designed by the physician and refined using the patient's history and experience. In addition to this kind of administration, called *meal bolus* or *fast-acting*, the T1D-patients have to assume also the so-called *long-acting, background* or *basal insulin*. It gives the insulin needed for the organism, whether the occurrence of a meal or not, and it should keep the blood glucose stable overnight and between meals. The common way to deliver the insulin to the body is the use of needle and syringe, of the insulin pen or the insulin pump.

Figure 1.2: Insulin delivery methods most commonly used

1.1.4 Type 2 Diabetes (T2D)

Type 2 diabetes, called *non-insulin-dependent*, or *adult-onset*, results from the body's ineffective use of insulin. It can be due either to the resistance of the organism to the insulin or to the not enough production of it to maintain a normal glucose level. Type 2 diabetes comprises the majority of people with diabetes and is largely the result of excess body weight, physical inactivity and certain health problems such as high blood pressure. Type 2 diabetes is caused by several factors, including overweight and obesity, not being physically active, insulin resistance and genetic. The management of T2D includes diabetes medicines, as pills or as insulin injections, and blood glucose testing. It may be necessary to take medicines for high blood pressure, high cholesterol or other conditions. The treatment of T2D includes also the control of the diet, a correct physical exercise that helps the body use insulin and lower the blood sugar level, the maintaining of a healthy weight that helps insulin work better in your body, lower the blood pressure and decrease the risk for heart disease.

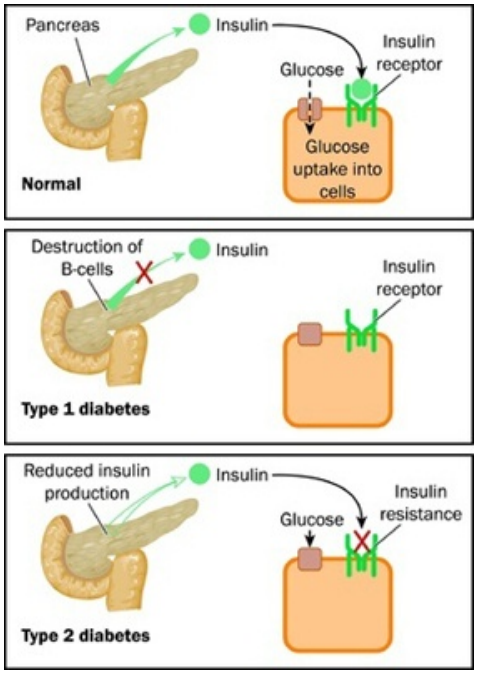

Figure 1.3: Insulin delivery methods most commonly used

1.2 Monitoring of Blood Glucose level

Monitoring blood glucose is fundamental in order to keep the glucose concentration within the euglycemic range $70 - 180\ mg/dl$ and then minimize the risk related to long and short term complications. In fact for diabetic people, most of the type 1 diabetes, it is fundamental to check the BG levels every time they have a meal or when they go to sleep or whenever they have physical activity and so more.

1.2.1 Self Monitoring of Blood Glucose level (SMBG)

The common and traditional method to read the glucose level is the *Self Monitoring of Blood Glucose (SMBG)* system, available from the seventies. The measurement consists of inserting in the device a reagent test-strip, in which a small sample of blood kept in the finger with a lancet is applied. Then the device determines the glucose concentration using different technologies, even if most systems measure an electrical characteristic proportional to the amount of glucose in the blood sample. These devices have a low error on the measures (5%<CV<10%) and they are easy to manage. The limits of this type of BG monitoring are the number of samples that a person can take from the finger in a day (7-8 maximum) and the fact that the

treatment decisions would be based only on a single number obtained with a SMBG.

Figure 1.4: Self Monitoring of Blood Glucose device

1.2.2 Continuous Glucose Monitoring system

Another way to assess the BG levels is the *Continuous glucose monitoring (CGM)* system, first appeared in 1999. CGM automatically tracks blood glucose levels every 1/3/5 minutes throughout the day and night for several days, allowing the diabetic people to check the BG level anytime and control its trend.
The CGM systems are composed by a sensor, a transmitter and a receiver. The sensor is applied on the patient skin, typically in the zone of the abdomen or in the arm, and it measures an electrical current proportional to the interstitial fluid glucose concentration. Then the transmitter, with a wireless connection, sends the information to a receiver device. Hence a sensor calibration procedure, that involves also some SMBG entries, processes the signal comes from the interstitial fluid to convert it into the actual CGM signal, that is the blood glucose concentration. The receiver device stores, processes, visualizes the information; some CGMs can also send information directly to a smartphone or tablet.

Figure 1.5: Continuous Glucose Monitoring device

The system also gives reviews on how the glucose changes over a few hours or days to see trends, it generates glucose direction and rate of change. Controlling glucose levels in real-time can help diabetic people make more informed decisions throughout the day about how to balance the meal, the physical activity and the medicines. CGM can contribute to better diabetes management by helping the patients minimize the guesswork that comes with making treatment decisions based solely on a number from a blood glucose meter reading.
Furthermore, many CGM systems have additional features that work with information from the glucose readings:

- An alarm can sound when the glucose level goes too low or too high, exceeding the normal range thresholds;
- It is possible to record the meals, the physical activity and the medicines in the CGM device, to have a complete vision of the patient's situation;
- The data can be downloaded in a computer or smart device to more easily see the glucose trends and to keep track of the history of the BG trends.

All the functionalities of the CGM devices allow diabetic people to be more informed on the real-time and long-term BG situation, achieving better management of the glucose levels and so to have fewer low blood glucose emergencies. In addition, also fewer finger sticks are necessary during the day.
Basing on studies [Castle and Jacobs, 2016],[Aleppo *et al.*, 2017], [Facchinetti, 2016] that have demonstrated the safety and the validity of the use of the CGM systems without confirmatory SMBG fingersticks (the so-called *nonadjunctive use*), the approval of this use to make treatment decisions has given to many CGM sensors, as the FreeStyle Navigator II, the FreeStyle Libre, the Dexcom G5 Mobile and the Dexcom G6.

On the other hand, glucose monitoring technologies have aroused the development of several applications such as predictive alerts, automatic basal insulin attenuation methods for hypoglycemia prevention, and artificial pancreas. The *Artificial Pancreas Device System* is a system of devices that closely mimics the glucose regulating function of a healthy pancreas.

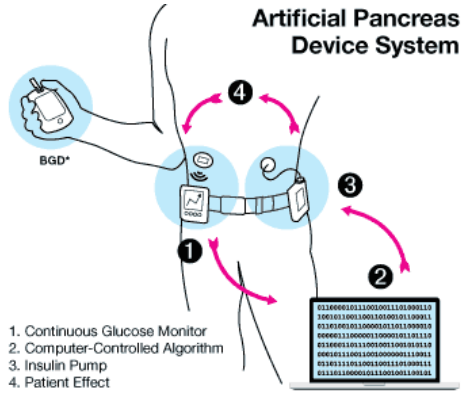

Figure 1.6: Artificial pancreas device system

In 2016, the U.S. Food and Drug Administration approved a type of artificial pancreas system called a *hybrid closed-loop system*. This system checks the glucose level every 5 minutes by the continuous glucose monitor device, and automatically computes and gives the right amount of basal insulin through a separate insulin pump. In this way, the system not only monitors glucose levels in the patient but also automatically adjusts the delivery of insulin to reduce high blood glucose levels and minimize the incidence of low blood glucose, with little or no input from the patient. To compute the correct action and dose of insulin, the fundamental role is given to a control algorithm software embedded in an external processor (the controller) that receives information from the CGM and performs a series of mathematical calculations. Based on these calculations, the controller sends dosing instructions to the infusion pump. The control algorithm can be run on any number of devices including an insulin pump, computer or cellular phone.

1.3 Glucose levels and events prediction based on CGM data

In the open-loop context, the availability of real-time frequent CGM acquisitions give the possibility of predicting glucose concentration from its only past history, as proven in [Bremer and Gough, 1999], [Sparacino *et al.*, 2007], [Reifman *et al.*, 2007]. The short-term prediction of the future glucose levels (ahead in time of 20-30 minutes) could be a fundamental part of the control algorithm that regulates the insulin dose. In fact, knowing what would be the future BG level, will allow the user to take some countermeasures to avoid dangerous situations. For example, in system in which the CGM sensor is coupled with an insulin pump, the prediction of an upcoming hypoglycemic event could lead to the suspension of the insulin delivery to avoid, or at least to mitigate, the duration of that event [Gillis *et al.*, 2007], [Bruttomesso *et al.*, 2009], [Kropff *et al.*, 2015].

A CGM-insulin pump system already available in the market that implements a control based also on the 30 minutes ahead in time glucose prediction is the MiniMed 640G from Medtronic [Zhong *et al.*, 2016]. The controller suspends the delivery of basal insulin if in the 30 minutes predicted windows the BG would go under the hypoglycemic threshold. This kind of approach, that uses Kalman-Filter to make the glucose level prediction, is demonstrated to be safe and effective in the prevention of hypoglycemia, even if there is the possibility of the increase of the hyperglycemia [Buckingham *et al.*, 2010], [Buckingham *et al.*, 2013], [Choudhary *et al.*, 2016].

Furthermore, another application of the BG prediction that does not include necessarily the use of an insulin pump is the generation of alarms 20-30 minutes before that an hypo/hyperglycemic event is predicted. In this way, the patient could himself takes some actions to avoid the predicted situation, basing also on his experience and personal knowledge of how his organism reacts.

Hypoglycemia management is the most limiting factor for T1D people, particularly during the night period or other periods when the individual is engaged in activities that take their focus away from glucose monitoring. Nocturnal hypoglycemia has been associated with increased morbidity and prolonged hypoglycemic episodes

(two to four hours) often precedes seizure activity in people with T1D. If extremely severe, the nocturnal hypoglycemic events have been associated with the sudden dead-in bed syndrome [Hsieh and Twigg, 2014], [Graveling and Frier, 2017]. On the other hand, the resultant fear of hypoglycemia may lead to treatment noncompliance, that can cause major long-term complications.

The algorithms developed and proposed in literature about these tasks are many. The following are aimed to predicting the BG concentration every time a new CGM sample is available:

- Polynomial or auto-regressive models (AR) of order 1 with recursive identification [Sparacino *et al.*, 2007];
- AR models with higher order [Gani *et al.*, 2009];
- Auto-Regressive moving average (ARMA) [Eren-Oruklu *et al.*, 2009];
- Auto-regressive integrated moving average (ARIMA) with adaptative orders [Yang *et al.*, 2018];
- Algorithms based on non-linear techniques as Artificial Neural Networks (ANNs), kernel-based methods, support vector machine and regression (SVM and SVR) [Pérez-Gandía *et al.*, 2010], [Bertachi *et al.*, 2018],[Georga *et al.*, 2013], [Georga *et al.*, 2015].

While the following are specifically focused on the prediction of the hypoglycemic events:

- Kalman Filter to predict glucose values under the threshold 70 mg/dl [Palerm and Bequette, 2007];
- Classification and Regression trees (CART) [Miyeon *et al.*, 2017];
- Ensemble method that includes many different algorithms to make the prediction [Dassau *et al.*, 2010];
- Linear discriminant function (LDA) combined with features selection to perform long-term prediction[Jensen *et al.*, 2019];

- Support Vector Regression [Georga *et al.*, 2013];
- Artificial Neural Network and Recurrent-NN [Bertachi *et al.*, 2018], [Doike *et al.*, 2018];
- Data Mining technique [Eljil *et al.*, 2013];
- Different ML algorithms as SVM, Random Forest, K-nearest neighbor and Logistic Regression [Seo *et al.*, 2019], [Gadaleta *et al.*, 2019]

These studies suggest that hypoglycemia can be predicted to enable people with diabetes to take preventive actions, such as adjusting basal insulin, and then it is possible to investigate the long-term prediction of the nocturnal hypoglycemia.
Several of these algorithms included in the features set information coming from other sources in addition to the CGM data. In fact, taking advantage of meal information, insulin dosing and physical activity information improve the performance in the glucose level prediction and on the hypoglycemic prediction [Zecchin *et al.*, 2016], [Wilson *et al.*, 2015].

Unfortunately in the CGM devices this functionally is not implemented yet, due to the lack of comparing studies on the performances of different developed algorithms on the same dataset. Some proof-of-concept studies have already been proposed to compare the functionality of several prediction algorithms as cited previously, but proof of the benefits that could be achieved on real data from prediction methods that aim to prevent/mitigate hypoglycemia are still unaccounted but necessary.

1.4 Aim of the thesis work

A prediction of a nocturnal hypoglycemia in the evening, before bedtime, would allow diabetic patients to adjust long-term glucose-affecting factors, as for example basal insulin. Even if nocturnal hypoglycemic episodes are simpler because in general meals are not consumed, insulin is not administered, and physical activity is absent, in reality they may occur at any time of the night and the prediction horizon thus varies, which makes the prediction difficult.

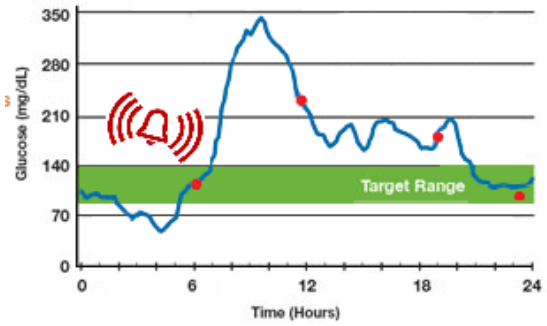

Figure 1.7: Example of a hypo alarm

The aim of this thesis work is investigating the possibility of forecast nocturnal hypoglycemic events using only the CGM data of the previous day, in order to understand if the glucose concentration information alone could be predictive of the nocturnal hypoglycemia. The algorithms chosen to solve this task are taken from the machine-learning framework, that in presence of sufficient samples-size should be suitable to learn patterns in the data and thus to solve the problem. The main advanteges of developing such an algorithm are:

- the possibility of implementing this method on all diabetic people who use the CGM system, despite being under multiple daily insulin
- the simplicity of the prediction setting, that makes the approach more easily generalizable than more complex methodologies exploiting also meal and insulin data, which are unlikely available in real-time.

On the other hand, this task is made difficult to solve by the following factors:

- the impossibility of taking advance of other fundamental information in the awareness of the patient situation as the meal or insulin information;
- the complexity of the hypoglycemia occurrence and the inter-person variability of the factors that caused it;
- the relatively short period of CGM acquisition to learn the task that is limited to the day before the night in which make the prediction only, that may not be enough to understand the hypoglycemia occurs.

In the next chapters the datasets used for the work, the features and the methods chosen to make the prediction are described. Finally, the results and further approaches are illustrated, together with a discussion on the work outcomes.

CHAPTER 2

Datasets and Pre-Processing

This chapter introduces the datasets chosen for the analysis and the methodologies used to processing the data to make them suitable to perform the task. The pre-processing procedures are implemented with the software *Matlab R2018a*.

2.1 Description of the Datasets

2.1.1 ReplaceBG Dataset

This dataset derives from the ReplaceBG study [Aleppo *et al.*, 2017] . This study is about a randomized trial performed to determine whether the use of the Continuous Glucose Monitoring (CGM) without confirmatory blood glucose monitoring (BGM) measurements is as safe and effective as using CGM adjunctive to BGM in well-controlled T1D adult subjects. The randomized clinical trial was conducted in 14 sites belonging to the T1D Exchange Clinic Network in the U.S., the patients had a minimum age of 18 years, they had T1D for 1-year minimum and they were treated with insulin pump for at least 3 months. The ratio of the CGM-only and CGM+BGM groups is 2:1 and both used a *Dexcom G4 Platinum CGM System* to measures glucose concentrations. The final number of participants is 224: 148 in the CGM-only group and 76 in the CGM+BGM group. For the purpose of this work that is not focused on

this distinction, patients are considered equal and then treated in the same manner. The trial lasted 26-week, that means about six months of acquisition. In Table 2.1 the principal outcomes of the study are reported.

CGM results	**CGM-only (n=148)**	**CGM+BGM (n=76)**
Hours of CGM data	4007 (3709-4166)	4021 (3725-4136)
% Time in target (70-180 mg/dl)	63±13	65±11
Mean Glucose	162±23	158±20
Hypoglycemia		
%Time $< 70mg/dl$	3.0 (1.6-5.1)	3.7 (1.9-4.9)
%Time $< 60mg/dl$	1.3 (0.5-2.4)	1.6 (0.6-2.2)
%Time $< 50mg/dl$	0.3 (0.1-0.6)	0.4 (0.2-0.5)
%Days with ≥20 consecutive min-glucose val $< 60mg/dl$	28 (13-42)	32 (16-46)

Table 2.1: Outcomes of the trial, data are median (interquartile range) or mean±SD

2.1.2 Real Dataset

This dataset includes 6-months of CGM records of 25 T1D patients and it includes also the information about the assumption of alcohol and carbohydrates, the physical activity and the insulin dosing. The dataset has been recorded for a study that is not published.

2.2 CGM data preprocessing

Because failures and irregularities are present during data acquisition, the first necessary step is to align the samples in a fixed grid to create an homogeneous dataset and to compute all the necessary features accurately. The grid contains an acquisition every 5 minutes. The samples are assigned to the point of the grid nearest to their times of acquisition and if a sample is missing the value NaN is placed. In this way, an equally-spaced dataset is obtained.

2.3 Creation of the day and night CGM vectors

Data have been divided in samples of 24 hours each. Sample values are labelled as "day" or as "night". "Night" is defined as the period from 23:00 to 6:55 that are 8 hours (96 samples), so the day is from 7:00 to 22:55 that are 16 hours (192 samples). A "day" of data is kept only if there is a corresponding "night" of data. Values are then saved in separate vectors, one vector corresponding to "day" data and one corresponding to "night" data for each sample.

2.4 Definition of the hypoglycemic event and creation of the label vector

To implement a supervised machine-learning method, each sample of the dataset has to contain the features needed to train the algorithm and the label. In this task the label of a sample is the information about the presence of the nocturnal hypoglycemic event or the absence of it in the "night" vector. Therefore to assign the label to each sample, it is necessary to define the occurrence of an hypoglycemic event. There are many possible definitions of a nocturnal hypoglycemic event:

- the presence of at least one CGM value under the hypoglycemic-threshold during the night period;
- the presence of at least p CGM values under hypoglycemic-threshold during the night period, not necessarily consecutive;
- the presence of at least p consecutive CGM values under hypoglycemic-threshold.

Among these options, the widely accepted definition of nocturnal hypoglycemia is the presence of at least 3 consecutive samples (corresponding to 15 minutes) under the hypoglycemic-threshold, fixed to be 70 mg/dl.
For the creation of the label vectors, not all the samples can be used. First of all, a selection based on the number of missing values present in the day and night vectors is made. The scenarios can be:

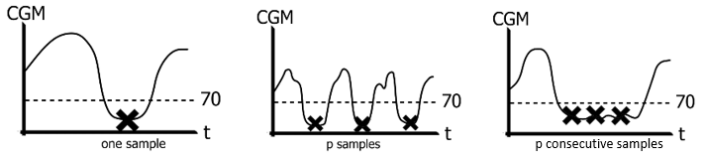

Figure 2.1: Representation of the possible definitions of an hypoglecymic event, $hypo - threshold = 70\ mg/dl$

- the "day" vector has less than 6 consecutive NaN values and also the "night" vector $\implies$ the sample *is* valid
- the "day" vector has less than 6 consecutive NaN values, the "night" vector has more than 6 consecutive NaN values but the hypoglycemic event is however present $\implies$ the sample *is* valid
- the "day" vector has less than 6 consecutive NaN values, the "night" vector has more than 6 consecutive NaN values and the hypoglycemic event is not present $\implies$ the sample *is not* valid
- the "day" vector has more than 6 consecutive NaN values $\implies$ the sample *is not* valid

The choice of the number of consecutive NaN acquisitions equal to 6 is made because they correspond to at least 30 minutes of no information about the CGM trend and so, in the case of a "day" vector, it is not possible to have a correct description of that samples (Figure 2.2). If the "night" vector has more than 6 consecutive NaN values, but the hypoglycemic event is present, it is possible to keep the sample because the information that the NaN values would give is not necessary to label the sample; instead if there is not the event, it could be that it would be in the period of no acquisition, so the sample has to be discarded. For the NaN values in the kept vectors, a *linear interpolation* is performed, avoiding possible negative values and interpolating the possible NaN in the extremities keeping the successive or previous value in case of NaN in beginning or at the end of the vector. The interpolation is made both in the "day" vector, to have a better computation of the features later, and in the "night" vector only if the hypoglycemic event has not been found yet, to

understand if it could be in the period of no acquisition.

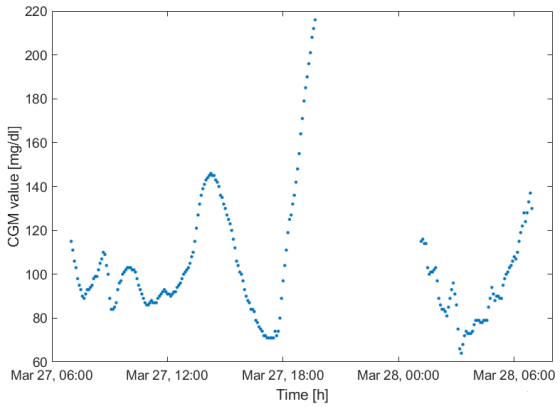

Figure 2.2: CGM curve of a sample to discard that have more than 6 consecutive NaN

The second selection is made to avoid the case in which the nocturnal hypoglycemic event actually has been started in the "day" period, and so it is not correct to classify it as a nocturnal hypoglycemic event. Furthermore, if the hypoglycemia has been started when the patient was awake, he may have taken some countermeasures and then the classification could be distorted. Therefore the cases could be:

- the patient has at least one CGM record under the hypoglycemic-threshold of 70 mg/dl in the half an hour preceding the night (22:30-22:55 period);
- the patient has an hypoglycemic event in the first hour of night (23:00-23:30 period) and has at least a CGM value under 80 mg/dl in the half an hour preceding the night.

If a sample satisfies one of these conditions it has to be removed from the dataset. In Figure 2.3 there are two examples of the cases explained above.

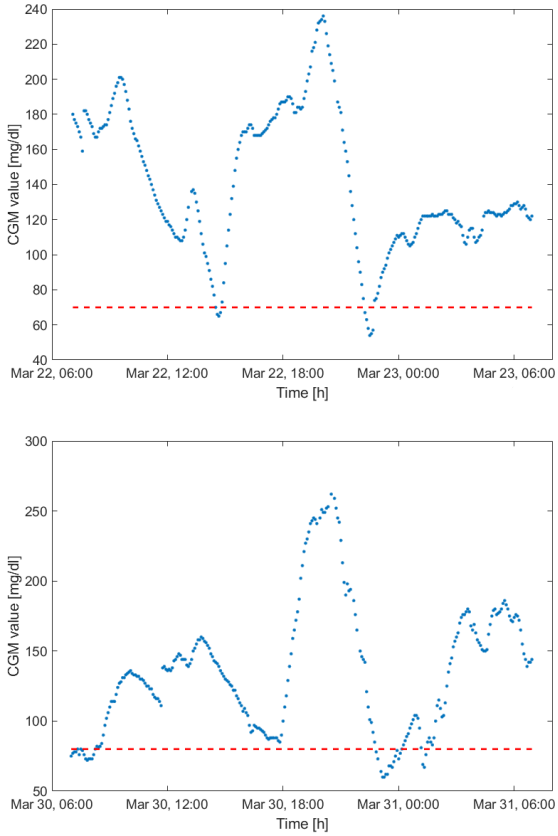

Figure 2.3: CGM curves of two samples discarded for the first and second condition

After all the checks, the valid samples are stored, including the "day" vector, the "night" vector, the label. Some examples of CGM curve are presented above. In figure 2.4 a curve of a whole day of a not hypoglycemic sample and in figure 2.5 the curve of a whole day of an hypoglycemic sample are shown.

