Detection of Myocardial Infarction in Electrocardiograms using Machine Learning

Paulo Vinicius Masnik
Federal Institute of Santa Catarina (IFSC)
950, Av. Mauro Ramos
Florianópolis – SC, Brazil

Roberto Alexandre Dias
Federal Institute of Santa Catarina (IFSC)
950, Av. Mauro Ramos
Florianópolis – SC, Brazil

Mario de Noronha Neto
Federal Institute of Santa Catarina (IFSC)
608, José Lino Kretzer Street
São José – SC, Brazil

ABSTRACT

Currently, millions of people in the world have some type of deficiency in the cardiovascular system, more specifically anomalies in the heart and its heartbeat, most of these individuals end up not discovering these problems in advance, which would have a great impact on the chance of survival. In Brazil, the number of deaths caused by heart problems exceeds 350 thousand per year. The solution found to assist in the prevention and detection of pre-existing problems starts from the approach of analyzing electrocardiograms of people with already known conditions and anomalies, starting from the machine learning method for preventing conditions with only data input to a model. The proposal of this work designs in a prototype in which, in just a few moments, it generates a prediction with a considerable success rate, capable of assisting health professionals to make decisions regarding the patient's situation, based on the analysis of waves from an electrocardiogram (ECG). During this work, it is demonstrated the entire process of data acquisition and selection, treatment and filtering of wave signals until the development of an exam prediction. The results found were correct rates in the infarction class, higher than 80, 90 and up to 95%. It is also important to understand that the increase in the hit rate of the class with the anomaly tends to decrease the hit rate of normal exams.

General Terms

Electrocardiograms, data science, data analytics, machine learning and cardiovascular diseases.

Keywords

Myocardial detection, electrocardiograms analysis, machine learning in healthcare.

1. INTRODUCTION

The introduction of the term machine learning occurred in the 1950s by Arthur Samuel, who was then an engineer at Massachusetts Institute of Technology. This technology has its origin linked to the area of robotics related to artificial intelligence, where its focus is characterized by the construction of algorithms and technologies that are able to "learn" with a certain amount of data and finally "predict" certain behaviors and patterns complex for a human being.

The algorithms are developed through the analysis of a wide range of data used for interpretation and validation by the system and as this occurs, patterns are defined and found through a unique learning of the operation itself. Thus, the algorithms are able to determine reasons and classify elements as well as to determine percentages related to the desired classification. In the healthcare area, artificial intelligence and machine learning techniques are being increasingly supported and used in hospitals and clinics, and it is becoming simpler to develop specific machine learning models to address problems in individual areas in the health field and to indicate different types of diagnoses. However, it should be noted that for certain applications where a generic standard is sought, a machine learning model does not need to be assigned to just a specific prediction. The advantages found are wide, highlighting the agility of a diagnostic and the diagnostic error tends to decrease considerably [1]. Another competitive advantage of using artificial intelligence in the health area would be to be able to make a diagnosis without the need for a doctor, or even, without needing to be present on the spot, such as in rural, riverside and inland areas, where there is a certain difficulty in accessing the internet and there isn't a doctor available to perform the exam [2]. A current example where the use not only of machine learning but also of artificial intelligence can be displayed is the detection of 26 types of skin conditions, a study developed by Google AI [3], where it was found that the learning model achieved 90% accuracy in diagnosis, while dermatologists achieved 75% and nursing practitioners who achieved 55% accuracy.

In the field of cardiology, several studies and articles carried out recently have highlighted the importance of using machine learning algorithms [4].

2. PROBLEM DEFINITION

The problem to be addressed in this work is to perform the analysis of electrocardiograms signals using a data science approach, transforming the wave signals (from a data source on the internet) from the graphic format to vector formats containing the 12 exam leads (stored in tables), for the detection exclusively of myocardial infarction using the Machine Learning technique. The problem itself includes: analysis of the set of exams, pre-processing of the data (vital stage of the treatment, because the higher the quality of the data, the greater the chance of having a quality model), creation of models, parameterization, optimization, training of ECG data and subsequently the post-processing of the data. The algorithm must also generate indicators with the individual probability of myocardial infarction for each patient analyzed, as well as allowing the insertion of new data and making their prediction.

Seeking to clarify the tasks to be performed and the purpose of the work, a general objective was defined, represented by a main task to be carried out and also some specific objectives, mainly secondary tasks, but that helped in the better accomplishment of the work itself.

2.1 General Objective

Conduct a study and prediction of myocardial infarction in electrocardiograms, classifying the electrocardiograms in common exams and exams with incidence of myocardial infarction. This objective comprises the selection of the machine learning algorithm for the analysis and prediction of electrocardiograms, as well as the interpretation of the wave signals of the exam in order to develop a prototype system to indicate the possibility of myocardial infarction in the patient and make it available on the platform what was the prediction.

3. ELECTROCARDIOGRAM FUNDAMENTALS

The electrocardiogram is one of the most used exams around the world nowadays for heart analysis, being used in all spheres of health, from health and screening centers, clinics and offices to hospitals and specialized centers. Because it is a relatively accessible test to be applied, the ECG serves as a kind of screening for patients in care, in addition to making the service much more democratic. However, the exam is still totally dependent on the analysis of the images by a specialized professional. An electrocardiogram must be analyzed by a cardiologist, general practitioner or a doctor who works in urgencies, and they must have the ability to find and diagnose the pathophysiological principles of electrocardiographic abnormalities found in the exam [5].

The advantages of an electrocardiogram are wide, mainly because it is an exam that demonstrates innumerable problems in the heart, it is non-invasive and has a fast performance (mainly compared to the holter exam, a kind of long-term electrocardiogram for detecting mainly arrhythmias) and it can also be easily used in large studies. The common electrocardiogram is composed of 12 vector formats that refer to different visualizations of the electrical activity occurring in the heart muscle, these settings being the result of calculating the difference in electrical potential of a negative electrode and a positive electrode. Electrodes are connectors that connect the patient's skin to the instrument that performs the examination, through the electrical conductivity they will play the role of the carrier of the heart activity signal to the performing instrument and store the data. These electrodes are then placed on the chest and on some parts of the body of the patient undergoing the examination, depending on which lead is to be measured. The views can be divided into some types: vertical (frontal derivations of the type D1, D2 and D3 and of the members of the type: aVR, aVL and aVF) and horizontal (precordial derivations of the type: V1, V2, V3, V4, V5 and V6).

3.1 Derivations

The derivations are