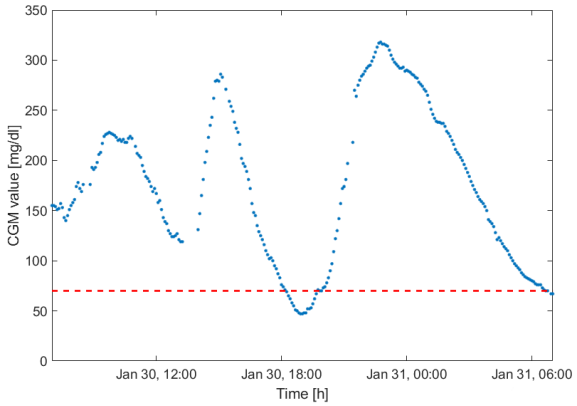

Figure 2.4: CGM curve of a not hypoglycemic sample

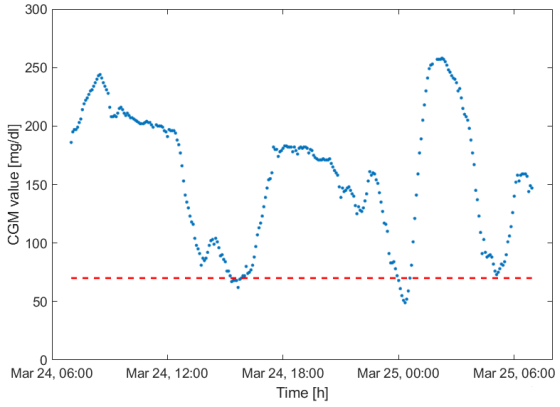

Figure 2.5: CGM curve of an hypoglycemic sample

In the end, it is possible to compute the total number of samples and the proportion of the two classes for the two datasets analyzed. In Table 2.2 are shown the number of total samples, hypo and not-hypo samples for the two datasets and in figure 2.6 there are the pie-charts of the classes proportion.

	ReplaceBG dataset	**real dataset**
Number of patients	224	25
Number of total samples	25386	2044
Number of Hypoglycemic samples	4394	414
Proportion of the Hypoglycemic class	17.30%	20.74%

Table 2.2: Composition of the two datasets

Furthermore, the hours in which the hypoglycemic events occur are stored, to evaluate if they are equally distributed during the night or if there exist some period in which they are more frequent. From the plot of the histogram of the number of events that occur for all the hours of the night it is possible to conclude that for the ReplaceBG dataset there is not a period of night with the majority of the events but they are homogeneously distributed during all the night period, except for a larger number of events in the hours 0:00 and 1:00. In the Real dataset, there are more events in the period from the 0:00 to 1:00 and less in the final period of the night.(Figure 2.7).

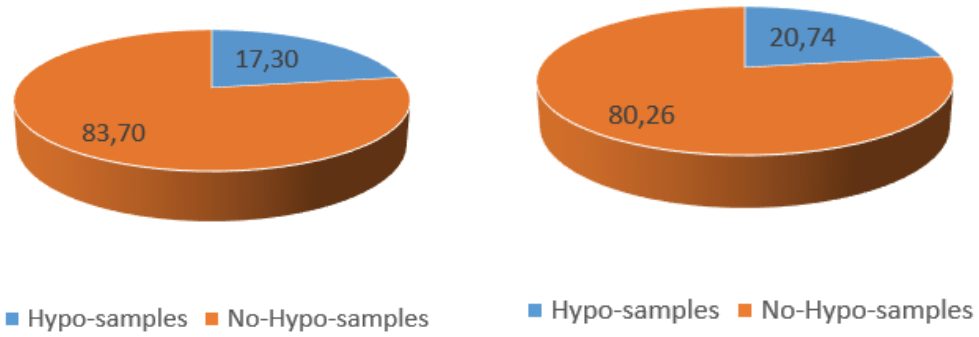

Figure 2.6: Pie-charts of the classes proportion of the ReplaceBG dataset on the left and of the Real dataset on the right

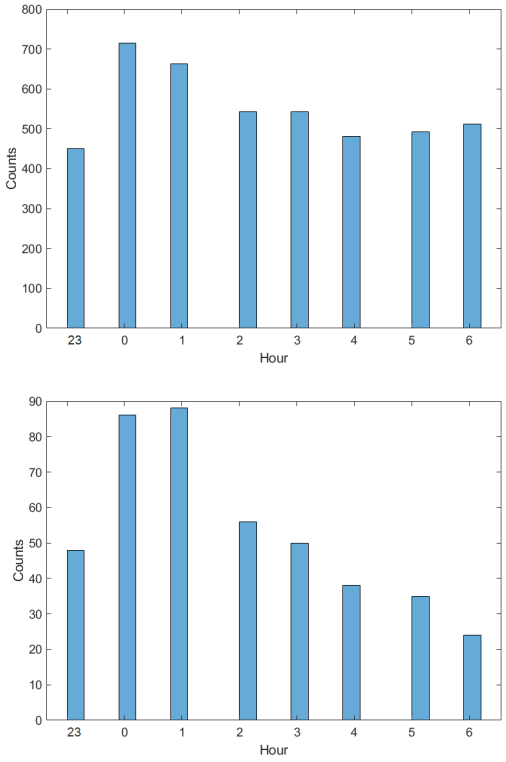

Figure 2.7: Histograms of the counts of the events in each night hour, ReplaceBG and Real dataset

2.5 Definition of the features set

To determine what features of the CGM trace are the most appropriate to be used for the prediction of the nocturnal hypoglycemic events, a bibliographic research is conducted. Because most of the studies aim to predict hypoglycemic events in

the short-term, some features used are the last value before the prediction horizon, the derivative of the CGM curve in the period preceding the time of the prediction [Miyeon *et al.*, 2017], [Choleau *et al.*, 2002], the CGM data of a window preceding the time of prediction and other indices obtained from the CGM data [Palerm and Bequette, 2007], [Seo *et al.*, 2019],[Gadaleta *et al.*, 2019], features obtained from physiological models [Bertachi *et al.*, 2018]. Other features that can correlate with the hypoglycemia and so potentially used for this kind of task are the *Glycemic Variability (GV)* indices. These indices are measures of the trend and evolution of the glycemia; they are demonstrated to be correlated with the glycemic control and in particular with the hypoglycemia and hyperglycemia [Kovatchev *et al.*, 2000a],[Kilpatrick *et al.*, 2007], [Rodbard, 2009], [El-Laboudi *et al.*, 2016]. In some studies they are used as features for tasks similar to the prediction of the nocturnal hypoglycemic events, therefore they can represent a way to assess the variability of the CGM signal and allow using this information to predict the events [Cox *et al.*, 2007], [Chandran *et al.*, 2018], [Rodbard, 2012]. Furthermore, a selection of specific GV indices can be more predictive of the hypoglycemia, so 17 indices are chosen for the specific task [Saisho *et al.*, 2014], [Fabris *et al.*, 2014]. These indices are computed using the whole day CGM signal and are defined as:

- *Mean ($MEAN_{tot}$)*: mean value of the CGM records of the total day period;
- *Standard Deviation (SD_{tot})*: standard deviation of the CGM records of the total day period;
- *Coefficient of Variation (CV_{tot})*: this index is the coefficient of variation of the CGM records of the total day period;
- *Median ($MEDIAN_{tot}$)*: median of the CGM records of the total day period;
- *Interquartile Range (IQR_{tot})*: interquartile range of the CGM records of the total day period, defined as the difference between the third and first quartiles;
- *Range ($RANGE_{tot}$)*: range of the CGM records of the total day period, defined as the mean of the maximum and minimum records of the period;

- *Blood Glucose below target (BG_{below})*: percentage of records of the total day period that are under the hypoglycemic-threshold of 70 mg/dl;

- *Blood Glucose in target (BG_{in})*: percentage of records of the total day period that are in the euglycemic interval $70 - 180$ mg/dl;

- *Mean Amplitude of Glycemic Excursion-ascendant ($MAGE_+$)*: this index is defined as the mean amplitude of the excursion of two CGM records that differ more than the SD of the total day, with the first CGM value lower than the second;

- *Mean Amplitude of Glycemic Excursion-descendant ($MAGE_-$)*: this index is defined as the mean amplitude of the excursion of two CGM records that differ more than the SD of the total day, with the first CGM value higher than the second. With the index $MAGE_+$, they are indicators of glycemic instability [Service *et al.*, 1970];

- *Excursion Frequency (EF)*: number of total excursion between two samples that is wider than 75 mg/dl in the total day period;

- *Low Blood Glucose Index (LBGI)* and *High Blood Glucose Index (HBGI)*: these indices are defined as glycemic control indices that take into consideration the non-symmetry of the range of glycemic values that a person can have [Kovatchev *et al.*, 2000b]. This asymmetry results in different health-risks of glycemic values distant the same quantity from the lower euglycemic-threshold and from the higher euglycemic-threshold. For examples if a person has a value of glycemia equal to 40 mg/dl, that is 30 mg/dl under the eu-threshold, is clearly more in danger in that moment with respect to one that has a value equal to 210 mg/dl, that is the same quantity 30 mg/dl but above the eu-threshold. For handling this problem in the evaluation of the glycemic variability, a transformation on the glycemic range $20 - 600 mg/dl$ is made in order to make it symmetric.

  $$f(BG, \alpha, \beta) = ln(BG)^{\alpha} - \beta$$

  with α and β defined to respect some characteristic in the transformation.
  Then a risk function is defined on this transformation to evaluate the possible

danger of a glycemic value (Figure 2.8).

$$r(BG) = 10 * f(BG)^2$$

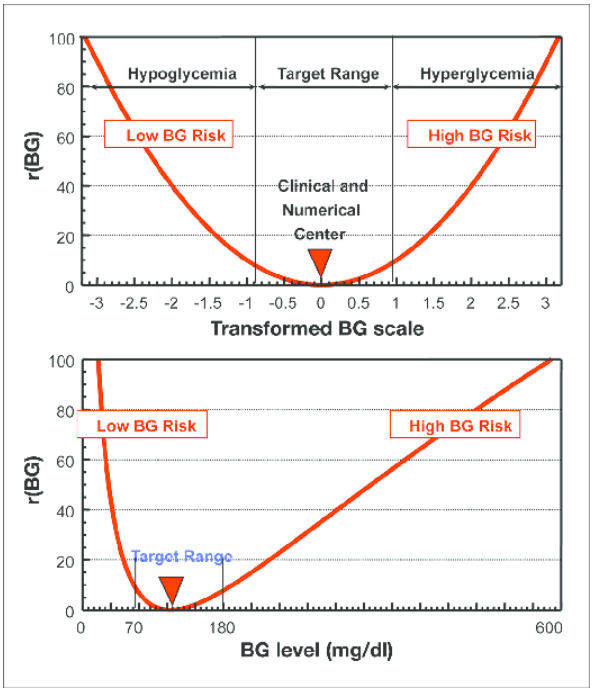

Figure 2.8: Risk function in the transformed and original glycemic range

The two indices are not-negative, the LBGI is the index that considers the risk of glycemic values under 112.5 mg/dl, that is the zero of the transformed range, instead the HBGI considers the risk of glycemic values above 112.5 mg/dl, so values higher than zero in the transformed range. The LBGI increases when the number and/or the values in the hypoglycemic zone of the range increase and vice versa the HBGI increases when the number and/or the values in the hyperglycemic zone of the range increase.

These indices are known to be predictive of the adverse events that can happen to the diabetic patients and they are used as control quantity.

- *Hypoglycemia and Hyperglycemia index* ($HYPO_{ind}$, $HYPER_{ind}$): these two indices are very flexible because they contain parameters that can be varied to weight more some events with respect to other [Rodbard, 2009]. Furthermore, also the euglycemic-thresholds can be modified to be more restrictive or more tolerant on the glycemic range. In this task they are kept as the standard values $70 - 180$ mg/dl.

- *Glycemic Risk Assessment Diabetes Equations ($\%GRADE_{hypo}$, $\%GRADE_{hyper}$)*: these two indices are based on a risk function of the glycemic values too [Hill *et al.*, 2007]. To derive the function, different risk-weights are assigned by 50 diabetes professionals to possible values of glycemia. Then this function is used to compute the two indices that assess the risk of a glycemic profile.

The first six metrics, the $MEAN$, SD, CV, $MEDIAN$, IQR and $RANGE$ are computed also for the CGM records of the periods of the 3 hours and one hour preceding the night period, so for the time intervals from 20:00 to 22:55 and from 22:00 to 22:55.
In the end, in addition to all these features, other five metrics are considered:

- *Mean of the two half an hour preceding the night (MEAN-HALF-1, MEAN-HALF-2)*: the means of the CGM records of the half an hour from 22:00 to 22:25 and of the half an hour from 22:30 to 22:55;

- *Derivative (DER)*: the slope of the straight line obtained interpolating the last four CGM records preceding the night;

- *Last CGM value (LAST-DAY-VALUE)*: the last day CGM value, i.e the CGM value at 22:55;

- *Flag on the previous label (FLAG-PREV-LABEL)*: a 0-1 flag on the presence of the hypoglycemic event in the previous night, if the information is not available the value assigned to this feature is 0.5.

In Table 2.3 a summary of all the features used with their definitions is reported. After the creation of the effective dataset to use, the choice of the appropriate methodologies to solve the task is made, as will be reported in chapter 3.

Feature Number	Definition	Feature Number	Definition
1	$m_{tot} = \frac{1}{N}\sum_i CGM_i$	**18**	$MEAN_3$
2	$SD_{tot} = \sqrt{\frac{\sum_i (CGM_i - m_{tot})^2}{N-1}}$	**19**	SD_3
3	$CV_{tot} = \frac{SD_{tot}}{m_{tot}}$	**20**	CV_3
4	$MEDIAN_{tot} = CGM\left(\frac{N+1}{2}\right)$	**21**	$MEDIAN_3$
5	$IQR_{tot} = Q_3 - Q_1$	**22**	IQR_3
6	$RANGE_{tot} = \frac{CGM_{max} - CGM_{min}}{2}$	**23**	$RANGE_3$
7	$BG_{below} = \frac{100}{N}\#(CGM < 70)$	**24**	$MEAN_1$
8	$BG_{in}\frac{100}{N}\#(70 \leq CGM \leq 100)$	**25**	SD_1
9	$MAGE_+ = \frac{\sum_{i=1}^{n_{e+}} \Delta_i(CGM,SD)}{n_{e+}}$	**26**	CV_1
10	$MAGE_- = \frac{\sum_{i=1}^{n_{e-}} \Delta_i(CGM,SD)}{n_{e-}}$	**27**	$MEDIAN_1$
11	$EF = \#\Delta_{>75mg/dl}$	**28**	IQR_1
12	$LBGI = \frac{\sum_i r_l(CGM_i}{N}$, $r_l = 22.7 * f(CGM_i)^2$ $if\ f(CGM_i) < 0\ and\ 0\ otherwise$	**29**	$RANGE_1$
13	$HBGI = \frac{\sum_i r_h(CGM_i}{N}$, $r_h = 22.7 * f(CGM_i)^2$ $if\ f(CGM_i) < 0\ and\ 0\ otherwise$ $f(CGM_i) = ln(CGM_i)^{2.084} - 5.381$	**30**	MEAN-HALF-1
14	$HYPO_{index} = \frac{\sum_{i=1} N(70 - CGM_i)^b}{N*d}$	**31**	MEAN-HALF-2
15	$HYPER_{index} = \frac{\sum_{i=1} N(CGM_i - 180)^a}{N*c}$ $a = 1.1, b = 2, c = d = 30$	**32**	DER
16	$GRADE_{hypo} = 100 * \frac{\sum risk_{BG}(CGM<70)}{\sum risk_{BG}}$	**33**	LAST-CGM-VALUE
17	$GRADE_{hyper} = 100 * \frac{\sum risk_{BG}(CGM>180)}{\sum risk_{BG}}$	**34**	FLAG-PREV-LABEL

Table 2.3: Feature set used for each of the two datasets

CHAPTER 3

Methodologies for the hypoglycemic-events prediction

In this chapter, all the methodologies implemented to solve the task of the nocturnal-hypoglycemic events prediction are presented. Five algorithms have been included in our analysis: *Logistic Regression*, *Logistic Regression with L2 and L1 regularization*, *Support Vector Machine*, *Gradient Boosted Decision Trees*. All of them have been implemented using *Python* as programming language, using the $Scikit-learn$ library for the first four algorithms and the *XGBoost* (*Extreme Gradient Boosting*) library for the last. The first three algorithms, Logistic Regression without, with L2 and with L1 regularization, are generalized linear models; the Support Vector Machine algorithm is implemented with a non-linear kernel function; the Gradient Boosted Decision Trees is a model form of an ensemble of weak prediction models, the decision trees.

3.1 Logistic Regression

Logistic Regression is a statistical method used to model the probability of a certain class or event existing using some predictor variables. The predictor variables $\mathbf{x_j}$, $j = 1, ..., d$ with d the number of variables, are supposed to be relevant to predict the outcome; the outcome y of the method is a dichotomous variable (i.e. the label), so it contains data coded as 1 (*True, hypoglycemic sample*) or -1 (*False, not hypoglycemic sample*). Instead of directly modeling the outcome, the method models the *logit* transformation of the probability p of the event $y = 1$. The logit transformation is

called *log odds(Y)*, that is the logarithm of the ratio (*odds*) of the probability p of the event $y = 1$ on the probability $(1 - p)$ of the event $y = -1$. Hence the logistic model solves:

$$logit(p) = log\ odds(Y) = ln\left(\frac{p}{1-p}\right) = \langle \mathbf{w}, \mathbf{x} \rangle$$

where $\mathbf{w}$ are the estimated model coefficients of the predictor variables.
Therefore the hypothesis class is the composition of a sigmoid function $\phi_{sig} : \mathbb{R} \to [0, 1]$, the *logistic function*, over the class of *linear function* L_d. The logistic function is defined as $\phi_{sig}(z) = \frac{1}{1+e^{-z}}$, it's an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits (Figure 3.1). The hypothesis class is :

$$H_{sig} = \phi_{sig} \circ L_d = \{\mathbf{x} \mapsto \phi_{sig}(\langle \mathbf{w}, \mathbf{x} \rangle) : \mathbf{w} \in \mathbb{R}^d\}$$

.

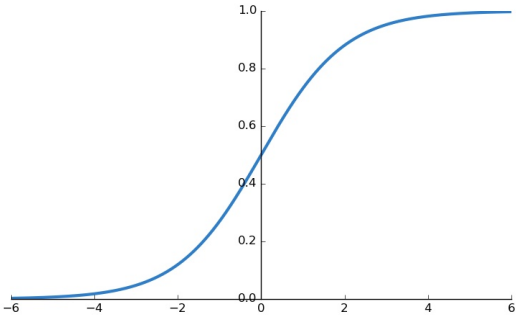

Figure 3.1: Representation of the *logistic function*

It is interesting to notice that when the internal product $\langle \mathbf{w}, \mathbf{x} \rangle$ is very large the result of the application of the sigmoid function is close to 1, whereas if $\langle \mathbf{w}, \mathbf{x} \rangle$ is very small then the result is close to 0. In these cases, whenever $|\langle \mathbf{w}, \mathbf{x} \rangle|$ is large, the prediction is very similar to the one of the *halfspace* hypothesis; instead when $|\langle \mathbf{w}, \mathbf{x} \rangle|$ is close to 0, then $\phi_{sig}(\langle \mathbf{w}, \mathbf{x} \rangle) \approx \frac{1}{2}$. This means that in this case, the logistic hypothesis is not sure about the value of the label, so it guesses that the label is the sign of $|\langle \mathbf{w}, \mathbf{x} \rangle|$ with probability slightly larger than 50%. In contrast, the halfspace hypothesis always outputs a deterministic prediction of either 1 or -1, even if $|\langle \mathbf{w}, \mathbf{x} \rangle|$ is very close to 0.

Besides, because of the output of the model is a probability, a key aspect in Logistic Regression is set the probability-threshold for determining in which class assign the samples. A standard choice is a threshold equal to 0.5 but for different type of data, a different choice could explain better the class distribution. A way to set an optimal probability-threshold is analyzing the *Receiver Operating Characteristic* curve, that plot some trade-off metrics at the variation of the probability-threshold and allow finding the best point that gives good results in both metrics.

To define the *loss function*, it is important to take in consideration that one would like that the probability of predicting 1 (that is $h_{\mathbf{w}}(x)$) would be large if the label is 1 and that the probability of predicting -1 (that is $1 - h_{\mathbf{w}}(x)$) would be large if the label is -1. So considering as loss function:

$$l(h_{\mathbf{w}}(\mathbf{x}, y)) = \frac{1}{1 + e^{y\langle \mathbf{w}, \mathbf{x} \rangle}}$$

it is clear that if the label is 1 then the loss would be the probability of the label -1 ($1 - h_{\mathbf{w}}(x)$), that one would have small, and vice versa for the case of label -1. Hence the loss function has to increase monotonically with $\frac{1}{1+e^{y\langle \mathbf{w}, \mathbf{x} \rangle}}$, that is equivalent to increase monotonically with $1 + e^{-y\langle \mathbf{w}, \mathbf{x} \rangle}$. A good modification of this loss to have a better formulation is the application of the logarithm (that is a monotonic function), that gives a convex loss function with respect to $\mathbf{w}$. In this way the *Empirical Risk Minimization (ERM)* problem associated to the Logistic Regression, that is the minimization of the empirical risk (the cost function) on some training set $S = ((\mathbf{x_1}, y_1), .., (\mathbf{x_m}, y_m))$, can be solved efficiently with different standard method.

$$ERM\ problem : \hat{\mathbf{w}} = \underset{\mathbf{w} \in \mathbb{R}^d}{\arg\min} \left(\frac{1}{m} \sum_{i=1}^{m} log(1 + e^{-y_i \langle \mathbf{w}, \mathbf{x_i} \rangle}) \right)$$

In the Python-Scikit Learn implementation of the Logistic Regression, it is possible to add a regularization term but in this first implementation no regularization is added to the cost function to minimize. The library includes different possible solvers for the cost function minimization and the chosen is the *Newton-CG*. This solver is analogous to the *gradient descent* iterative algorithm, but in the approximation of the

cost function at each step it implements the quadratic approximation using both the first and second partial derivative of the function.

Logistic Regression has been chosen because of its hypothesis of use, in fact the predictor variables chosen are supposed to be relevant for the prediction of the nocturnal-hypoglycemic events. Furthermore, the relationship between them is possibly linear and the Logistic Regression is a good starting point to approach the task. The presence of some multicollinearity in the predictor features is not a problem for the training of the algorithm, in this case the number of samples is large and permits to compensate this multicollinearity. However, it is proper to not consider the amplitudes of the coefficients of each predictor as a measure of the importance of it in the prediction.

.

3.2 Logistic Regression with L2-regularization

In the implementation of the Logistic Regression using a regularization term, there are different choices of possible regularize-function to minimize jointly with the cost function. The L2-regularization is made by adding a term that penalizes the complexity of the predictor, to avoid the over-fitting on the training process. In the cost function to minimize, to solve the so-called *Regularized Loss Minimization (RLM)* problem, it is added the l_2 *norm* of the parameters vector $\mathbf{w}$:

$$RLM\ problem: \hat{\mathbf{w}} = \underset{\mathbf{w}\in\mathbb{R}^d}{\arg\min} \left(\frac{1}{m} \sum_{i=1}^{m} log(1 + e^{-y_i \langle \mathbf{w}, \mathbf{x_i} \rangle}) + \lambda \|\mathbf{w}\|_2 \right)$$

where $\lambda > 0$ is a scalar hyper-parameter that controls the weight of the regularization term on the minimization and does not depend on data. The logic under the choice of this regularization function is that the algorithm balances between low empirical risk on the training set, that can lead to over-fitting, and "less-complexity" of the predictor.
To tune the hyper-parameter λ, in the Scikit-Learn library of Python there is an implementation of the Cross-Validation (CV) Grid Search, called *LogisticRegressionCV*.

In this way it possible to try a grid of different regularization hyper-parameters, set a number K of CV-folds, define the score to minimize and, chosen the solver and the type of regularization, get the best value of λ for the model. Specifically in this Python-implementation, the parameter of the regularization is the inverse of the regularization-strength, that is the parameter $C = \frac{1}{\lambda}$. Lower values of C indicate stronger regularization and bigger value indicate lower regularization.
It is important to notice that the CV is implemented dividing the training samples in K different folds that keep the percentage of samples for each class, that is the *stratified K-folds Cross Validation*.
The solver chosen is the *Newton-CG* also in this case, that is the same but with the addition of the regularization term in the cost function.

3.3 Logistic Regression with L1-regularization

The regularization with the L1-norm of the model parameters vector $\mathbf{w}$ is another possible way to improve the performance of the Logistic Regressor. The new RLM problem is:

$$RLM\ problem: \hat{\mathbf{w}} = \underset{\mathbf{w}\in\mathbb{R}^d}{\arg\min} \left(\frac{1}{m} \sum_{i=1}^{m} log(1 + e^{-y_i \langle \mathbf{w}, \mathbf{x_i} \rangle}) + \lambda \|\mathbf{w}\|_1 \right)$$

This type of regularization induces "sparse" solutions, that is $\hat{\mathbf{w}}$ with some components equal to zero. If a predictor variable x_i would have $w_i = 0$, it means that this characteristic is not descriptive of the data, equivalently that the feature is not enough correlated with the labels vector y. Also in this case, the algorithm has been implemented using the *LogisticRegressionCV* to find the optimal value of the parameter $C = \frac{1}{\lambda}$ that controls the trade-off between model-complexity and capacity of explaining training data.
To find the solution of the minimization problem another solver is used, the *LibLinear*. This solver allows the use of the L1-regularization (*Newton-CG* does not); it uses a

coordinate descent algorithm that means that at each iteration it minimize along one coordinate direction keeping the other fixed, following some coordinate selection rule.

3.4 Support Vector Machine

The Support Vector Machine (SVM) classifier is a learner of linear predictors in high dimensional feature spaces that employs hyperplanes to make the classification of the samples. This kind of models is defined *halfspaces*. The objective of SVM is finding the hyperplane that allows distinguishing the classes and that maximizes the separation (i.e. the *margin*) between them. The margin of a hyperplane with respect to a training set is defined as the minimal distance between a point and the hyperplane. Therefore from all the possible separation- hyperplane, the chosen is the one that maximizes the distance from it to the nearest data points on each side. The nearest data points are called *support vectors* (the circled data points in Figure 3.2) and they are used for the margin maximization. The idea under this approach is that the maximization of the margin distance provides some reinforcement so that future data points can be classified with more confidence.

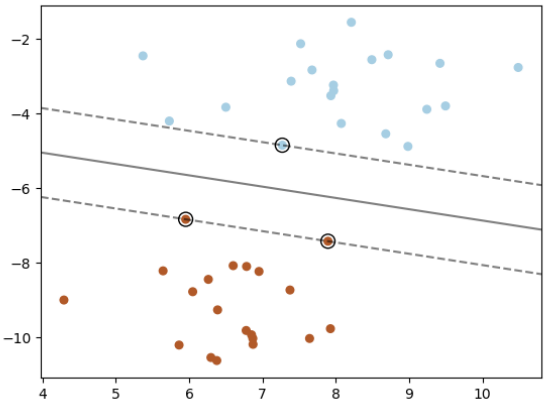

Figure 3.2: Representation of the separation-hyperplane and of the support vectors ($d = 2$)

The problem to solve is often not linearly separable and so a "soft" implementation of this concept is chosen. The idea is that the hyperplane to find has to separate the classes, so in the cost function, a constraint on the correctness of the classification is

present to force all example data to be in the correct side of the space (i.e. beyond the margin of its class). But, considering the relaxation on the linearly separability of the classes, some errors in the classification are tolerated, then some data points in the training set could violate the constraint (Figure 3.3). This concept can be implemented giving a constraint of being inside the space defined by the margin of its class fixed by the hyperplane, but with some measure of error given by the so-called *slack variables* ϵ_i.

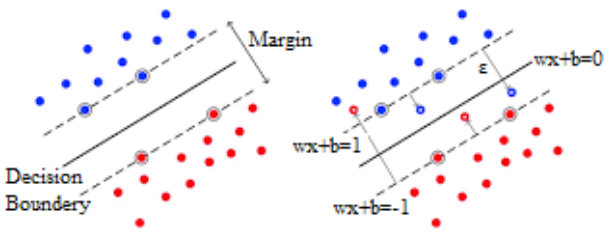

Figure 3.3: Representation of the hard and soft approach

To avoid that this relaxation leads to completely not correct classification, in the cost function a term that controls the amplitude of these errors is added as the mean of the slack variables. The trade-off between the correctness of the classification in term of maximization of the margin and the possible violation of the constraint is controlled by the hyper-parameter $C > 0$ in the cost function. Given as input the training set $(\mathbf{x_1}, y_1)$,...,$(\mathbf{x_m}, y_m)$, the final *soft-SVM* problem is define as finding $(\mathbf{w}, b)$ that solve:

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\epsilon}} \quad & \left(\|\mathbf{w}\|_2 + C\frac{1}{m}\sum_{i=1}^{m}\epsilon_i \right) \\ \text{s.t.} \quad & \forall i,\ y_i(\langle \mathbf{w}, \mathbf{x_i} \rangle + b) \geq 1 - \epsilon_i \ and \ \epsilon_i \geq 0 \end{aligned} \tag{3.4.1}$$

Furthermore, the SVM algorithm allows handling the cases of sets (the majority indeed) that are not linearly separable in the original feature space, using the so-called *kernel trick*. This implementation enriches the expressive power of the halfspaces without explicitly handling a mapping Φ of the data into a high dimensional feature space that would raise the computational complexity. The embedding of the data into other dimensional feature spaces, mostly of a higher dimension, would make the

halfspaces capable of dividing the classes in that space (Figure 3.4) but it also would make the computation heavy and requiring of many samples to learn the halfspaces.

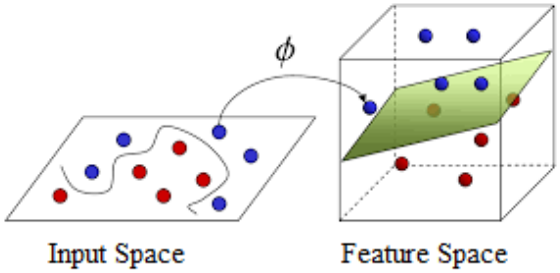

Figure 3.4: Representation of the non linearly separability and of the feature mapping

The *kernel* is a function that represents an inner product in the feature space and can be seen as a specifying similarity between instance vectors, realized as an inner product in the feature space that one would have obtained with a mapping of the data. The main advantage of the use of the kernels is that many learning algorithms for halfspaces can be implemented with just the computation of the values of the kernel function over pairs of domain points, as in the SVM. In this way, it is possible to implement linear separators in high dimensional feature spaces without having to specify points in that space or expressing the feature map explicitly.
The dual form of the problem to solve that specifies the use of the kernel is:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \left(\boldsymbol{\alpha}^T Q \boldsymbol{\alpha} - e^T \boldsymbol{\alpha}\right) \\ \text{s.t.} \quad & \forall i,\ 0 \leq \alpha_i \leq C \ and \ y^T \boldsymbol{\alpha} = 0 \end{aligned} \tag{3.4.2}$$

where $\mathbf{e}$ is a vector of ones, Q is a nxn positive semidefinite matrix such that $Q_{ij} = y_i y_j K(\mathbf{x_i}, \mathbf{x_j})$ and $K(\mathbf{x_i}, \mathbf{x_j}) = \phi(\mathbf{x_i})\phi(\mathbf{x_j})$ is the kernel function.
The choice of the kernel function is in some way an expression of some prior information about the problem that allows making the classification even if in the original space was not possible. There are several possible kernel function but the most common are the *polynomial* and the *radial basis function*.

$$polynomial\ kernel\ of\ degree\ d : K(\mathbf{x_i}, \mathbf{x_j}) = (1 + \gamma \langle \mathbf{x_i}, \mathbf{x_j} \rangle)^d$$

$$radial\ basis\ function\ kernel\ (rbf) : K(\mathbf{x_i}, \mathbf{x_j}) = e^{-\gamma \|\mathbf{x_i} - \mathbf{x_j}\|^2}$$

To choose the best one for a problem, the grid search with CV is always a good way to test different combinations of kernel functions and also hyper-parameters. The hyper-parameter C is the one that regulates the trade-off between the correct classification of training examples and maximization of the decision function's margin, as seen before, and it is common to all the possible implementations. Higher values of C tend to lead a correct classification of all the training samples accepting smaller margin, a lower C penalize the model complexity giving a simpler decision function and accepting larger margin.
The kernel-specif hyper-parameters are d and γ in the polynomial case and only γ in the rbf case. The d hyper-parameter is the degree of the polynomial kernel and controls the dimension of the correspondent feature space in which the data would be mapped. Higher values of d allow the discrimination of the classes in case of very non-linearity, giving enough flexibility to the decision function to separate the classes and also keeping a sizeable margin.
The hyper-parameter γ is a measure of the influence of a single training instance, it can be seen as the inverse of the *radius* of influence of samples selected by the model as support vectors. Lower values of γ mean that the support vectors can influence very far data until the region of influence of any selected support vector would include the whole training set. In this way, the model cannot capture the complexity or "shape" of the data because it would have a smooth decision boundary, make it similar to a linear model. Instead, with higher and higher values of γ, the locality of the support vector influence increases until the radius of the area of influence of the support vectors only includes the support vector itself. This would lead to greater curvature of the decision boundary, that can make the model overfits the training data. If the value of γ is not too extreme, the choice of both the hyper-parameters is important because one can compensate for the effect of the other. For example, smooth models given by lower γ values can be made more complex by increasing the importance of classifying each point correctly, i.e. with larger C values.

The SVM algorithm has been chosen for its versatility given by the possibility of using different kernel functions and of tuning many hyper-parameters in order to find the best one for the type of data.

3.5 Gradient Boosted Decision Trees

The Gradient Boosted Decision Trees algorithm is an ensemble model that consists of a set of classification and regression trees (CARTs). CARTs are decision trees that classify the instances into different leaves (the classes) giving also a score on the corresponding leaves. In this way a richer interpretation of the classification is possible, providing a unified approach to the optimization. The power of the ensemble model, which sums the prediction scores of multiple trees together to get the final score, is clearly higher respect the use of a single decision tree. The final prediction of a sample i is:

$$\hat{y}_i = \sum_{k=1}^{K} \mathbf{f_k}(\mathbf{x_i}), \mathbf{f_k} \in \boldsymbol{F}$$

where K is the number of decision trees, $\mathbf{f}$ is a function in the functional space $\boldsymbol{F}$, that is the set of all the possible decision CARTs. The objective function to minimize to identify the model is composed of two trade-off terms, one required to fit well the training data and one for the regularization that penalizes model complexity to avoid the overfitting:

$$obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{i=1}^{t} R(\mathbf{f_i})$$

The functions $\mathbf{f_i}$, each containing the structure of the tree and the leaves scores, have to be estimated; the problem is that it is almost impossible learning all the trees at once. Instead, an additive strategy is adopted, fixing what has been already learnt and adding another tree at time. At each step, the tree to add in the model is the one that optimizes the objective function. Therefore, the updating of the model uses a gradient descent approach and hence the name, *gradient boosting*. In this way the identification problem becomes feasible and, at each step t, it is possible to write the

prediction value y_i in function of the previous ones:

$$\begin{aligned}
\hat{y_i}^{(0)} &= 0 \\
\hat{y_i}^{(1)} &= \mathbf{f_1}(\mathbf{x_i}) = \hat{y_i}^{(0)} + \mathbf{f_1}(\mathbf{x_i}) \\
\hat{y_i}^{(2)} &= \mathbf{f_1}(\mathbf{x_i}) + \mathbf{f_2}(\mathbf{x_i}) = \hat{y_i}^{(1)} + \mathbf{f_2}(\mathbf{x_i}) \\
&... \\
\hat{y_i}^{(t)} &= \sum_{k=1}^{t} \mathbf{f_k}(\mathbf{x_i}) = \hat{y_i}^{(t-1)} + \mathbf{f_t}(\mathbf{x_i})
\end{aligned}$$

Therefore the objective function at step t becomes:

$$obj^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y_i}^{(t)}) + \sum_{i=1}^{t} R(\mathbf{f_i}) = \sum_{i=1}^{n} l(y_i, \hat{y_i}^{(t-1)} + \mathbf{f_t}(\mathbf{x_i})) + R(\mathbf{f_t}) + constant$$

Then the *Taylor expansion* of the loss function up to the second order is taken to get the final form:

$$obj^{(t)} = \sum_{i=1}^{n} [\mathbf{g_i}\mathbf{f_t}(\mathbf{x_i}) + \frac{1}{2}\mathbf{h_i}\mathbf{f_t}^2(\mathbf{x_i})] + R(\mathbf{f_t})$$

where g_i is the partial derivative with respect of $y_i^{(t-1)}$ of the loss computed with $y_i^{(t-1)}$ and h_i is the second order partial derivative with respect of $y_i^{(t-1)}$ of the loss computed with $y_i^{(t-1)}$. In this form the objective depends only on h_i and g_i to support any loss function, even personalized. The one chosen for this kind of classification problem is the *logistic regression* for binary classification that gives in output probability values.

The tree can be formulated in this form:

$$\mathbf{f_t}(\mathbf{x}) = \mathbf{w}_{\mathbf{q}(\mathbf{x})},\ \mathbf{w} \in \mathbb{R}^T,\ \mathbf{q} : \mathbb{R}^d \to \{1, 2, ..., T\}$$

with $\mathbf{w}$ a vector of scores on leaves, $\mathbf{q}$ is the function that assigns data points to the corresponding leaves and T is the number of leaves. With this definition it is possible to set the regularization function R in term of the complexity of the model, including the number of leaves T and the norm of the vector $\mathbf{w}$ (l_2 norm in this case) in the formulation:

$$R(\mathbf{f}) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \mathbf{w_j}^2$$

Finally, after another re-formulation and using the quadratic norm in the regularization function, the objective function at step t becomes:

$$obj^{(t)} = \sum_{j=1}^{T} \left[G_j \mathbf{w_j} + \frac{1}{2}(H_j + \lambda)\mathbf{w_j}^2 \right] + \gamma T$$

with $G_j = \sum_{i \in I_j} g_i$, $H_j = \sum_{i \in I_j} h_i$ and $I_j = \{i | \mathbf{q}(\mathbf{x_i}) = j\}$ that is the set of indices of data points assigned to the j-th leaf. In this way the objective is in a quadratic form, the weight vectors $\mathbf{w_j}$ are independent and so, given the tree-structure $\mathbf{q}(\mathbf{x})$, the solution $\mathbf{w_j}$ is:

$$\mathbf{w_j}^* = -\frac{G_j}{H_j + \lambda}$$

that gives the best reduction in the objective function: $obj^* = -\frac{1}{2}\sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \gamma T$. With this formula it is possible to evaluate the goodness of a tree, so ideally all the possible trees would be evaluated and then the best one in term of objective function would be chosen, but in practice this is not feasible. The procedure implemented is to consider one level of tree at time, split each leaf in two, evaluate the score gain and decide if it is sufficient to keep the division and so to add the branch (this is the so-called *pruning* technique). This approach works well most of the time and allows this model to be scalable, accurate, flexible and implementable in parallel.
The number of estimators in the model is a parameter that is tuned by a CV-xgb function, given the other hyper-parameters. The hyper-parameters to tune chosen are many:

- *learning rate*: at each boost step shrinks the feature weights to make the boosting process more conservative;
- *max-depth*: maximum depth of a tree, increasing the value will make the model more complex and more likely to overfit;
- *min-child-weight*: minimum sum of instance weight needed in a child, the higher the more conservative is the model;
- *gamma*: minimum loss reduction required to make a further partition on a leaf node of the tree, make the algorithm conservative;

- *subsample*: subsample ratio of the training instances randomly kept at each boosting iteration, values lower than 1 can prevent overfitting;
- *colsample-by-tree*: subsample ratio of columns when constructing each tree;
- *reg-lambda*: L2-regularization term on weights, increasing this value will make model more conservative.

The Gradient Boosted Decision Trees model has been chosen for the capacity of dealing with many types of data, even if they does not present regularity, for its flexibility in the tuning of many different parameters, for the parallel processing that speeds up the computation.

The chosen algorithms have many implementative properties to define based on the task to solve. Furthermore, different approaches could have been tried to try to solve the problem of the prediction of the nocturnal hypoglycemic events. In chapter 4 all the procedures for model-parameters estimation are reported, according to the approaches chosen.

CHAPTER 4

Procedures for model-parameters estimation

In this chapter, the procedures used for the model-parameters estimation are described, pointing out the techniques and the strategies adopted for the task of the prediction of the nocturnal-hypoglycemic events. In particular, two different approaches have been tried, with different procedures implemented. The first approach consists in the *population analysis*, in which all the patients are treated together as a single patient and the models are identified with all the data to find the *population models*. In the second approach instead the patients are treated separately for the *individual analysis*. So, for each of them ,different models are identified using only the data of the specific patient. In the end, the metrics useful to determine the quality of the methods implemented are defined, according to the type of problem.

4.1 General implementative choices

The implementative choices reported in this section are common to both the population and individual analysis.

4.1.1 Data Standardization

The data matrix X, containing the values of the 34 features for each sample, is normalized to have each feature with zero mean and unitary standard deviation. The

features are normalized independently from the others; each sample is standardized removing in all the features values the corresponding mean and dividing by the corresponding standard deviation. If x_{ij} is the value of the feature j in the sample i, the normalization would be:

$$z_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}$$

This pre-processing is necessary in many machine-learning algorithms, as in the L2-L1 Regularized Logistic Regression, where the assumption is that all features are centred around zero and have variance in the same order. If a feature has a variance that is orders of magnitude larger than others, it might dominate the objective function and make the estimator unable to learn from other features correctly as expected.

4.1.2 Hyper-parameters tuning

In the models that need the tuning of some hyper-parameters, the *Grid Search-Cross Validation* has been used when it was possible. For the XGBoost implementation of the Gradient Boosted Decision Trees model this approach was not possible because of computational reason, so a manual tuning of the hyper-parameters was used.
In the finding of the optimal hyper-parameters, the metric to evaluate the performance chosen is the F1-score because of its informative power in this kind of task.

4.1.3 Class-weights

In all the models implemented, the weights on the classes have been added in the cost function to optimize. It means that each sample has a different weight in the cost function, based on its label class, to make more relevant the errors on the hypoglycemic samples, that is the minority class and that is also the most important to predict. The hypoglycemic samples have a weight equal to the proportion of the not hypoglycemic class and vice versa. This is a possibility to try to handle the unbalancing of the two classes.

4.2 Implementative choices specific of the population analysis

In this analysis, all the patients of a dataset are used together to find the population models, for the ReplaceBG and the Real dataset data separately. The samples are randomly dived in a *training set* and in a *test set* with a division 50-50%, keeping the original proportion of the classes in each of them. The training set is used for the identification of the models and the test set for the evaluation of their performances. The pairwise *pearson correlation* of the features and the labels vector are plotted using the *corr* function in Python, to see if there exist some strong correlation. In the graph, the yellow and dark blue colors represent strong correlation between two characteristics, respectively positive and negative correlation. All the color tones between these two indicate different degree of weak correlation while the color dark green indicates no correlation at all.
Unfortunately, observing the graphs (Figure 4.1), it does not seem that the labels vector is highly correlated with some features, only some of them present a little evident negative correlation.

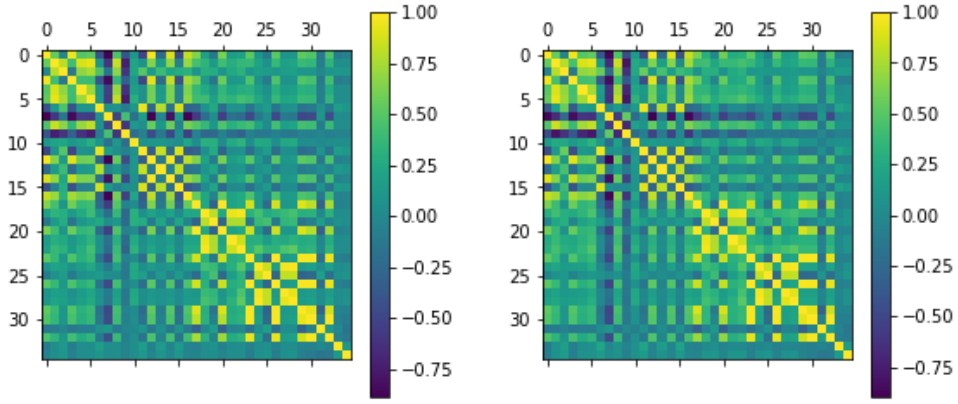

Figure 4.1: Correlation Matrices, ReplaceBG dataset on the left and Real dataset on the right

4.2.1 Regularized Logistic Regression

For the L2 and L1-regularized Logistic Regression, a grid of possible C regularization parameters is tried in the Grid Search-Cross Validation, using the training samples. In the first attempts, a wider but less dense grid of values is used, to find in which zone of values searching the hyper-parameters and then use a tighter but denser parameters-grid. In this way it is possible to find the optimal hyper-parameter C in term of f1-score, using a 10-folds cross-validation to evaluate the performance of each parameter tried.
The final values of the regularization parameter C are in Table 4.1.

C-regularization parameter	L2-reg	L1-reg
ReplaceBG dataset	1.63	2.60
Real dataset	1.33	1.50

Table 4.1: Values of the C- regularization hyper-parameter in the different implementation of the *regularized logistic regression* for the two datasets

The different implementations of the Logistic Regression give different parameter estimation $\mathbf{w}$. In Figure 4.2 it is possible to see the coefficient of each feature in the three implementations. It is possible to notice the lowering of the coefficients amplitudes in the L2-regularization and the expected shrinking of the coefficients in the L1-regularization.

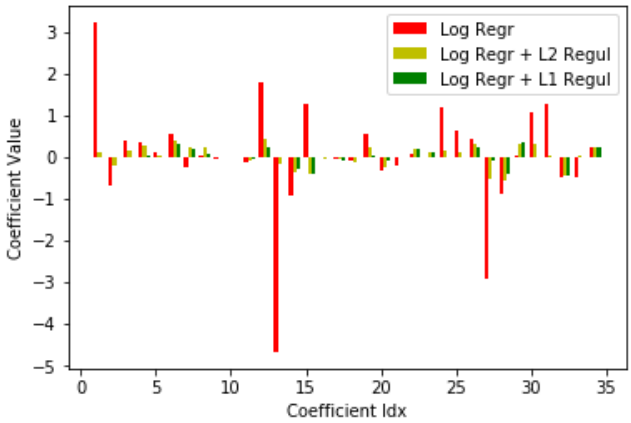

Figure 4.2: Coefficient amplitude of the estimated parameters in the different implementation of the LR algorithm, ReplaceBG and Real dataset

4.2.2 Choice of Logistic models using the Receiver Operating Characteristic (ROC) Curves

For the Logistic Regression models, it is possible to plot the ROC curves in function of the probability-threshold applied to make the classification in hypoglycemic and not hypoglycemic samples. In the standard ROC curve the two metrics computed at the variation of the threshold are the *False Positive Rate (FPR)* and the *True Positive Rate (TPR)*.

$$FPR = \frac{FP}{FP + TN} \qquad TPR = \frac{TP}{TP + FN} = recall$$

The aim of plotting this curve is finding the best probability-threshold, that is the one that allows obtaining good results in both the metrics. So the point to look for is in the left-upper part of the curve, further from the bisector (that is the "random" classifier).
For this kind of task, with an imbalanced data-set and in which the false-positive error is more tolerable than the false-negative one, it is useful also to consider the curve of the *precision* and the *recall* obtained at the variation of the probability threshold. With a lower probability threshold, the recall will increase and the precision will decrease and vice versa with a higher one. The goal of the analysis of this curve is to

find an optimal threshold that gives a compromise in the trade-off of the two metrics. From the analysis of the first curve, it is possible to set the optimal probability-threshold via visual inspection, the second curve does not allow to understand the optimal threshold, due to the shape of it. Unfortunately, the setting of the threshold via visual inspection does not improve the result metrics and so, to keep the model not human-dependent and more generalizable, the default threshold of 0.5 is kept for all the analysis.

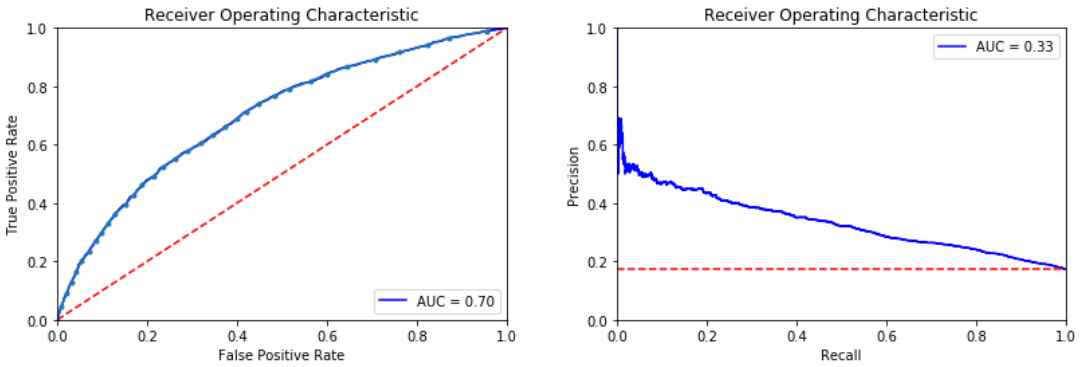

Figure 4.3: ROC and PrecisionVSRecall curves of the Logistic Regressor, ReplaceBG dataset

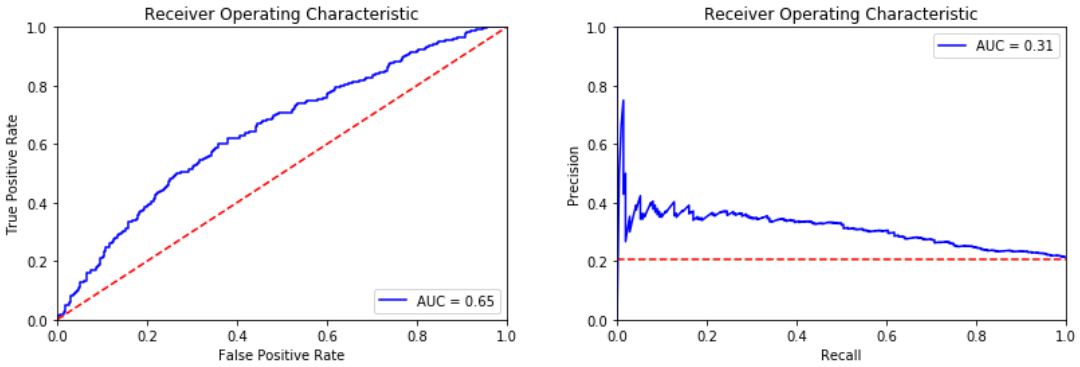

Figure 4.4: ROC and PrecisionVSRecall curves of the Logistic Regressor, Real dataset

4.2.3 Support Vector Machine

In the implementation of the SVM algorithm, the first choice to do is the kernel function. The Grid Search-Cross Validation approach has been implemented with a 10-folds CV on the training set data, in order to find the optimal choice between the *linear* kernel, the *3-rd degree polynomial* kernel and the *radial basis function* kernel. The optimal one is the *radial basis function* kernel, both for the RepalceBG and for the Real datasets. Then the same procedure applied for the Regularized-Logistic Regression is used to find the optimal values of the hyper-parameter C and γ. The final values found are in Table 4.2.

	C	γ
ReplaceBG dataset	990.10	0.001
Real dataset	21.70	0.001

Table 4.2: Values of the hyper-parameters C and γ for the two datasets

4.2.4 Gradient Boosted Decision Trees

In this model, the tuning of the hyper-parameters is a fundamental aspect to obtain good results on the specific problem. To find the optimal set of hyper-parameters a step-by-step procedure is adopted, starting with a fixed higher learning-rate and finding the optimal number of trees with the $xgb - CV$ function. After the tuning continues with the tree-specific ones and then with the regularization parameters. In the end, lower values of learning-rate are tried, increasing the number of trees.
The first two tree-specific hyper-parameters are the *max-depth* and the *min-child-weight* then the parameter *gamma* and finally the *subsample* and the *colsample-bytree*. The regularization tried is the L2 and so the hyper-parameter investigated is the *reg-lambda*. The final hyper-parameters are in Table 4.3.

	ReplaceBG dataset	Real dataset
learning-rate	0.1	0.1
number of trees	50	20
max-depth	2	1
min-child-weight	1	1
gamma	0.1	0
subsample	0.5	0.9
colsample-bytree	0.3	0.7
reg-lambda	1	1

Table 4.3: Values of the hyper-parameters for the two datasets

4.3 Implementative choices specific of the individual analysis

In the analysis of the patient-specific models, instead of considering all the samples together to find the population models, the data of each patient are used in a separate way. The total number of different datasets are 249 but, to train the models, a minimum-threshold of 10 hypoglycemic samples in the patient-dataset is fixed, to allow the algorithms having enough examples of the minority class. For this reason, only 181 patients are kept. The dataset of each patient is divided in a *training set* and in the *test set* of samples with proportions 2/3 and 1/3.

4.3.1 Analysis of all the patient-specific models

In the first part, the models are trained using the same hyper-parameter grids for the grid search-CV in the Regularized Logistic Regression, in the SVM and in the Gradient Boosted Decision Trees. The results are analyzed in term of mean and standard deviation of the four metrics, observing also the boxplots containing the results of each patient model, considering each metric and each algorithm implemented.

4.3.2 Analysis of specific patient-specific models

Considering the analysis of one patient at time, it is possible to optimize some aspects of the algorithms:

- make a statistical selection of the features, based on their predictive power;
- set the optimal *probability-threshold* by the observation of the ROC curves and of the PrecisionVSRecall curves for the Logistic Regressors;
- search the optimal hyper-parameters for all the algorithms with a specif grid-search.

Statistical Features Selection

To assess if a feature is useful in the prediction of the nocturnal-hypoglycemic events, a statistical analysis is performed. This analysis is based on the *likelihood metric* of a logistic-model to compute the *deviance* of that model, that is a measure of lack of data-fit power. To find the contribution of a predictor on the full model, a way is to subtract the deviance of the model trained with only that predictor to the so-called *null-model*. The null-model is the model trained without any features but only with the intercept.

$$D_{null} - D_{test} = -2ln\frac{null\ model\ likelihood}{test\ model\ likelihood}$$

with

$$likelihood = Prob(\mathbf{y}|\mathbf{X}, \mathbf{w})$$

It can be shown that the deviance follows a $\chi^2(k)$ distribution with k degree of freedom equal to the difference of the number of estimated parameters of the model to test and the null one (that is the number of features used to train the tested model). If the deviance of the test model is significantly smaller than the one of the null model it is possible to conclude that the tested predictors improve the model fit.
Therefore to test some null hypothesis $H_0 : w_k = 0$ that a feature k (or a set of features) is not predictive, the difference of the deviance is computed and a level of significance α is fixed (in this case $\alpha = 0.05$). If the null-hypothesis is true the deviance-difference would be smaller than the value of a $\chi^2(p_{test} - p_{null})$ at the level of significance α

fixed. If this is not verified, it is possible to reject the null hypothesis and conclude that the predictor tested are improving the model-fit performance.
Hence, in the patient-specif analysis, it is possible to apply this feature selection because the samples are not so many as in the population model, where all the features result relevant for this reason. Therefore, before estimate the model parameters, only the predictors that are shown to be relevant are kept for the successive steps.

Analysis of the ROC and PrecisionVSRecall curves

Whit this procedure it is possible to set an optimal threshold for the classification, observing the plot of the metrics at the variation of the probability-threshold in the Logistic Regression algorithms. The curve chosen to make the choice is the *PrecisionVSRecall curve*, that allow evaluating the trade-off between the precision and the recall, that is the most relevant for this kind of task.

Hyper-parameters tuning

Controlling the tuning of the hyper-parameters for one patient at time, the grid to test is specific for the dataset analyzed and so it is possible to find the optimal set of hyper-parameters.

4.4 Metrics used for the assessment of the results

To evaluate the goodness of a model in solving the task of the hypoglycemic-events prediction, in addition to the computation of the confusion matrices, four metrics are chosen: *accuracy, precision, recall, F1-score*.
The metrics are defined finding the total number of these four types of events that can happen considering the classification of one sample:

- a *true positive event (TP)* is defined as the case in which the sample has the positive label (is an hypoglycemic sample in this task) and it is correctly classified as positive by the model;
- a *true negative event (TN)* is defined as the case in which the sample has the negative label (is not an hypoglycemic sample) and it is correctly classified as negative by the model;
- a *false positive event (FP)* is defined as the case in which the sample has the negative label and it is not correctly classified as positive by the model;
- a *false negative event (FN)* is defined as the case in which the sample has the positive label and it is not correctly classified as negative by the model.

Therefore, the metrics are defined as:

- *Accuracy*: is the total number of correctly-classified samples on the total number of samples;
$$acc = \frac{TP + TN}{TP + TN + FP + FN}$$
- *Precision*: is the total number of correctly-classified positive samples on the total number of samples classified as positive;
$$prec = \frac{TP}{TP + FP}$$
- *Recall*: is the total number of correctly-classified positive samples on the total number of positive samples;
$$rec = \frac{TP}{TP + FN}$$

- *F1-score*: is the harmonic mean of the precision and of the recall metrics;

$$F1 = 2 * \frac{precision * recall}{precision + recall} = \frac{2TP}{2TP + FN + FP}$$

In this kind of prediction task, with an unbalanced dataset and with the minority class that is dangerous not to predict, the most useful metric is the *F1-score*. In fact, in the accuracy metric, the weight of the true negative events could hide the bad performance in the prediction of the positive samples, that are in a smaller number and more important to predict. Instead, in the F1-score, the number of the true negative samples are not present, making its value more informative of the goodness of the model performance.
The precision and the recall metrics are possible in trade-off, because the fact that if the model is not able to predict the positive samples and so it has bad performance in term of recall (i.e. there is a large number of FN), it is possible that the model would not predict anything and so the precision would be near to the maximum value one (i.e. there would be a low number of FP) and vice-versa. However, the recall is more important in this kind of task, because of the fact that the FN events, that are the not predictions of the hypoglycemic events, are more dangerous than the FP ones, that are the predictions of false hypoglycemic events.
Finally, the confusion matrices are the count of the total number of the four different events that happen in the classification of the samples.

	Classified Positive	**Classified Negative**
Positive	TP	FN
Negative	FP	TN

Table 4.4: Confusion Matrix

Defined all these implementation aspects of the methodologies for the different approaches tested, the algorithms identified are tested. The found results are reported in chapter 5.

CHAPTER 5

Results

In this chapter, the results of the identified models for the population and for the individual analysis are presented, trying to understand the capabilities and the limits of the implemented methodologies.

5.1 Results of the population models

After the identification of the population models, the results on the test set of the ensemble dataset are found. The algorithms were tested both in terms of metrics described previously and in terms of confusion matrices. Then an analysis on some borderline-samples, that result in classification-errors, is carried out in order to better understand the capability of the algorithms, considering the type of task.

5.1.1 Result Metrics

The five different algorithms implemented give similar results. In term of F1-score, that is the most informative one in this task, the Logistic Regressors and the XGB-Decision Trees algorithm give slightly better results compared with the SVM algorithm in the ReplaceBG dataset; in the Real dataset instead, the XGB-Decision Trees one is the best.

It is important to notice that the different implementations of the Logistic Regression give the same results, suggesting that the regularization is not improving the performances. This can be due to the fact that the Logistic Regressor is not learning the problem and that there is not any kind of over-fitting in the training part for what regards the L2 regularization; for the L1, probably because the algorithm is not learning how the data works, there are not benefits from shrinking the features and the results are similar to the ones without the regularization.

Algorithm	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.66	0.28	0.61	0.39
L2-Reg Logistic Regression	0.66	0.28	0.61	0.38
L1-Reg Logistic Regression	0.66	0.28	0.61	0.38
SVM	0.67	0.29	0.62	0.24
XGB-Decision Trees	0.72	0.32	0.54	0.40

Table 5.1: Results on test data of the five identified population models, ReplaceBG dataset

Algorithm	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.66	0.31	0.52	0.39
L2-Reg Logistic Regression	0.67	0.32	0.54	0.40
L1-Reg Logistic Regression	0.67	0.33	0.56	0.41
SVM	0.67	0.32	0.56	0.28
XGB-Decision Trees	0.64	0.32	0.65	0.43

Table 5.2: Results on test data of the five identified population models, Real dataset

Furthermore, the populations models are applied on each patient test set, to test their performance separately. The results presented are the ones of the logistic regression as examples; the ones of the other are similar and are shown in the *Appendix A* to not burdening the presentation.

In the boxplot graph, the red line represents the median of all the result metrics obtained in the application of the population models on each patient datasets while

the limits of the box are the first and third quartiles of the distribution of the result metrics. The whiskers extend from the box to show the range of the data (the values that limit the extreme values of samples on both side of the distribution) and flier points are those past the end of the whiskers.

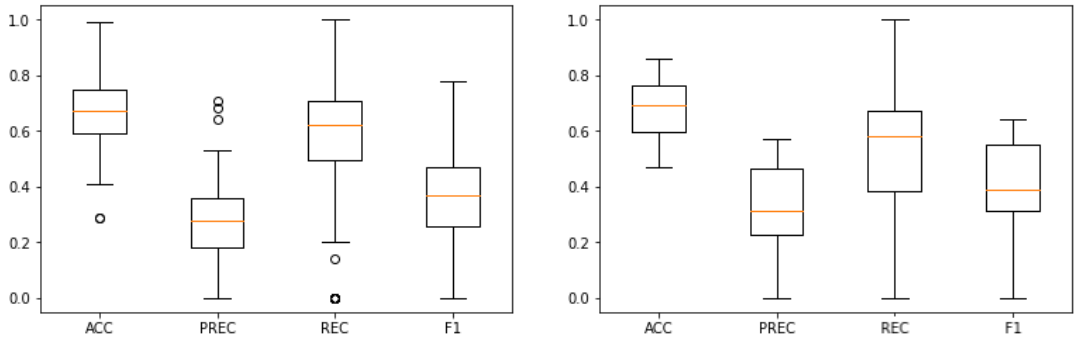

Figure 5.1: Boxplots of the result metrics of the LR population models on each patient datasets, ReplaceBG and Real datasets

To confirm the fact that the training is not working, also the results on the training set of the ensemble dataset are presented, that are the samples on which the models are identified. These results, particularly for the F1-score, are not much better than the ones on the test set, suggesting that the problem is not the over-fitting of the train data. Probably the problem is that the algorithms are not capable of solving the task; another possibility is that there are not enough samples to understand the problem, even if the numerosity is not very bad. Try new methodologies is the way to understand if the task can be solved and to confirm the inadequacy of the algorithms tried so far.

Algorithm	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.66	0.29	0.64	0.40
L2-Reg Logistic Regression	0.66	0.29	0.64	0.40
L1-Reg Logistic Regression	0.66	0.29	0.64	0.40
SVM	0.68	0.30	0.66	0.42
XGB-Decision Trees	0.73	0.33	0.57	0.42

Table 5.3: Results on train data of the five identified population models, ReplaceBG dataset

Algorithm	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.69	0.36	0.66	0.47
L2-Reg Logistic Regression	0.70	0.37	0.65	0.47
L1-Reg Logistic Regression	0.69	0.37	0.65	0.47
SVM	0.70	0.38	0.67	0.49
XGB-Decision Trees	0.65	0.34	0.75	0.47

Table 5.4: Results on train data of the five identified population models, Real dataset

5.1.2 Confusion Matrices

As examples of confusion matrices obtained with the Logistic Regressors, the ones obtained without regularization are shown together with the ones obtained with the SVM algorithm. The first rows contain the *True Positive* and the*False negative* samples, the second rows contain the *False Positive* and *True Negative* samples. As expected from the result metrics, there are a lot of errors in the classification, indicating that the models have not learned the problem.

	Classified Positive	Classified Negative
Positive	1348	849
Negative	3454	7042

Table 5.5: Confusion Matrix obtained using the Log Reg algorithm, ReplaceBG dataset

	Classified Positive	Classified Negative
Positive	1347	850
Negative	3411	7085

Table 5.6: Confusion Matrix obtained using the SVM algorithm, ReplaceBG dataset

	Classified Positive	Classified Negative
Positive	111	101
Negative	245	565

Table 5.7: Confusion Matrix obtained using the Log Reg algorithm, Real dataset

	Classified Positive	Classified Negative
Positive	118	94
Negative	247	563

Table 5.8: Confusion Matrix obtained using the SVM algorithm, Real dataset

5.1.3 Identification of possible borderline cases

Regarding this classification problem, it is clear that the gravity of an error can vary from a situation to another. For example, if the algorithm does not classify as hypoglycemic a night in which the patient had a very long period under the hypoglycemic-threshold and maybe also with very low values of glycemia, the error would be important and clearly dangerous. Instead, if the algorithm classifies a sample as hypoglycemic and the patient did not have an event as defined but maybe he had low glycemia for a certain time, the error would not be so relevant and the alarm to the patient would be however useful.
To consider this fact, an analysis of the so-called *"borderline-samples"* of the test set is made to be more tolerant on errors that are not so dangerous. The first type of borderline-samples is the ones related to a *false positive error*, where the algorithm has classified a sample as hypoglycemic and it was not. The condition to consider a false positive error as a borderline is the fact that the nocturnal CGM profile contains at least three consecutive points under a more conservative threshold, that is 80 mg/dl instead of 70 mg/dl. In this way, an error of this type is not considered and the sample is removed from the computation of the results, because of its ambiguity.
The second type of borderline-samples are the ones related to a *false negative error*, where the algorithm has classified a sample as not glycemic and it was, that could be very dangerous for the patient. The condition to consider a false negative error as a borderline is that the nocturnal profile of the sample contains only the three consecutive points under the threshold that make it a glycemic sample, and so that there is not any other CGM value under 70 mg/dl. If this happens the error is not considered and the sample is removed from the computation of the results.
In the end, the final results are the ones that do not consider both these borderline-samples and they are reported above. There are some improvements in all the result metrics, suggesting that maybe the problem of the prediction of the nocturnal hypoglycemia is much more complex than a simple binary-classification task. In order to be more useful, the models have to include also some kind of information on the severity of the hypoglycemia predicted; maybe that can be realized using other types of prediction and other parameters to predict, for example, a GV index like the LBGI.

In this way the patient could have a more realistic view of what it could happen in the night and, basing on his experience and on his way of menage these problems, he could decide if take some provision, which kind of provision and in which measure.

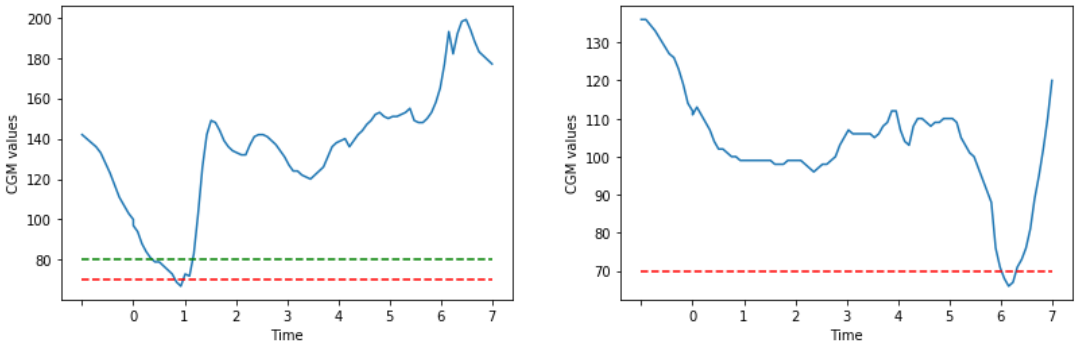

Figure 5.2: CGM night curves of a FP and FN borderline samples

Algorithm	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.70	0.33	0.63	0.43
L2-Reg Logistic Regression	0.70	0.32	0.63	0.43
L1-Reg Logistic Regression	0.70	0.33	0.63	0.43
SVM	0.71	0.33	0.64	0.44
XGB-Decision Trees	0.76	0.37	0.56	0.45

Table 5.9: Results of the five population models with the borderline-samples correction, ReplaceBG dataset

Algorithm	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.70	0.36	0.53	0.43
L2-Reg Logistic Regression	0.70	0.37	0.55	0.44
L1-Reg Logistic Regression	0.70	0.37	0.57	0.45
SVM	0.70	0.37	0.57	0.45
XGB-Decision Trees	0.68	0.35	0.60	0.45

Table 5.10: Results of the five population models with the borderline-samples correction, Real dataset

5.2 Results of the patient-specific models

In the case of personalized-models for each patient, the analysis of the results is composed of a general part and of a patient-specific one.

5.2.1 Result of all the patient-specific models

The mean metrics of the patient-specific models and the boxplots of the result metrics of each patients for all the algorithms are presented above. The boxplot graph are analogous to the one presented above for the population model. The mean results are similar to the metrics resulted from the population models. The model that in mean has better result in term of F1-score is the L1-Regularized Logistic Regression, that has also the best Recall result. The XGB-Decision Tree has not a good result since it needs a personalized tuning of the hyper-parameters that cannot be achieved via

CV.

ALGORITHM	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.66 ± 0.11	0.27 ± 0.18	0.35 ± 0.21	0.30 ± 0.17
L2-Reg Logistic Regression	0.66 ± 0.09	0.27 ± 0.15	0.42 ± 0.20	0.32 ± 0.15
L1-Reg Logistic Regression	0.59 ± 0.10	0.27 ± 0.14	0.50 ± 0.22	0.33 ± 0.15
SVM	0.63 ± 0.13	0.33 ± 0.16	0.42 ± 0.26	0.31 ± 0.17
XGB-Decision Trees	0.67 ± 0.11	0.27 ± 0.16	0.35 ± 0.22	0.29 ± 0.17

Table 5.11: Results of the patient-specific models on all the patients, mean and SD

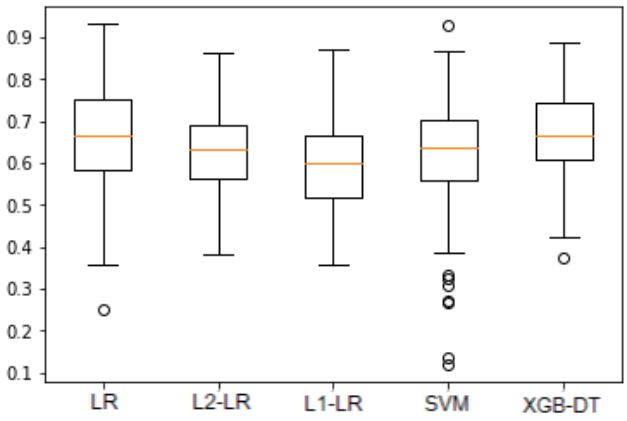

Figure 5.3: Boxplot of the *accuracy* of all the patient-specific models for the five algorithms

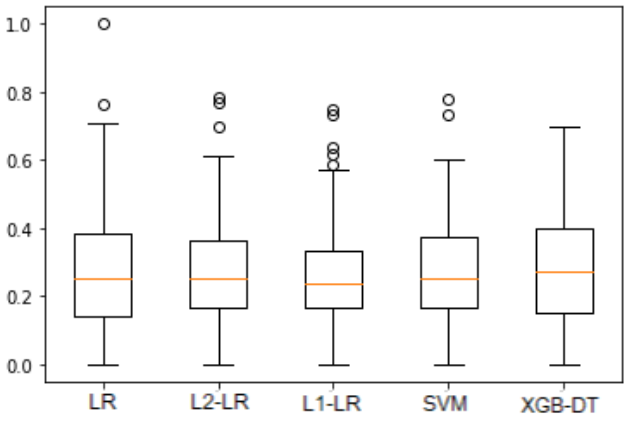

Figure 5.4: Boxplot of the *precision* of all the patient-specific models for the five algorithms

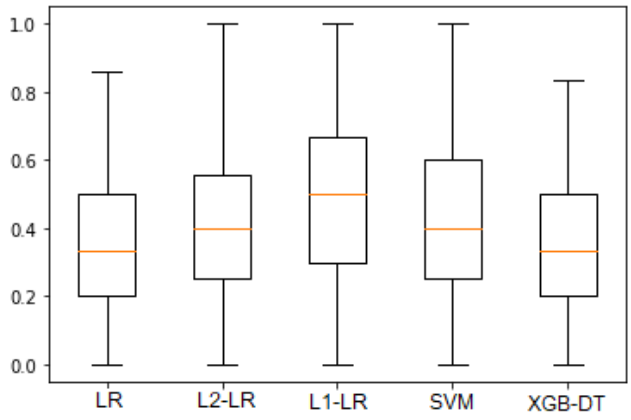

Figure 5.5: Boxplot of the *recall* of all the patient-specific models for the five algorithms

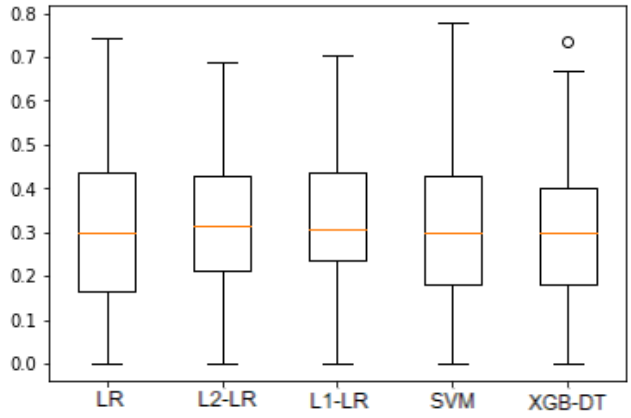

Figure 5.6: Boxplot of the *F1-score* of all the patient-specific models for the five algorithms

5.2.2 Result of the patient-specific analysis

Considering the analysis of one patient at time, it is possible to optimize some aspects of the algorithms. Two examples of these procedures are reported, one of a patient of the ReplaceBG dataset and one of a patient of the Real dataset.

Patient number 19, Real dataset

For this patient, the optimizations improve in some aspect the performance of the models. The total number of samples is 90, 67 in the training and 23 in the test for a 75-25% division. The percentage of the hypoglycemic class is 24% in the training set (16 samples) and 22% in the test set (5 samples).
The correlation matrix, analogous to the one presented in the population analysis, does not show any very correlated feature with the labels vector, even if there is some feature with correlation different from zero (Figure 5.7)

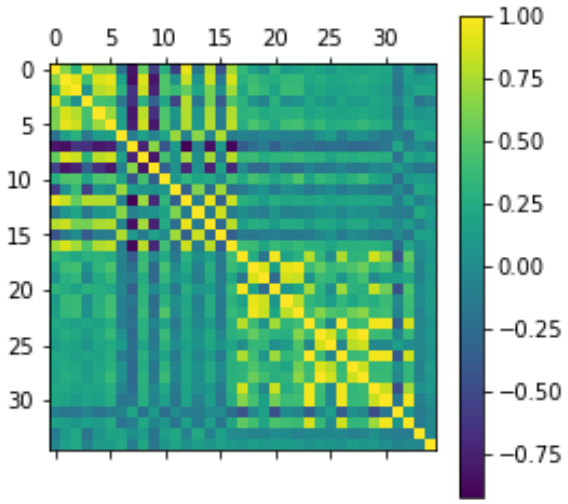

Figure 5.7: Correlation-Matrix of the features plus the labels vector

The statistical feature selection gives as significant predictive the features: *Coefficient of Variation*, *fraction of BG samples below target*, $Mage_+$, *Low Blood Glucose Index*, *Hypo Index* .

As example of ROC curves analysis on training data, the one of the Logistic Regression without regularization is shown (Figure 5.8). The choice of the probability threshold is made on the *PrecisonVSRecall* curve and the optimal point gives a threshold equal to 0.38, lower than the standard 0.5. In fact, the choice of the optimal point was made to rise the recall metric for the motivation already discussed, so the probability-threshold has been lowered. For the choice of the hyper-parameters in the regularized Logistic Regressors, the selection is made more accurately trying denser grids of parameters, starting from wider grids and continuing with more and more restricted ones. The original and the final parameters are shown in the result tables.

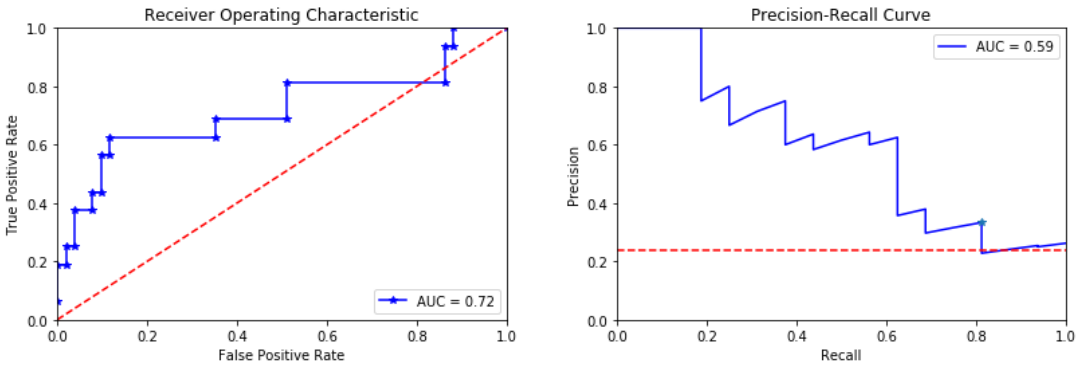

Figure 5.8: ROC and PrecisionVSRecall curve of the Logistic Regression, patient 19 of the Real dataset

For the Support Vector Machine model, the hyper-parameter selection is made more accurately as described before for the Logistic Regressors and the parameters selected are shown in the result tables.

Also for the XGB-Decision Trees, the tuning of the hyper-parameters follow a step-by-step selection, permitting to obtain a patient-specific choice of them and so better result. The final hyper-parameters are:

- # estimators=20 (instead of 30)
- max-depth=3 (keeping the same)
- min-child-weight=1 (keeping the same)
- gamma=0 (keeping the same)

- reg-alpha=0 (keeping the same)
- subsample=0.9 (instead of 0.6)
- colsample-bytree=0.9 (instead of 0.6)

The results are in Table 5.12, to compare with the one obtained in the previous analysis without the procedures in Table 5.13. The final results are better than the original for all the models and for all the metrics, except for the accuracy that is worse in the L2 and L1-Regularized Logistic Regression.
The largest improvement in the F1-score is in the L2-Regularized Logistic regression, from a value of 0.24 to a value of 0.38, that is anyway a bad result. The L2-Regularized Logistic regression also obtains a value of 1.00 in the recall from a value of 0.40 but this leads to a worsening in the accuracy from 0.43 to 0.30. In the complex, the best improvements are the one of the SVM because it obtains a larger values of all the metrics, without worsening the accuracy.

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Logistic Regression	0.39	0.09	0.20	0.13
L2-Reg Logistic Regression (C=0.30)	0.43	0.17	0.40	0.24
L1-Reg Logistic Regression (C=0.50)	0.48	0.18	0.40	0.25
SVM (C=10, $\gamma = 0.01$)	0.61	0.17	0.20	0.18
XGB-Decision Trees	0.52	0.12	0.20	0.15

Table 5.12: Original results on test data, patient 19 of the Real dataset

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Logistic Regression	0.43	0.21	0.60	0.32
L2-Reg Logistic Regression (C=0.045)	0.30	0.24	1.00	0.38
L1-Reg Logistic Regression (C=1.72)	0.39	0.20	0.60	0.30
SVM (C=407.4, $\gamma = 0.21$)	0.67	0.32	0.56	0.28
XGB-Decision Trees	0.52	0.20	0.40	0.27

Table 5.13: Results on test data with the model modifications, patient 19 of the Real dataset

Patient number 31, ReplaceBG dataset

The patient 31 of the ReplaceBG dataset seems very difficult to predict, in fact the models obtained bad results even with the procedures tried, because of the small number of samples and hypoglycemic samples available. The patient has 85 samples, 62 in the training and 23 in the test set. The percentage of hypoglycemic samples is 13% both in the training (8 samples) and in the test set (3 samples). Clearly, the number of hypoglycemic samples is very small and make the learning very difficult for the models. The correlation matrix of the features plus the labels vector highlights some correlation, in fact in the last row, that is the one of the labels vector, some squares are a little bit darker (Figure 5.9).

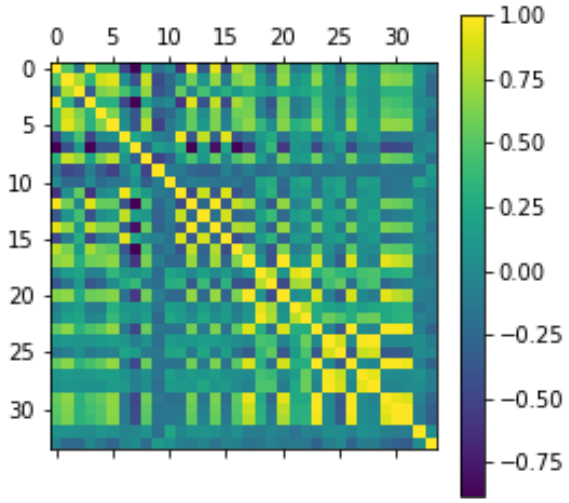

Figure 5.9: Correlation-Matrix of the features plus the labels vector

As predictable, the statistical feature selection give as significant predictive 10 features: *Standard Deviation*, *Coefficient of Variation*, $Range_{TOT}$, $Mage_{+}$, *Hyper Index*, $Mean_1$, $Median_1$, *Mean-Half-1*, *Mean-Half-2*, *Derivative* .

As example of ROC curves analysis on training data, the one of the Logistic Regression without regularization is shown (Figure 5.10). The choice of the probability threshold is made on the *PrecisonVSRecall* curve and the optimal point gives a threshold equal to 0.38, lower than the standard 0.5. In fact, the choice of the optimal point was made to rise the recall metric for the motivation already discussed, so the probability-threshold has been lowered.

For the choice of the hyper-parameters for all the models the procedure followed

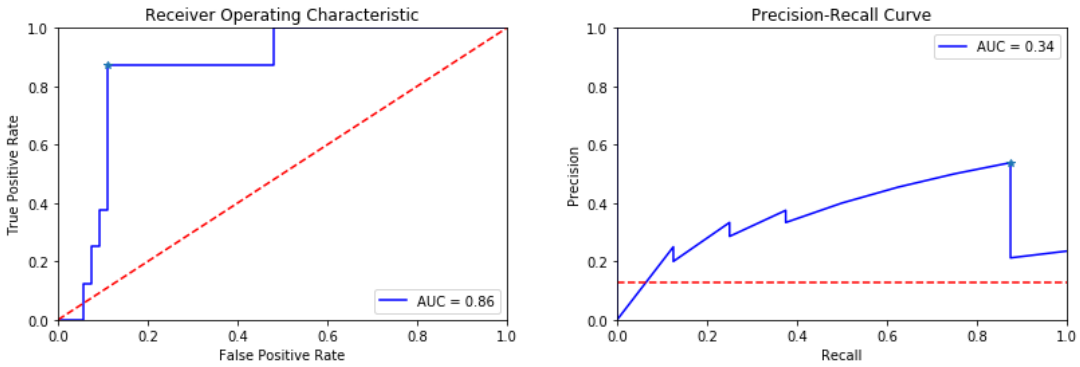

Figure 5.10: ROC and PrecisionVSRecall curves of the Logistic Regression, patient 31 of the ReplaceBG dataset

is the same described for the previous patient. The original and the final parameters are shown in the result tables, except the one of the XGB-Decision Trees that are the follows:

- learning-rate=0.1 (kept the same)
- # estimators=50 (instead of 30)
- max-depth=3 (kept the same)
- min-child-weight=1 (keeping the same)
- gamma=0 (kept the same)
- reg-alpha=0 (kept the same)
- subsample=0.5 (instead of 0.6)
- colsample-bytree=0.3 (instead of 0.6)

The results are in Table 5.14, to compare with the one obtained in the previous analysis without the procedures (Table 5.15). The final results are better than the original for the L2 and L1-Regularized Logistic Regression, worse for the SVM; only the accuracy is improved in the Logistic Regression and in the XGB-Decision, Tree where no true hypoglycemic samples are predicted.
The largest improvement is for L2-Regularized Logistic Regression where 2 out of 3 hypoglycemic events are predicted, but it is clear that with this kind of data is difficult to learn a stable and informative model.

Algorithm	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.61	0.00	0.00	0.0
L2-Reg Logistic Regression (C=0.02)	0.62	0.00	0.00	0.00
L1-Reg Logistic Regression (C=3.6)	0.61	0.12	0.33	0.18
SVM (C=10, $\gamma = 0.01$)	0.61	0.12	0.33	0.18
XGB-Decision Trees	0.70	0.00	0.00	0.00

Table 5.14: Original results on test data, patient 31 of the ReplaceBG dataset

Algorithm	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.83	0.00	0.00	0.00
L2-Reg Logistic Regression (C=0.06)	0.52	0.17	0.67	0.27
L1-Reg Logistic Regression (C=4.5)	0.61	0.20	0.67	0.31
SVM (C=839.2, $\gamma = 0.021$)	0.65	0.00	0.00	0.00
XGB-Decision Trees	0.65	0.00	0.00	0.00

Table 5.15: Results on test data with the model modifications, patient 31 of the ReplaceBG dataset

In conclusion, the results obtained in both population and individual analyses and for both the datasets are not satisfactory. The problem of the prediction of the nocturnal hypoglycemic events does not seem treatable with the implemented methodologies and the reasons could be many. However this work is a preliminary analysis, useful to assess the feasibility and the possibility of this kind of approach. In the next chapter a further analysis on the predictability of the patients is carried out, to verify if the problem of the bad results could be tackle considering only some kind of "predictable" patients.

CHAPTER 6

Patient-predictability analysis

This chapter explains the idea of finding some distinction in the patient that would allow separating the individuals that could be predicted by the models from those unpredictable. To make this distinction two different approaches have been developed: the first performs an analysis of the mean features of each patient, the second makes a distinction based only on the result obtained with the population models applied in each patient separately. Furthermore, the results of the analysis are illustrated, with a consequent discussion on the discrepancy with the expected results.

6.1 Patients-Clustering Idea

The idea of the presence of some clustering in the patients is consequent to the results obtained in both the population and in the patient-specific analysis. In fact, the insufficiency of the population models and the variability of the results in the patient-specific models suggest the fact that maybe some patients are not possible to predict using only the CGM-data of the previous day. However, analyzing the scatterplots of the metrics on each patient, some of them have obtained better results and so maybe are more predictable than others. If the "not-predictable" patients would not be considered in the model estimations maybe the results would be better on the "predictable" ones and potentially on the discarded also. All the results above refer

to the Logistic Regression model because there are no large differences between the different algorithms and so the simpler and more basic one is chosen; the complete report is in the section *Appendix B*.

6.2 Scatterplot-analysis

The first approach tested is the one that starts from the analysis of some result-feature scatterplots. The scatterplot graph contains a point for each sample, plotted with respect to the features that represent the samples. In this way, it is possible to evaluate if there exists any separation in the sample distribution based on the values of some features. In this analysis, the aim is to understand if there exist some characteristics of the patient-datasets that allow distinguishing two types of patients: the "predictable" one and the "not-predictable" one.
First of all, the 249 patients are divides into a *training group* and into a test group of patients, with a division 50-50% that would keep the original distribution of the frequency of the hypoglycemic events in the patient-datasets. In the training group, 124 patients are used to train the *population model MOD0*; the MOD0 are then applied on each patient-dataset separately to find the result metrics and allow to find the criteria of division of the patients. With the training patients that result predictable, new models are trained using only their samples (*MOD*1). The dataset of each test patient is divided in a training set and in a test set of samples with a division 60-40%; the training samples are used to apply the criteria of division found with the training group and the test samples are used to evaluate the performance of the different models.

To find the criteria of division, the means on all the samples of each training patient of all the features used for the prediction are computed, together with the proportion of the hypoglycemic class in each patient-dataset. With these it is possible to create the scatterplots that contain one point for each training patient, considering one metric and one mean feature at time. The boxplots are analyzed to find some thresholds in the mean features using the training set of patients. The thresholds would allow separating the predictable patients from the not predictable patients

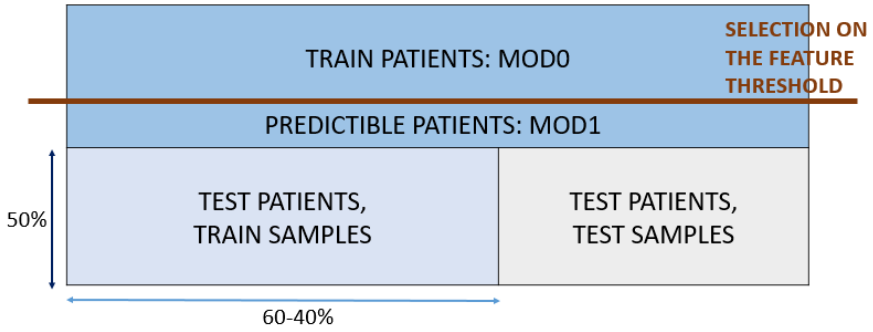

Figure 6.1: Scheme of the division of the patients in the *train* and *test* groups, with the further division of the samples of the test patients in train and test samples.

also in the test set, without computing the results of the population models (MOD0).

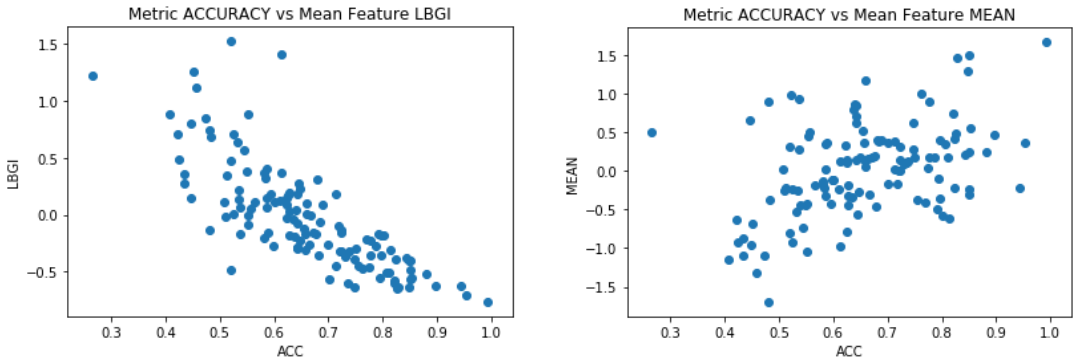

Figure 6.2: Examples of some scatterplots, *accuracy* metric

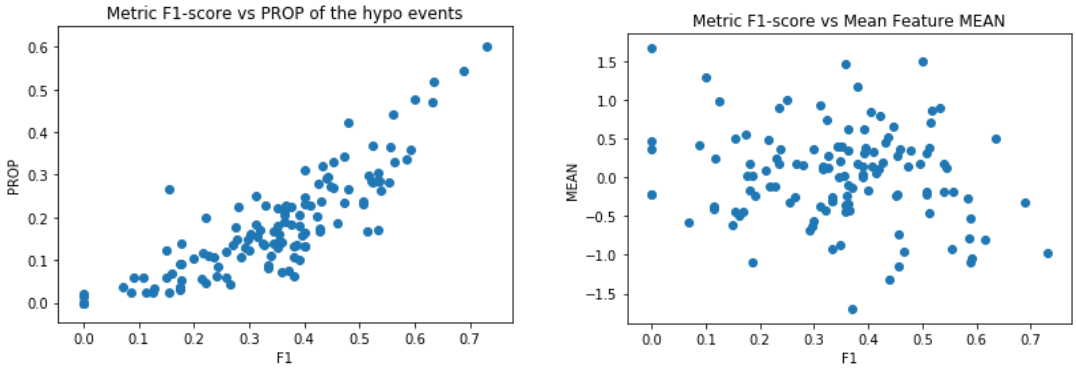

Figure 6.3: Examples of some scatterplots, *F1-score* metric

The mean feature chosen for try this separation for the metric *accuracy (ACC)* is the *Low Blood Glucose Index (LBGI)* and the characteristic *proportion* of the hypoglycemic class in the dataset *(PROP)* is chosen for the metric *F1-score* (the scatterplots on the left in Figures 6.2,6.3). The criteria of division are then applied to the training samples of the test patients, to find the predictable and not predictable ones. With the training samples of the predictable test patient, other new models are trained (*MOD*2). With these samples plus the ones of the training-predictable patients (the ones that have trained the MOD1), also the MOD3 new models are trained (Figure 6.4).

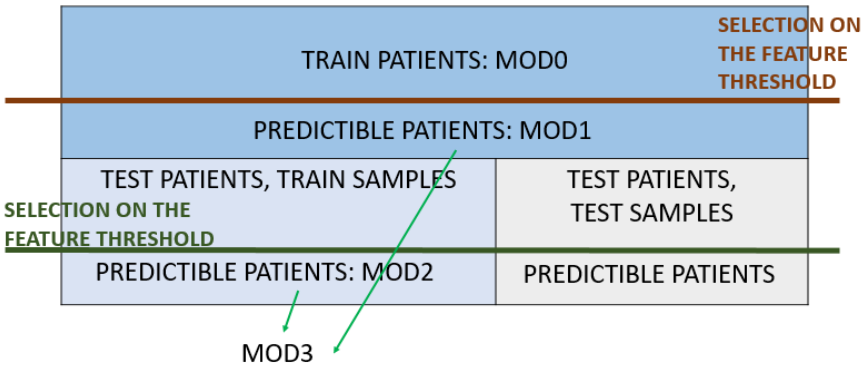

Figure 6.4: Scheme of the division of the train and test groups of patients in the *predictable* and *not predictable* ones based on the feature threshold and indication of the new identified models.

6.2.1 Accuracy-Low Blood Glucose Index analysis

For the LBGI, the threshold is set to -0.50, the patients selected are the ones with a lower value of mean LBGI w.r.t. the threshold and in the training set they are 18 out of 124. The samples used for training the new models MOD1 are 2102 out the total 13798. Figure 6.5 is the boxplot that represents the distribution of the mean feature LBGI of each train patient dataset with respect to the result metric *accuracy* obtained with the application of the MOD0 on each patient dataset. The dotted red line is the mean feature threshold that divides the *predictable* patients, the orange dots under the line, from the *not predictable* ones, the blu dots above the line.

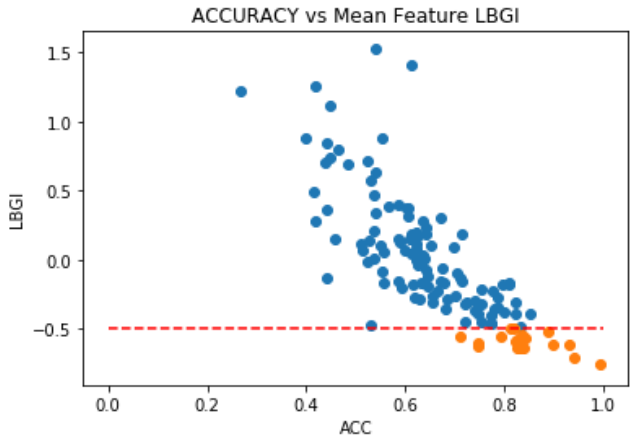

Figure 6.5: Scatterplot of the training patients and division of them based on the mean feature threshold (red dotted line): *predictable* in orange and *not predictable* in blue

The result of the population models and of the MOD1 models on all the training-patients samples and on the training-predictable-patients sample are shown below (Table 6.1). It is possible to notice that the accuracy is improved both in the application of the MOD0 and of the MOD1 on the predictable-train patients but on the contrary, the performances in term of F1-score are worse than the application of the population model on all the training samples.

Model	Accuracy	Precision	Recall	F1-Score
MOD0, all train patients	0.66	0.29	0.65	0.40
MOD0, predictable train patients	0.84	0.19	0.44	0.26
MOD1, all train patients	0.54	0.23	0.70	0.35
MOD1, predictable train patients	0.69	0.14	0.73	0.24

Table 6.1: Result of the Logistic Regression model in the different analyses

Then the mean features of the training samples of the test patients are computed, to apply the threshold on the mean feature LBGI to find the predictable test patients. The number of selected patients is 20 out 125, the number of training samples is 1425 and of test samples is 950. In Figure 6.6 there is the representation of the selected *predictable* patients in the same scatterplot graph used in the previous analysis on the train patients. The expectation is that the dots that represents the selected *predictable*

patients lay under the mean feature threshold (the red dotted line) and on the right side, in correspondence of high values of *accuracy*. Actually three of them are above the threshold but they are all on the right side of the graph, indicating high values of the result metric.
With the training samples of the selected patients, the models MOD2 and MOD3 are trained.

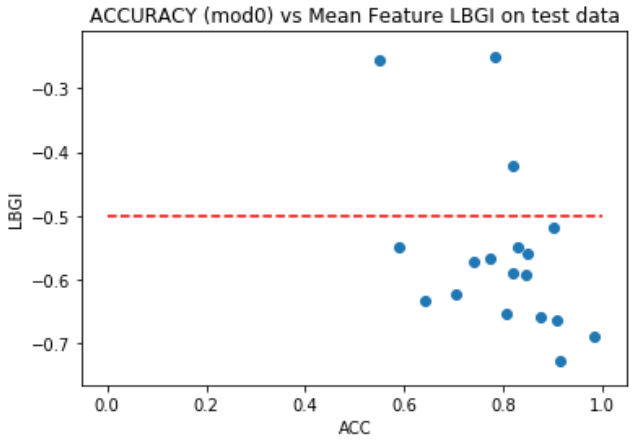

Figure 6.6: Scatterplot of the predictable test patients using MOD0 results

To have a correct evaluation of the model, the test samples of the test patients are always used because they are not included in the training of any models. All the result of the different models analyzed are in Table 6.2 for the result on all the test samples and in Table 6.3 on only the ones of the predictable patients.
The results show that the performance of the new models MOD1, MOD2, MOD3 are not better than the one of the population model, except for the recall that is the only improved. The metric on which the selection was made, the accuracy, is improved in the results of the selected patients with respect to the ones of the totality of the test patients but the F1-score that is more informative is lower. To try a better selection the second analysis is made on the F1-score instead of the accuracy.

All test samples	Accuracy	Precision	Recall	F1-Score
MOD0	0.65	0.28	0.61	0.38
MOD1	0.52	0.22	0.66	0.33
MOD2	0.60	0.24	0.59	0.34
MOD3	0.53	0.22	0.67	0.33

Table 6.2: Result of the Logistic Regression models of the different analyses on all the test samples of the test patients

Predictable test samples	Accuracy	Precision	Recall	F1-Score
MOD0	0.80	0.17	0.44	0.25
MOD1	0.59	0.11	0.61	0.18
MOD2	0.68	0.12	0.52	0.19
MOD3	0.61	0.11	0.62	0.19

Table 6.3: Result of the Logistic Regression models of the different analyses on the test samples of the predictable test patients

6.2.2 F1 score-Proportion of hypoglycemic class analysis

In this analysis, the choice of the metric F1 is made to try to improve the performances and TO overcome the trade-off between the precision and the recall. For the characteristic *proportion of hypoglycemic class (PROP)* the threshold is set to 0.40, the selected patients are the ones with a higher value of PROP w.r.t. the threshold and in the training set they are 7. In Figure 6.7 there is the scatterplot graph of the train patients with the red dotted line that is the mean feature threshold: the orange dots above the line are the *predictable* patients while the blue dots under the line are the *not predictable* ones.

The samples used for training the new models MOD1 are 614 out of 13798.

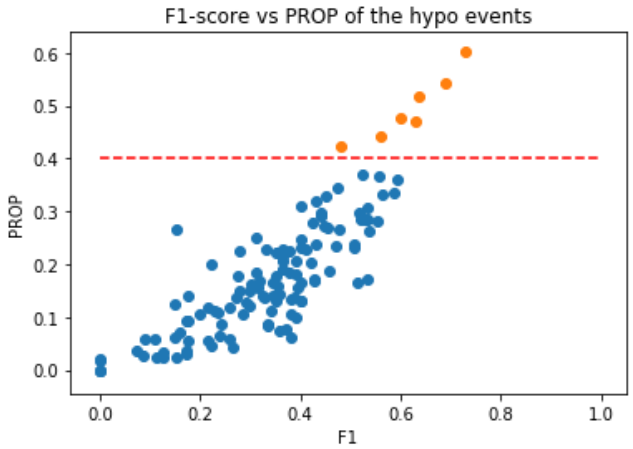

Figure 6.7: Scatterplot of the training patients and division of them based on the mean feature threshold (red dotted line): *predictable* in orange and *not predictable* in blue.

The result of the population models and of the MOD1 models on all the training-patients samples and on the training-predictable-patients sample are shown below (Table 6.4). It is possible to notice that the F1-score is improved both in the application of the MOD0 and of the MOD1 on the predictable-train patients only (0.62 and 0.63). Furthermore, the trade-off between the precision and the recall is not present, especially in the result of the MOD1. The results of MOD1 an all the training samples are worse than the ones of the population model MOD0, probably because 7 patients are too few to give a complete description of the problem and so to have a good performance on all the possible situations present in the totality of the training data.

Model	Accuracy	Precision	Recall	F1-Score
MOD0, all train patients	0.66	0.29	0.65	0.40
MOD0, predictable train patients	0.54	0.52	0.76	0.62
MOD1, all train patients	0.53	0.20	0.55	0.29
MOD1, predictable train patients	0.63	0.61	0.66	0.63

Table 6.4: Result of the Logistic Regression model in the different analyses

The number of selected patients in the test group is 5 out of 125. In Figure 6.8 there is the scatterplot representation of the selected test patients that are expected to lay above the threshold line and on the right. Actually, on the contrary of the expectation, only one of them is in this side of the graph and the other four are under the threshold and also in the left side, indicating low values of result metric F1-score. The number of training samples is 300 and of test samples is 203. The train samples are very few and so probably the new models would not be able to describe the problem. With the training samples of the selected patients, the models MOD2 and MOD3 are trained.

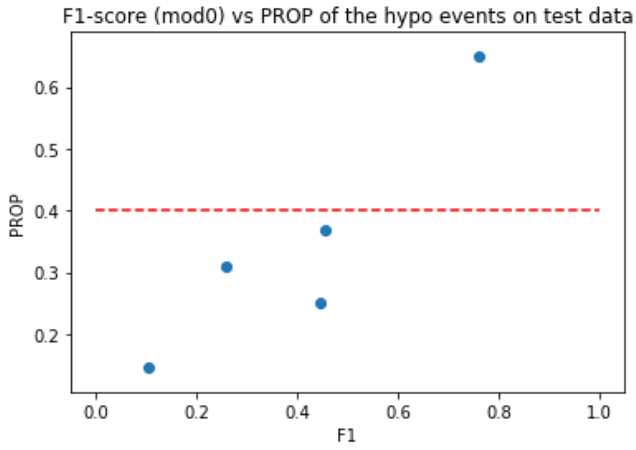

Figure 6.8: Scatterplot of the predictable test patients using MOD0 results

All the result of the different models analyzed are in Table 6.5 for the result on all the test samples and in Table 6.6 on only the ones of the predictable patients.

As it was possible to predict, the results of the F1 score are a little better in the predictable patients, but unfortunately the ones of the new models MOD1, MOD2, MOD3 are worse than the MOD0 one. This trend is present also in the result of the totality of test patients, as expected for the fact of the small number of samples that have trained the new models.

All test samples	Accuracy	Precision	Recall	F1-Score
MOD0	0.65	0.28	0.61	0.38
MOD1	0.53	0.20	0.53	0.29
MOD2	0.68	0.24	0.39	0.30
MOD3	0.63	0.23	0.47	0.30

Table 6.5: Result of the Logistic Regression models of the different analyses on all the test samples of the test patients

Predictable test samples	Accuracy	Precision	Recall	F1-Score
MOD0	0.50	0.37	0.63	0.47
MOD1	0.55	0.38	0.50	0.43
MOD2	0.56	0.36	0.39	0.38
MOD3	0.55	0.38	0.49	0.42

Table 6.6: Result of the Logistic Regression models of the different analyses on the test samples of the predictable test patients

In conclusion, these analyses point out that the possibility of some clustering in the patients based on the characteristic of the patient-dataset has no evidence. The results obtained with the training of new models with the samples of the selected patients are not sufficient and the improvement is not always present, especially in the accuracy-based selection.

In the next paragraph another approach tested is described, based on the results in a reverse way with respect to this first analysis.

6.3 F1-based analysis

The results in Table 6.6 suggest that the selection on the F1-score could be a possible way to improve the results. The problem is that the selection based on the characteristics in the patient-dataset results in too few samples to training new models. The selection could be made on the F1-score obtained with the population model applied to each patient directly, that potentially could give a division of the patients more realistic and powerful. The new approach tried starts with the division of the ensemble dataset in a training and in a test set with a 50-50% division. With the training set, the population models MOD0 are estimated and then they are applied on each patient-training-dataset separately to find the F1-score of each patient on which make the selection. Finally with the training samples of the selected patients the new models MOD1 are trained and then evaluated with the test samples. The scheme of the approach is shown in Figure 6.9.

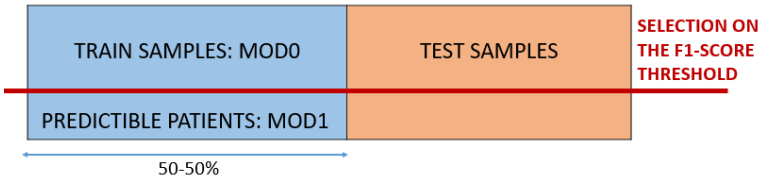

Figure 6.9: Scheme of the F1-analysis tested with the division of all the samples in train and test ones and the selection of the *predictable* patients based on the F1-score threshold

For this analysis, all the patients of the two datasets are used together and a grid of possible F1-threshold is tested for the selection of the predictable patients, from the value 0.35 to 0.75 with a step of 0.05. In Table 6.7 it is possible to see the number of selected patients for all the F1-threshold tried.
In Figure 6.10 there is the boxplot graph of the result metric F1-score obtained with the application of the MOD0 on each patient train-dataset. In this graph there is also the representation of all the patients, plotted with different coloured crosses.

For higher values of F1-threshold, the number of selected patients is very small and so the new models trained with those samples probably would not have the possibility of learning the problem completely. After the new models MOD1 are

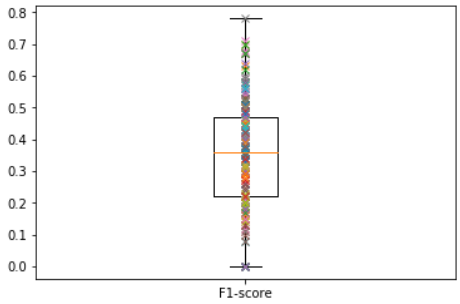

Figure 6.10: Boxplot of each patient F1-score obtained with the application of the MOD0

F1-threshold	# of patients selected
0.35	132
0.40	106
0.45	76
0.50	55
0.55	30
0.60	13
0.65	9
0.70	4
0.75	1

Table 6.7: Number of selected patients for each F1-threshold tried

trained, their performances are evaluated both on all the test samples and on the ones of the predictable patients only. In Figure 6.11 the metrics accuracy and F1-score of the MOD0 and MOD1 are plotted at the variation of the F1-threshold on the test samples of the predictable patients.

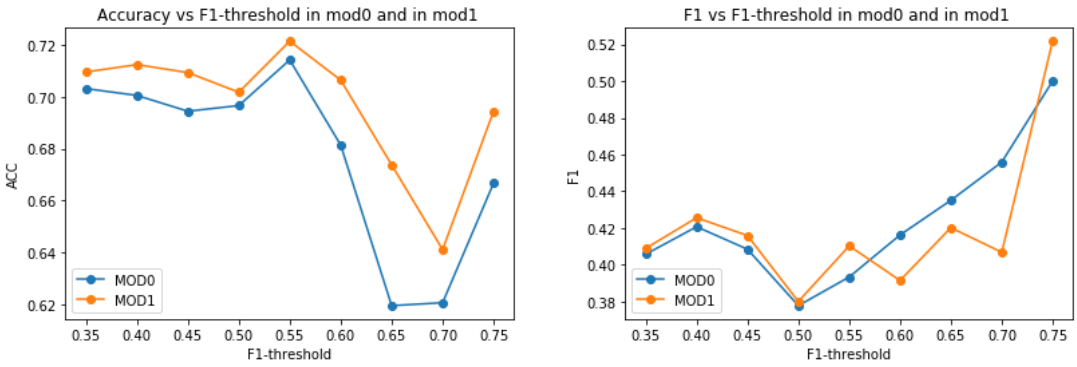

Figure 6.11: Result metrics for each F1-threshold on the test samples of the selected patients

With the increase of the F1-threshold, the accuracy of both the models is slightly increasing at the begging but with higher values, it drops. For the last two thresholds, there is a sudden increase in both the models, maybe for the fact that the samples on which the model is tested are very few and it is possible that neither one hypoglycemic sample is included. This evolution is probably due to the fact that the accuracy is good when the samples correctly predicted are not-hypoglycemic, so selecting the patients that have better performances in the F1-score probably means that they have better results in the prediction of the hypoglycemic samples and worse for the not-hypoglycemic samples.

The F1-score instead has an opposite trend, it slightly decreases in the third and fourth thresholds in both the MOD0 and MOD0, then the MOD0 starts to increase and the MOD1 has a fluctuating increasing evolution. This trend is reasonable considering that the selection is based on the F1-score so the performance of the MOD0 is initially decreasing because there are still many patients to evaluate, and then the F1-score increases with higher values of thresholds.

A compromise between the number of selected patients and good F1 and accuracy performances could be $F1 = 0.55$, that leads to 29 selected patients with 1561 samples in the training set and 1529 samples in the test set. The results of the MOD0 and of the MOD1 in all the test samples and in the test samples of the predictable patients are in Table 6.8.

Test samples	Accuracy	Precision	Recall	F1-Score
MOD0, all patients	0.71	0.33	0.62	0.44
MOD0, predictable patients	0.71	0.30	0.56	0.39
MOD1, all patients	0.68	0.29	0.58	0.39
MOD predictable patients	0.72	0.31	0.59	0.41

Table 6.8: Results of the Logistic Regression models in the different analyses, F1-threshold=0.55

Unfortunately, on the contrary to the expectations, the F1-score of the MOD0 on all the test samples is better than all the other F1-scores, that would be better if the approach would be correct. Besides, the F1-scores of the predictable patients is practically equal in the MOD0 and in the MOD1, indicating that the new model trained with only the samples of the predictable patients does not learn better the problem as one could think. This means that the clustering of the patients based on the F1-threshold is not the correct way to find some possible distinction in the patients.

In the end, both the approaches tried to cluster the patients in predictable and not-predictable ones have failed. The problem could be the fact that the methodologies analyzed were not the correct way to find this separation or maybe it is the fact that there is not any division in the patients, so there is no way to improve the results with these approaches.

In the next chapter, after a discussion on the results obtained, other possible further analyses are presented. In fact, from the conclusion of this preliminary work on the problem of the prediction of the nocturnal hypoglycemic events, many other approaches can be evaluated to achieve more satisfactory outcomes.

CHAPTER 7

Conclusion and Further Analyses

All the methodologies and strategies applied to solve the task of the prediction of the nocturnal hypoglycemic events using the CGM data of the preceding day only have been analyzed, therefore it is possible to make the conclusions. Finally, possible further analyses and methodologies that could complete this work are presented.

7.1 Discussion on the result and conclusion

Unfortunately, all the different methodologies applied in the different approaches, and also with different datasets, have not shown satisfying results. The algorithms seem not appropriate to predict the nocturnal hypoglycemic events and so to solve the task. The causes of this conclusion can be many and of different nature:

- Features not adequate to predict the hypoglycemic events: it could be that other features derived by the CGM data of the day can be more informative about the problem to solve;
- The CGM data of the preceding day only are not sufficient to make the prediction: it could be that the information obtained from the preceding day is not enough to learn how the nocturnal hypoglycemic events are allocated and when they occur but maybe adding the records of other preceding days could

improve the prediction. It could be also that the CGM data alone cannot solve the task but maybe with other information of the patient-day, as the insulin doses and the carbohydrate intakes, the problem becomes feasible;

- The algorithms themselves could not learn the problem postulated in this way but maybe other models could be abler;

- The problem is too much complex, maybe because it is not possible predicting long-term nocturnal hypoglycemic events;

- The problem is ill-posed and other metrics correlated with the nocturnal hypoglycemic events could be more easily predictable. An example of another way to solve the task could be the prediction of the LBGI of the night.

7.2 Further Analyses

From the previous conclusion about the results of the work, other options and methodologies could be evaluated to perform future analyses. New algorithms can be tested as the Recurrent Neural Networks (RNN), the Long Short-Term Memory-NNs (LSTM-NN), the Temporal Convolutional-NNs (TC-NN). All these algorithms are based on the recognition of temporal patterns in the data and on the use of them to compute the output layer.
The use of the Neural Networks in the task of the short-term prediction of the glucose level and also in the prediction of the hypoglycemic events is already present in literature and has shown good results with respect of the traditional Machine Learning algorithms, models and filters [Pérez-Gandía *et al.*, 2010], [Zecchin *et al.*, 2015], [Bertachi *et al.*, 2018].
The RNNs are used a lot in time-series forecasting and also in the specific case of the prediction of the glycemic level [Allam *et al.*, 2011],[Doike *et al.*, 2018]. The results obtained demonstrate the ability to forecast the glucose level with prediction horizon long at maximum 60 minutes but not more, so the use of this algorithm in the events prediction could be an interesting variation to the RNNs already proposed.
The LSTM-NNs are RNNs with the ability to keep in memory some information

about the past states of the network and weighting them in order to achieve best results. In this way, it is possible to learn long-term dependencies that would enrich the prediction. This kind of framework has been applied to the prediction of the glucose level with a short-term horizon, maximum equal to 60 minutes [Sun *et al.*, 2018], [Li *et al.*, 2019] [Mosquera-Lopez *et al.*, 2019]. The results are very encouraging and demonstrate the ability of the LSTM-NNs of capturing the trends and the patterns of the CGM data series.

The Temporal Convolutional Neural Networks are analogous of the Convolutional Neural Networks that employ casual convolutions and dilations so that they are adaptive for sequential data due to their temporality and large receptive fields [Bai *et al.*, 2018]. For these reasons, the TC-architecture could be adequate to solve the prediction task using only the CGM data

In addiction also an *autoencoder (AE)* approach could be tested, to catch the most informative properties of the CGM data for the prediction task proposed. The autoencoder system is composed of two models, the first is called the *encoder*, it is a RNN or a CNN and it has to carry out the AE features. These features are not attributable to physical features but are the most appropriate to solve the final task. Then these features are put in input to a second model, often a SVM classifier, that has the aim of make the prediction, trained on data depicted by the AE features [Hinton and Salakhutdinov, 2006].

APPENDIX A

Result of the population models on each patient datasets

ReplaceBG Dataset

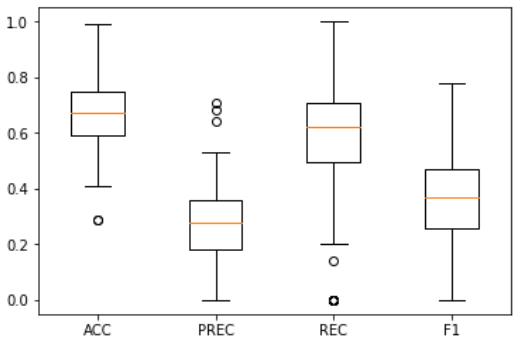

Figure A.1: Boxplots of the result metrics of the population model *Logistic Regression*

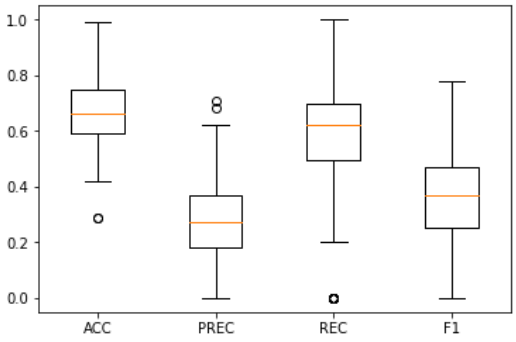

Figure A.2: Boxplots of the result metrics of the population model *L2-Regularized LR*

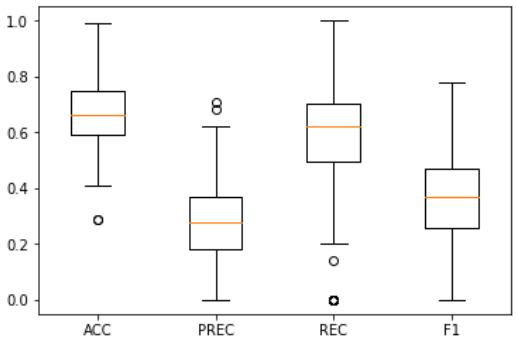

Figure A.3: Boxplots of the result metrics of the population model *L1-Regularized LR*

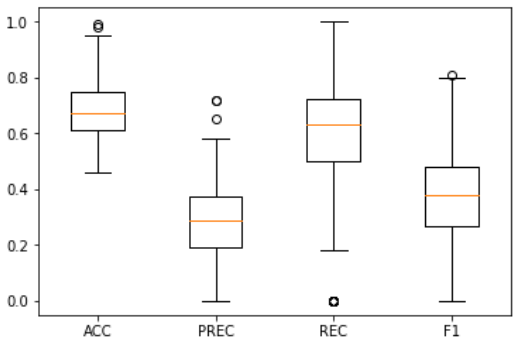

Figure A.4: Boxplots of the result metrics of the population model *Support Vector Machine*

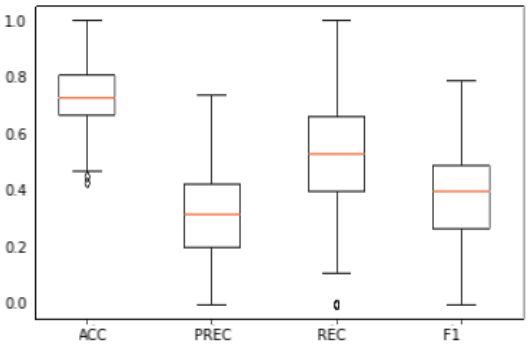

Figure A.5: Boxplots of the result metrics of the population model *Gradient Boosted Decision Trees*

Real Dataset

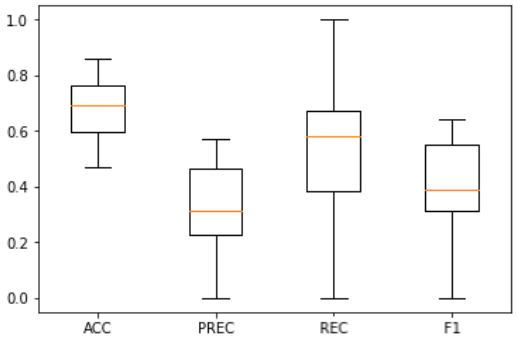

Figure A.6: Boxplots of the result metrics of the population model *Logistic Regression*

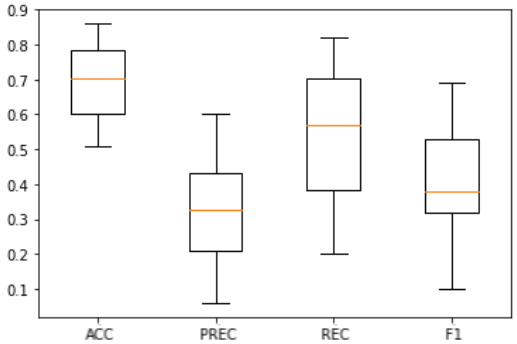

Figure A.7: Boxplots of the result metrics of the population model *L2-Regularized LR*

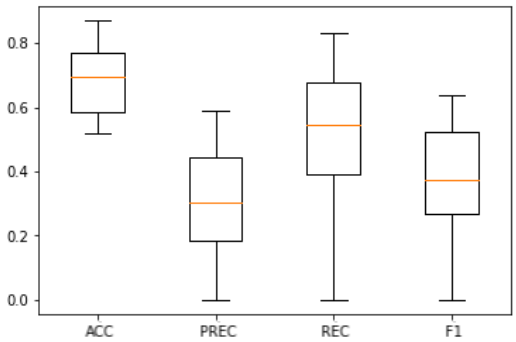

Figure A.8: Boxplots of the result metrics of the population model *L1-Regularized LR*

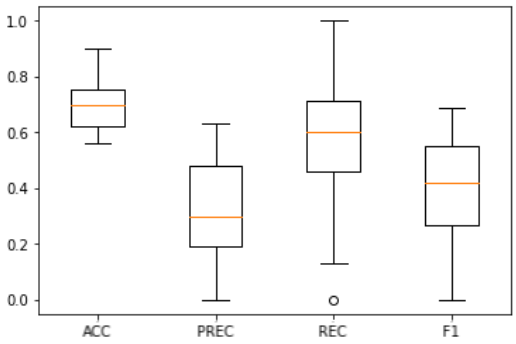

Figure A.9: Boxplots of the result metrics of the population model *Support Vector Machine*

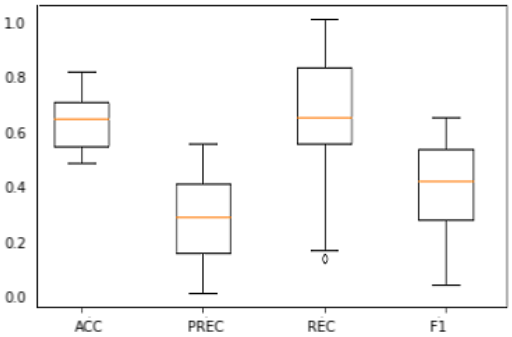

Figure A.10: Boxplots of the result metrics of the population model *Gradient Boosted Decision Trees*

APPENDIX B

Result of the patients-clustering analysis

B.1 Boxplot-analysis

B.1.1 Accuracy-Low Blood Glucose Index analysis

Logistic Regression

Results of the MOD0 and MOD1 on the training group of patients

Model	Accuracy	Precision	Recall	F1-Score
MOD0, all train patients	0.66	0.29	0.65	0.40
MOD0, predictable train patients	0.84	0.19	0.44	0.26
MOD1, all train patients	0.54	0.23	0.70	0.35
MOD1, predictable train patients	0.69	0.14	0.73	0.24

Table B.1: Result of the *Logistic Regression* model in the different analysis

Results of the MOD0, MOD1, MOD2, MOD3 on the test group of patients

All test samples	Accuracy	Precision	Recall	F1-Score
MOD0	0.65	0.28	0.61	0.38
MOD1	0.52	0.22	0.66	0.33
MOD2	0.60	0.24	0.59	0.34
MOD3	0.53	0.22	0.67	0.33

Table B.2: Result of the *Logistic Regression* models of the different analysis on all the test samples of the test patients

Predictable test samples	Accuracy	Precision	Recall	F1-Score
MOD0	0.80	0.17	0.44	0.25
MOD1	0.59	0.11	0.61	0.18
MOD 2	0.68	0.12	0.52	0.19
MOD3	0.61	0.11	0.62	0.19

Table B.3: Result of the *Logistic Regression* models of the different analysis on the test samples of the predictable test patients

L2-Regularized Logistic Regression

Results of the MOD0 and MOD1 on the training group of patients

Model	Accuracy	Precision	Recall	F1-Score
MOD0, all train patients	0.66	0.29	0.65	0.40
MOD0, predictable train patients	0.84	0.18	0.43	0.26
MOD1, all train patients	0.48	0.23	0.82	0.36
MOD1, predictable train patients	0.67	0.13	0.72	0.22

Table B.4: Result of the *L2-Regularized LR* model in the different analysis

Results of the MOD0, MOD1, MOD2, MOD3 on the test group of patients

All test samples	Accuracy	Precision	Recall	F1-Score
MOD0	0.65	0.28	0.61	0.38
MOD1	0.46	0.22	0.79	0.34
MOD2	0.60	0.24	0.60	0.35
MOD3	0.50	0.23	0.79	0.36

Table B.5: Result of the *L2-Regularized LR* models of the different analysis on all the test samples of the test patients

Predictable test samples	Accuracy	Precision	Recall	F1-Score
MOD0	0.81	0.19	0.45	0.26
MOD1	0.60	0.11	0.61	0.19
MOD2	0.68	0.12	0.50	0.19
MOD3	0.61	0.12	0.64	0.20

Table B.6: Result of the *L2-Regularized LR* models of the different analysis on the test samples of the predictable test patients

L1-Regularized Logistic Regression

Results of the MOD0 and MOD1 on the training group of patients

Model	Accuracy	Precision	Recall	F1-Score
MOD0, all train patients	0.66	0.29	0.65	0.40
MOD0, predictable train patients	0.82	0.16	0.41	0.23
MOD1, all train patients	0.55	0.24	0.72	0.36
MOD1, predictable train patients	0.69	0.14	0.69	0.23

Table B.7: Result of the *L1-Regularized LR* model in the different analysis

Results of the MOD0, MOD1, MOD2, MOD3 on the test group of patients

All test samples	Accuracy	Precision	Recall	F1-Score
MOD0	0.65	0.28	0.62	0.38
MOD1	0.54	0.23	0.71	0.35
MOD2	0.60	0.25	0.66	0.37
MOD3	0.54	0.24	0.74	0.36

Table B.8: Result of the *L1-Regularized LR* models of the different analysis on all the test samples of the test patients

Predictable test samples	Accuracy	Precision	Recall	F1-Score
MOD0	0.79	0.16	0.41	0.23
MOD1	0.63	0.12	0.62	0.20
MOD2	0.70	0.13	0.55	0.22
MOD3	0.62	0.12	0.67	0.21

Table B.9: Result of the *L1-Regularized LR* models of the different analysis on the test samples of the predictable test patients

Support Vector Machine

Results of the MOD0 and MOD1 on the training group of patients

Model	Accuracy	Precision	Recall	F1-Score
MOD0, all train patients	0.66	0.30	0.68	0.41
MOD0, predictable train patients	0.82	0.17	0.45	0.24
MOD1, all train patients	0.47	0.22	0.83	0.35
MOD1, predictable train patients	0.64	0.12	0.72	0.20

Table B.10: Result of the *Support Vector Machine* model in the different analysis

Results of the MOD0, MOD1, MOD2, MOD3 on the test group of patients

All test samples	Accuracy	Precision	Recall	F1-Score
MOD0	0.64	0.28	0.62	0.38
MOD1	0.45	0.22	0.80	0.34
MOD2	0.60	0.26	0.67	0.37
MOD3	0.50	0.23	0.79	0.36

Table B.11: Result of the *Support Vector Machine* models of the different analysis on all the test samples of the test patients

Predictable test samples	Accuracy	Precision	Recall	F1-Score
MOD0	0.78	0.16	0.44	0.23
MOD1	0.57	0.11	0.62	0.18
MOD2	0.69	0.14	0.58	0.22
MOD3	0.62	0.12	0.61	0.20

Table B.12: Result of the *Support Vector Machine* models of the different analysis on the test samples of the predictable test patients

Gradient Boosted Decision Trees

Results of the MOD0 and MOD1 on the training group of patients

Model	Accuracy	Precision	Recall	F1-Score
MOD0, all train patients	0.89	0.62	0.92	0.74
MOD0, predictable train patients	0.93	0.16	0.77	0.58
MOD1, all train patients	0.77	0.32	0.27	0.29
MOD1, predictable train patients	0.97	0.69	1.00	0.82

Table B.13: Result of the *Gradient Boosted Decision Trees* model in the different analysis

Results of the MOD0, MOD1, MOD2, MOD3 on the test group of patients

All test samples	Accuracy	Precision	Recall	F1-Score
MOD0	0.73	0.31	0.45	0.37
MOD1	0.69	0.27	0.44	0.33
MOD2	0.70	0.28	0.44	0.34
MOD3	0.70	0.30	0.48	0.37

Table B.14: Result of the *Gradient Boosted Decision Trees* models of the different analysis on all the test samples of the test patients

Predictable test samples	Accuracy	Precision	Recall	F1-Score
MOD0	0.82	0.16	0.31	0.21
MOD1	0.81	0.15	0.34	0.21
MOD2	0.79	0.14	0.34	0.20
MOD3	0.80	0.17	0.42	0.24

Table B.15: Result of the *Gradient Boosted Decision Trees* models of the different analysis on the test samples of the predictable test patients

B.1.2 F1 score-Proportion of hypoglycemic class analysis

Logistic Regression

Results of the MOD0 and MOD1 on the training group of patients

Model	Accuracy	Precision	Recall	F1-Score
MOD0, all train patients	0.66	0.29	0.65	0.40
MOD0, predictable train patients	0.54	0.52	0.76	0.62
MOD1, all train patients	0.53	0.20	0.55	0.29
MOD1, predictable train patients	0.63	0.61	0.66	0.63

Table B.16: Result of the*Logistic Regression* model in the different analysis

Results of the MOD0, MOD1, MOD2, MOD3 on the test group of patients

All test samples	Accuracy	Precision	Recall	F1-Score
MOD0	0.65	0.28	0.61	0.38
MOD1	0.53	0.20	0.53	0.29
MOD2	0.68	0.24	0.39	0.30
MOD3	0.63	0.23	0.47	0.30

Table B.17: Result of the *Logistic Regression* models of the different analysis on all the test samples of the test patients

Predictable test samples	Accuracy	Precision	Recall	F1-Score
MOD0	0.50	0.37	0.63	0.47
MOD1	0.55	0.38	0.50	0.43
MOD2	0.56	0.36	0.39	0.38
MOD3	0.55	0.38	0.49	0.42

Table B.18: Result of the *Logistic Regression* models of the different analysis on the test samples of the predictable test patients

L2-Regularized Logistic Regression

Results of the MOD0 and MOD1 on the training group of patients

Model	Accuracy	Precision	Recall	F1-Score
MOD0, all train patients	0.66	0.29	0.65	0.40
MOD0, predictable train patients	0.54	0.52	0.76	0.62
MOD1, all train patients	0.61	0.24	0.53	0.33
MOD1, predictable train patients	0.60	0.59	0.58	0.59

Table B.19: Result of the *L2-Regularized LR* model in the different analysis

Results of the MOD0, MOD1, MOD2, MOD3 on the test group of patients

All test samples	Accuracy	Precision	Recall	F1-Score
MOD0	0.65	0.28	0.61	0.38
MOD1	0.62	0.24	0.52	0.33
MOD2	0.67	0.24	0.39	0.30
MOD3	0.66	0.25	0.46	0.33

Table B.20: Result of the *L2-Regularized LR* models of the different analysis on all the test samples of the test patients

Predictable test samples	Accuracy	Precision	Recall	F1-Score
MOD0	0.52	0.38	0.63	0.47
MOD1	0.56	0.38	0.44	0.41
MOD2	0.57	0.39	0.41	0.40
MOD3	0.56	0.37	0.41	0.39

Table B.21: Result of the *L2-Regularized LR* models of the different analysis on the test samples of the predictable test patients

L1-Regularized Logistic Regression

Results of the MOD0 and MOD1 on the training group of patients

Model	Accuracy	Precision	Recall	F1-Score
MOD0, all train patients	0.66	0.29	0.65	0.40
MOD0, predictable train patients	0.54	0.52	0.78	0.62
MOD1, all train patients	0.58	0.24	0.65	0.35
MOD1, predictable train patients	0.57	0.55	0.64	0.59

Table B.22: Result of the *L1-Regularized LR* model in the different analysis

Results of the MOD0, MOD1, MOD2, MOD3 on the test group of patients

All test samples	Accuracy	Precision	Recall	F1-Score
MOD0	0.65	0.28	0.62	0.38
MOD1	0.58	0.24	0.62	0.34
MOD2	0.62	0.24	0.53	0.33
MOD3	0.60	0.24	0.59	0.34

Table B.23: Result of the *L1-Regularized LR* models of the different analysis on all the test samples of the test patients

Predictable test samples	Accuracy	Precision	Recall	F1-Score
MOD0	0.51	0.38	0.64	0.47
MOD1	0.52	0.36	0.51	0.43
MOD2	0.53	0.33	0.36	0.34
MOD3	0.51	0.34	0.47	0.40

Table B.24: Result of the *L1-Regularized LR* models of the different analysis on the test samples of the predictable test patients

Support Vector Machine

Results of the MOD0 and MOD1 on the training group of patients

Model	Accuracy	Precision	Recall	F1-Score
MOD0, all train patients	0.66	0.30	0.68	0.41
MOD0, predictable train patients	0.53	0.51	0.78	0.62
MOD1, all train patients	0.18	0.18	1.00	0.30
MOD1, predictable train patients	0.49	0.49	1.00	0.65

Table B.25: Result of the *Support Vector Machine* model in the different analysis

Results of the MOD0, MOD1, MOD2, MOD3 on the test group of patients

All test samples	Accuracy	Precision	Recall	F1-Score
MOD0	0.64	0.28	0.62	0.38
MOD1	0.18	0.18	1.00	0.30
MOD2	0.72	0.27	0.35	0.31
MOD3	0.68	0.27	0.45	0.34

Table B.26: Result of the *Support Vector Machine* models of the different analysis on all the test samples of the test patients

Predictable test samples	Accuracy	Precision	Recall	F1-Score
MOD0	0.49	0.37	0.66	0.47
MOD1	0.34	0.34	1.00	0.51
MOD2	0.60	0.43	0.46	0.44
MOD3	0.56	0.37	0.40	0.38

Table B.27: Result of the *Support Vector Machine* models of the different analysis on the test samples of the predictable test patients

Gradient Boosted Decision Trees

Results of the MOD0 and MOD1 on the training group of patients

Model	Accuracy	Precision	Recall	F1-Score
MOD0, all train patients	0.89	0.62	0.92	0.74
MOD0, predictable train patients	0.85	0.78	0.94	0.86
MOD1, all train patients	0.52	0.21	0.63	0.352
MOD1, predictable train patients	0.98	0.98	0.98	0.98

Table B.28: Result of the *Gradient Boosted Decision Trees* model in the different analysis

Results of the MOD0, MOD1, MOD2, MOD3 on the test group of patients

All test samples	Accuracy	Precision	Recall	F1-Score
MOD0	0.73	0.31	0.45	0.37
MOD1	0.52	0.20	0.58	0.30
MOD2	0.58	0.20	0.46	0.28
MOD3	0.61	0.23	0.51	0.32

Table B.29: Result of the *Gradient Boosted Decision Trees* models of the different analysis on all the test samples of the test patients

Predictable test samples	Accuracy	Precision	Recall	F1-Score
MOD0	0.57	0.39	0.41	0.40
MOD1	0.54	0.38	0.50	0.43
MOD2	0.52	0.35	0.47	0.40
MOD3	0.52	0.34	0.41	0.37

Table B.30: Result of the *Gradient Boosted Decision Trees* models of the different analysis on the test samples of the predictable test patients

B.2 F1-based analysis

B.2.1 Logistic Regression

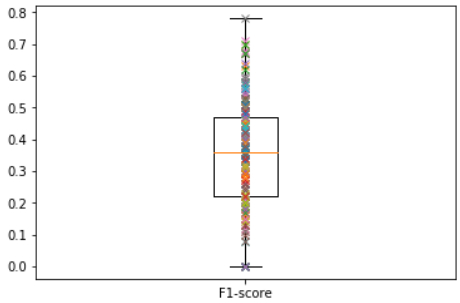

Figure B.1: Boxplot of the *Logistic Regression MOD0* F1-score

F1-threshold	# of patients selected
0.35	132
0.40	106
0.45	76
0.50	55
0.55	30
0.60	13
0.65	9
0.70	4
0.75	1

Table B.31: Number of selected patients for each F1-threshold, *Logistic Regression* model

Test samples	Accuracy	Precision	Recall	F1-Score
MOD0, all patients	0.71	0.33	0.62	0.44
MOD0, predictable patients	0.71	0.30	0.56	0.39
MOD1, all patients	0.68	0.29	0.58	0.39
MOD predictable patients	0.72	0.31	0.59	0.41

Table B.32: Result of the *Logistic Regression* models in the different analysis, F1-threshold=0.55

B.2.2 L2-Regularized Logistic Regression

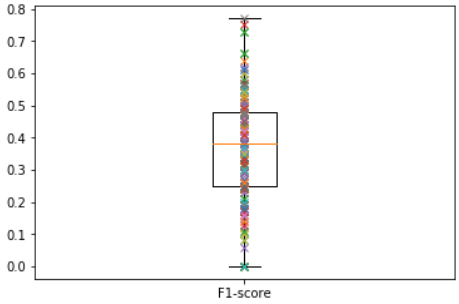

Figure B.2: Boxplot of the *L2-reg LR MOD0* F1-score

F1-threshold	# of patients selected
0.35	147
0.40	112
0.45	83
0.50	48
0.55	25
0.60	12
0.65	6
0.70	4
0.75	2

Table B.33: Number of selected patients for each F1-threshold, *L2-reg LR* model

Test samples	Accuracy	Precision	Recall	F1-Score
MOD0, all patients	0.71	0.33	0.63	0.44
MOD0, predictable patients	0.74	0.27	0.52	0.36
MOD1, all patients	0.65	0.27	0.60	0.37
MOD predictable patients	0.71	0.25	0.55	0.34

Table B.34: Result of the *L2-reg LR* models in the different analysis, F1-threshold=0.55

B.2.3 L1-Regularized Logistic Regression

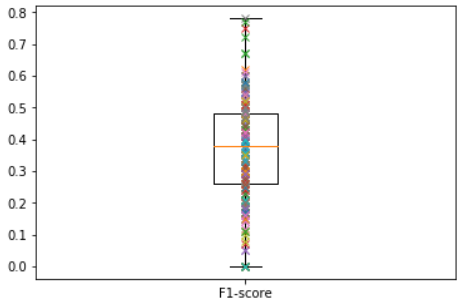

Figure B.3: Boxplot of the *L1-reg LR MOD0* F1-score

F1-threshold	# of patients selected
0.35	149
0.40	111
0.45	82
0.50	52
0.55	29
0.60	10
0.65	6
0.70	4
0.75	3

Table B.35: Number of selected patients for each F1-threshold, *L1-reg LR* model

Test samples	Accuracy	Precision	Recall	F1-Score
MOD0, all patients	0.70	0.33	0.63	0.44
MOD0, predictable patients	0.73	0.26	0.51	0.34
MOD1, all patients	0.65	0.27	0.60	0.37
MOD predictable patients	0.72	0.26	0.56	0.35

Table B.36: Result of the *L1-reg LR* models in the different analysis, F1-threshold=0.55

B.2.4 Support Vector Machine

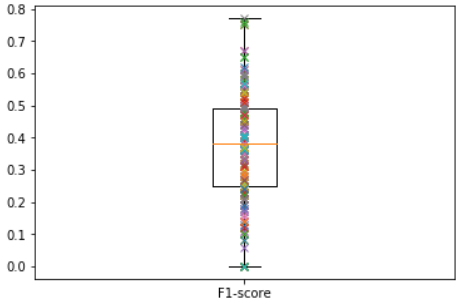

Figure B.4: Boxplot of the *SVM MOD0* F1-score

F1-threshold	# of patients selected
0.35	157
0.40	116
0.45	92
0.50	54
0.55	33
0.60	13
0.65	8
0.70	4
0.75	4

Table B.37: Number of selected patients for each F1-threshold, *SVM* model

Test samples	Accuracy	Precision	Recall	F1-Score
MOD0, all patients	0.71	0.34	0.63	0.44
MOD0, predictable patients	0.73	0.25	0.52	0.34
MOD1, all patients	0.68	0.29	0.57	0.38
MOD predictable patients	0.75	0.26	0.50	0.34

Table B.38: Result of the *SVM* models in the different analysis, F1-threshold=0.55

B.2.5 Gradient Boosted Decision Trees

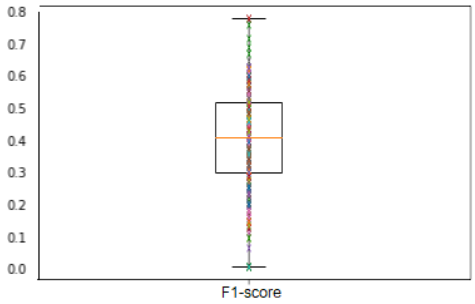

Figure B.5: Boxplot of the *Gradient Boosted Decision Trees MOD0* F1-score

F1-threshold	# of patients selected
0.35	156
0.40	126
0.45	95
0.50	71
0.55	43
0.60	18
0.65	6
0.70	4
0.75	3

Table B.39: Number of selected patients for each F1-threshold, *Gradient Boosted Decision Trees* model

Test samples	Accuracy	Precision	Recall	F1-Score
MOD0, all patients	0.72	0.35	0.62	0.44
MOD0, predictable patients	0.74	0.29	0.55	0.38
MOD1, all patients	0.63	0.26	0.61	0.37
MOD predictable patients	0.68	0.24	0.55	0.33

Table B.40: Result of the *Gradient Boosted Decision Trees* models in the different analysis, F1-threshold=0.55

Bibliography

ALEPPO, G., RUEDY, K. J., RIDDLESWORTH, T. D., KRUGER, D. F., PETERS, A. L., HIRSCH, I., BERGENSTAL, R. M., TOSCHI, E., AHMANN, A. J., SHAH, V. N., RICKELS, M. R., BODE, B. W., PHILIS-TSIMIKAS, A., POP-BUSUI, R., RODRIGUEZ, H., EYTH, E., BHARGAVA, A., KOLLMAN, C. and BECK, R. W. (2017), «REPLACE-BG: A Randomized Trial Comparing Continuous Glucose Monitoring With and Without Routine Blood Glucose Monitoring in Adults With Well-Controlled Type 1 Diabetes», *Diabetes Care*, vol. 40 (4), p. 538–545. (Cited at pages 9 e 17)

ALLAM, F., NOSSAI, Z., GOMMA, H., IBRAHIM, I. and ABDELSALAM, M. (2011), «A Recurrent Neural Network Approach for Predicting Glucose Concentration in Type-1 Diabetic Patients», *Iliadis L., Jayne C. (eds) Engineering Applications of Neural Networks, EANN 2011, AIAI 2011*. (Cited at page 90)

BAI, S., KOLTER, J. Z. and KOLTUN, V. (2018), «An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling», *arXiv e-prints*. (Cited at page 91)

BERTACHI, A., BIAGI, L., CONTRERAS, I., LUO, N. and VEHI, J. (2018), «Prediction of blood glucose levels and nocturnal hypoglycemia using physiological models and artificial neural networks», in «KHD@IJCAI», vol. 12. (Cited at pages 12, 13, 25 e 90)

BLOOM, D., CAFIERO, E., JANÉ-LLOPIS, E., ABRAHAMS-GESSEL, S., BLOOM, L. and FATHIMA, S. (2011), «The global economic burden of noncommunicable diseases (Working Paper Series)», *Geneva: Harvard School of Public Health and World Economic Forum*, vol. 33 (8), p. 811–31. (Cited at page 2)

BOURNE, R., STEVENS, G., RA, R. W., SMITH, J., FLAXMAN, S. and PRICE, H. (2013), «Causes of vision loss worldwide, 1990-2010: a systematic analysis», *Lancet Global Health*. (Cited at page 4)

BREMER, T. and GOUGH, D. (1999), «Is blood glucose predictable from previous values? a solicitation for data», *Diabetes*, vol. 48 (3), p. 445–451. (Cited at page 11)

BRUTTOMESSO, D., FARRET, A., COSTA, S., MARESCOTTI, M., VETTORE, M. and ET AL, A. A. (2009), «Closed-Loop Artificial Pancreas Using Subcutaneous Glucose Sensing and Insulin Delivery and a Model Predictive Control Algorithm: Preliminary Studies in Padova and Montpellier», *Journal of Diabetes Science and Technology*, vol. 3 (5), p. 1014–21. (Cited at page 11)

BUCKINGHAM, B., CHASE, H., DASSAU, E., COBRY, E., CLINTON, P., GAGE, V., CASWELL, K., WILKINSON, J., CAMERON, F., LEE, H., BEQUETTE, B. and DOYLE, F. (2010), «Prevention of nocturnal hypoglycemia using predictive alarm algorithms and insulin pump suspension», *Diabetes Care*, vol. 33, p. 1013–7. (Cited at page 11)

BUCKINGHAM, B., CAMERON, F., CALHOUN, P., MAAHS, D., WILSON, D., CHASE, H., BEQUETTE, B., LUM, J., SIBAYAN, J., BECK, R. and KOLLMAN, C. (2013), «Outpatient safety assessment of an in-home predictive low-glucose suspend system with type 1 diabetes subjects at elevated risk of nocturnal hypoglycemia», *Diabetes Technology & Therapeutics*, vol. 15, p. 622–7. (Cited at page 11)

CASTLE, J. R. and JACOBS, P. G. (2016), «Nonadjunctive Use of Continuous Glucose Monitoring for Diabetes Treatment Decisions», *Journal of Diabetes Science and Technology*, vol. 10 (5), p. 1169–1173. (Cited at page 9)

CHANDRAN, S. R., TAY, W. L., LYE, W. K., LIM, L. L., RATNASINGAM, J., TAN, A. T. B. and GARDNER., D. S. (2018), «Beyond HbA1c: Comparing Glycemic

Variability and Glycemic Indices in Predicting Hypoglycemia in Type 1 and Type 2 Diabetes», *Diabetes Technology & Therapeutics*, vol. 20 (5). (Cited at page 25)

CHOLEAU, C., DOKLADAL, P., KLEIN, J.-C., WARD, W. K., WILSON, G. S. and REACH, G. (2002), «Prevention of hypoglycemia using risk assessment with a continuous glucose monitoring system», *Diabetes*, vol. 51 (11), p. 3263–3273. (Cited at page 25)

CHOUDHARY, P., OLSEN, B., CONGET, I., WELSH, J., VORRINK, L. and SHIN, J. (2016), «Hypoglycemia prevention and user acceptance of an insulin pump system with predictive low glucose management», *Diabetes Technology & Therapeutics*, vol. 18, p. 288–9. (Cited at page 11)

COX, D. J., GONDER-FREDERICK, L., RITTERBAND, L., CLARKE, W. and KOVATCHEV, B. P. (2007), «Prediction of severe hypoglycemia», *Diabetes Care*, vol. 30 (6), p. 1370–1373. (Cited at page 25)

DASSAU, E., CAMERON, F., LEE, H., BEQUETTE, B. W., ZISSER, H., JOVANOVIČ, L., CHASE, H. P., WILSON, D. M., BUCKINGHAM, B. A. and DOYLE, F. J. (2010), «Real-Time Hypoglycemia Prediction Suite Using Continuous Glucose Monitoring», *Diabetes Care*, vol. 33 (6), p. 1249–1254. (Cited at page 12)

DOIKE, T., HAYASHI, K., ARATA, S., MOHAMMAD, K. N., KOBAYASHI, A. and NIITSU, K. (2018), «A Blood Glucose Level Prediction System Using Machine Learning Based on Recurrent Neural Network for Hypoglycemia Prevention», in «16th IEEE International New Circuits and Systems Conference (NEWCAS), Montreal, QC», . (Cited at pages 13 e 90)

EL-LABOUDI, A., I.F., G., D.G., J. and N.S., O. (2016), «Measures of glycemic variability in type 1 diabetes and the effect of real-time continuous glucose monitoring», *Diabetes Technology & Therapeutics*, vol. 18, p. 806–812. (Cited at page 25)

ELJIL, K. S., QADAH, G. and PASQUIER, M. (2013), «Predicting hypoglycemia in diabetic patients using data mining techniques», in «9th International Conference on Innovations in Information Technology (IIT)», vol. 23, p. 650 – 659. (Cited at page 13)

EREN-ORUKLU, M., CINAR, A., QUINN, L. and SMITH, D. (2009), «Estimation of the future glucose concentrations with subject specific recursive linear models», *Diabetes Technology & Therapeutics*, vol. 11 (4), p. 243–253. (Cited at page 12)

FABRIS, C., FACCHINETTI, A., SPARACINO, G., ZANON, M., GUERRA, S., MARAN, A. and COBELLI, C. (2014), «Glucose Variability Indices in Type 1 Diabetes: Parsimonious Set of Indices Revealed by Sparse Principal Component Analysis», *Diabetes Technology & Therapeutics*, vol. 16 (10). (Cited at page 25)

FACCHINETTI, A. (2016), «Continuous glucose monitoring sensors: past, present and future algorithmic challenges», *Sensors (Basel)*, vol. 16 (12), p. 825–833. (Cited at page 9)

FRIEDMAN, J. H., TIBSHIRANI, R. and HASTIE, T. (2016), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer,2nd edition.

GADALETA, M., FACCHINETTI, A., GRISAN, E. and ROSSI, M. (2019), «Prediction of Adverse Glycemic Events From Continuous Glucose Monitoring Signal», *IEEE Journal of Biomedical and Health Informatics*, vol. 23 (3), p. 650 – 659. (Cited at pages 13 e 25)

GANI, A., GRIBOK, A., RAJARAMAN, J. and REIFMAN, J. (2009), «Predicting subcutaneous glucose concentration in humans: Data-driven glucose modeling», *IEEE Trans Biomed Eng*, vol. 56 (2), p. 246–254. (Cited at page 12)

GEORGA, E., PROTOPAPPAS, V., ARDIGÒ, D., POLYZOS, D. and FOTIADIS, D. (2013), «A Glucose Model Based on Support Vector Regression for the Prediction of Hypoglycemic Events Under Free-Living Conditions», *Diabetes technology & therapeutics*, vol. 15 (8). (Cited at pages 12 e 13)

GEORGA, E. I., PROTOPAPPAS, V. C., POLYZOS, D. and FOTIADIS, D. I. (2015), «Evaluation of short-term predictors of glucose concentration in type 1 diabetes combining feature ranking with regression models», *Medical & Biological Engineering & Computing*, vol. 53 (12), p. 1305–1318. (Cited at page 12)

GILLIS, R., PALERM, C. C., ZISSER, H., JOVANOVIČ, L., SEBORG, D. E. and DOYLE, F. J. (2007), «Glucose Estimation and Prediction through Meal Responses Using

Ambulatory Subject Data for Advisory Mode Model Predictive Control», *Journal of Diabetes Science and Technology*, vol. 1 (6), p. 825–833. (Cited at page 11)

GRAVELING, A. and FRIER, B. (2017), «The risks of nocturnal hypoglycaemia in insulin-treated diabetes», *Diabetes Res Clin Pract*, vol. 133, p. 30–39. (Cited at page 12)

HILL, N. R., HINDMARSH, P. C., STEVENS, R. J., STRATTON, I. M., LEVY, J. C. and MATTHEWS, D. R. (2007), «A method for assessing quality of control from glucose profiles», *Diabetic Medicine*, vol. 24, p. 753–758. (Cited at page 28)

HINTON, G. E. and SALAKHUTDINOV, R. R. (2006), «Reducing the Dimensionality of Data with Neural Networks», *Science*, vol. 313. (Cited at page 91)

HSIEH, A. and TWIGG, S. (2014), «The enigma of the dead-in-bed syndrome: challenges in predicting and preventing this devastating complication of type 1 diabetes», *J. Diabetes Complications*, vol. 28 (5), p. 585–587. (Cited at page 12)

JENSEN, M. H., DETHLEFSEN, C., VESTERGAARD, P., HEJLESEN, O. and HEJL, O. (2019), «Prediction of Nocturnal Hypoglycemia From Continuous Glucose Monitoring Data in People With Type 1 Diabetes: A Proof-of-Concept Study», *Journal of Diabetes Science and Technology*. (Cited at page 12)

KILPATRICK, E., RIGBY, A., GOODE, K. and ATKIN, S. (2007), «Relating mean blood glucose and glucose variability to the risk of multiple episodes of hypoglycaemia in type 1 diabetes», *Diabetologia*, vol. 50 (12), p. 2553–2561. (Cited at page 25)

KOVATCHEV, B. P., COX, D. J., FARHY, L. S., STRAUME, M., GONDER-FREDERICK, L. and CLARKE, W. L. (2000a), «Episodes of Severe Hypoglycemia in Type 1 Diabetes Are Preceded and Followed within 48 Hours by Measurable Disturbances in Blood Glucose», *The Journal of Clinical Endocrinology & Metabolism*, vol. 85 (11), p. 4287–4292. (Cited at page 25)

KOVATCHEV, B. P., STRAUME, M., COX, D. J. and FARHY, L. S. (2000b), «Risk Analysis of Blood Glucose Data: A Quantitative Approach to Optimizing the Control of Insulin Dependent Diabetes», *Journal of Theoretical Medicine*, vol. 3, p. 1–10. (Cited at page 26)

KROPFF, J., FAVERO, S. D., PLACE, J., TOFFANIN, C., VISENTIN, R., MONARO, M. and ET AL, M. M. M. (2015), «AP@home consortium. 2 Month evening and night closedloop glucose control in patients with type 1 diabetes under free-living conditions: a randomised crossover trial», *Lancet Diabetes Endocrinol*, vol. 3, p. 939–47. (Cited at page 11)

LI, K., DANIELS, J., LIU, C., HERRERO, P. and GEORGIOU, P. (2019), «Convolutional Recurrent Neural Networks for Glucose Prediction», *IEEE Journal of Biomedical and Health Informatics*. (Cited at page 91)

MIYEON, J., YOU-BIN, L., SANG-MAN, J. and SUNG-MIN, P. (2017), «Prediction of Daytime Hypoglycemic Events Using Continuous Glucose Monitoring Data and Classification Technique», . (Cited at pages 12 e 25)

MOSQUERA-LOPEZ, C., DODIER, R., TYLER, N., RESALAT, N. and JACOBS, P. (2019), «Leveraging a Big Dataset to Develop a Recurrent Neural Network to Predict Adverse Glycemic Events in Type 1 Diabetes», *IEEE Journal of Biomedical and Health Informatics*. (Cited at page 91)

NATHAN, D. M. (2014), «The Diabetes Control and Complications Trial/Epidemiology of Diabetes Interventions and Complications Study at 30 Years: Overview», *Diabetes Care*, vol. 37 (1), p. 9–16.

NCD-RISC (2016), «Worldwide trends in diabetes since 1980: a pooled analysis of 751 population-based studies with 4*4 million participants», *Lancets*, vol. 33 (8), p. 811–31. (Cited at page 2)

PALERM, C. C. and BEQUETTE, B. W. (2007), «Hypoglycemia Detection and Prediction Using Continuous Glucose Monitoring—A Study on Hypoglycemic Clamp Data», *Journal of Diabetes Science and Technology*, vol. 1 (5), p. 3263–3273. (Cited at pages 12 e 25)

PÉREZ-GANDÍA, C., FACCHINETTI, A., SPARACINO, G., COBELLI, C., GÓMEZ, E., RIGLA, M., DE LEIVA, A. and HERNANDO, M. (2010), «Artificial Neural Network Algorithm for Online Glucose Prediction from Continuous Glucose Monitoring», *Diabetes Technology & Therapeutics*, vol. 12 (1). (Cited at pages 12 e 90)

REIFMAN, J., RAJARAMAN, S., GRIBOK, A. and WARD, W. (2007), «Predictive monitoring for improved management of glucose levels», *Journal of Diabetes Science and Technology*, vol. 1 (4), p. 478–486. (Cited at page 11)

RODBARD, D. (2009), «Interpretation of continuous glucose monitoring data: glycemic variability and quality of glycemic control», *Diabetes Technology & Therapeutics*, vol. 11 (S1). (Cited at pages 25 e 27)

RODBARD, D. (2012), «Hypo- and Hyperglycemia in Relation to the Mean, Standard Deviation, Coefficient of Variation, and Nature of the Glucose Distribution», *Diabetes Technology & Therapeutics*, vol. 14 (10). (Cited at page 25)

SAISHO, Y., TANAKA, C., TANAKA, K., ROBERTS, R., ABE, T., TANAKA, M., MEGURO, S., IRIE, J., KAWAI, T. and ITOH, H. (2014), «Relationships among different glycemic variability indices obtained by continuous glucose monitoring», *Primary Care Diabetes*, vol. 9. (Cited at page 25)

SARWAR, N., GAO, P., SESHASAI, S., GOBIN, R., KAPTOGE, S. and ANGELANTONIO, E. D. (2010), «Diabetes mellitus, fasting blood glucose concentration, and risk of vascular disease: a collaborative meta-analysis of 102 prospective studies», *Lancet*. (Cited at page 4)

SEO, W., LEE, Y.-B., LEE1, S., JIN, S.-M. and PARK, S.-M. (2019), «A machine-learning approach to predict postprandial hypoglycemia», *BMC Med Inform Decis Mak*, vol. 19 (210). (Cited at pages 13 e 25)

SERVICE, F. J., MOLNAR, G. D., ROSEVEAR, J. W., ACKERMAN, E., GATEWOOD, L. C. and TAYLOR, W. F. (1970), «Mean amplitude of glycemic excursions, a measure of diabetic instability», *Diabetes*, vol. 19 (9), p. 644–655. (Cited at page 26)

SEURING, T., ARCHANGELIDI, O. and SUHRCKE, M. (2015), «The economic costs of type 2 diabetes: A global systematic review», *PharmacoEconomics*, vol. 33 (8), p. 811–31. (Cited at page 2)

SHALEV-SHWARTZ, S. and BEN-DAVID, S. (2014), *Understanding Machine Learning: from theory to algorithms*, Cambridge University Press.

SPARACINO, G., ZANDERIGO, F., CORAZZA, S., MARAN, A., FACCHINETTI, A. and COBELLI, C. (2007), «Glucose concentration can be predicted ahead in time from continuous glucose monitoring sensor time-series», *IEEE Transactions on Biomedical Engineering*, vol. 54 (5), p. 931–7. (Cited at pages 11 e 12)

SUN, Q., JANKOVIC, M. V., BALLY, L. and MOUGIAKAKOU, S. G. (2018), «Predicting Blood Glucose with an LSTM and Bi-LSTM Based Deep Neural Network», in «14th Symposium on Neural Networks and Applications (NEUREL)», . (Cited at page 91)

WILSON, D., CALHOUN, P. and ET AL, D. M. (2015), «Factors Associated with nocturnal hypoglycemia in at-risk adolescents and young adults with type 1 diabetes», *Diabetes Technol Ther [Internet]*, vol. 17 (6), p. 385–391. (Cited at page 13)

YANG, J., LI, L., SHI, Y. and XIE, X. (2018), «An ARIMA Model With Adaptive Orders for Predicting Blood Glucose Concentrations and Hypoglycemia», *IEEE Journal of Biomedical and Health Informatics*, vol. 23 (3), p. 1251 – 1260. (Cited at page 12)

ZECCHIN, C., FACCHINETTI, A., SPARACINO, G. and COBELLI, C. (2015), «Jump neural network for online short-time prediction of blood glucose from continuous monitoring sensors and meal information», *Methods in molecular biology (Clifton, N.J.)*, vol. 1260, p. 245–59. (Cited at page 90)

ZECCHIN, C., FACCHINETTI, A., SPARACINO, G. and COBELLI, C. (2016), «How Much Is Short-Term Glucose Prediction in Type 1 Diabetes Improved by Adding Insulin Delivery and Meal Content Information to CGM Data? A Proof-of-Concept Study», *Journal of Diabetes Science and Technology*, vol. 10 (5), p. 1149–1160. (Cited at page 13)

ZHONG, A., CHOUDHARY, P., MCMAHON, C., AGRAWAL, P., WELSH, J., CORDERO, T. and KAUFMAN, F. (2016), «Effectiveness of automated insulin management features of the MiniMed(®) 640G sensor-augmented insulin pump», *Diabetes Technology & Therapeutics*, vol. 18, p. 657–63. (Cited at page 11)

Websites consulted

- World Health Organization. Diabetes, Fact Sheets and Global Report – `www.who.int/health-topics/diabetes` `www.who.int/news-room/fact-sheets/detail/diabetes` `www.who.int/diabetes/global-report/en/`
- Type-1 Diabetes – `www.diabetes.org/diabetes-basics/type-1/`
- Scikit-Learn, Machine Learning in Python – `www.scikit-learn.org/stable/`
- XGBoost Documentation – `https://xgboost.readthedocs.io/en/latest/`
- Dexcom – `www.dexcom.com/about-dexcom`
- ScienceDirect – `www.sciencedirect.com`
- SpringerLink – `www.springerlink.com`
- National Center for Biotechnology Information, PubMed – `www.ncbi.nlm.nih.gov/pubmed/`
- Wikipedia – `www.wikipedia.org`

Acknowledgement

I would first like to thank my thesis advisor Prof. Andrea Facchinetti who allows the development of my work and was always available to help me. I would like to thank also my co-advisors Prof. Giovanni Sparacino and Eng. Giacomo Cappon for the very valuable comments and contributes to the thesis. Therefore I would like to thank all the team involved in the evaluation of this research project that gave me a lot of interesting suggestions and advice.
Finally, I must express my gratitude to my parents Beatrice and Giulio and to my partner Marco for providing me with unfailing support and continuous encouragement throughout my years of study and through the development of this thesis. Another big thank to my sister Isabella and to all my friends that are always present in the good, but also in the bad, days.
This accomplishment would not have been possible without them. Thank you.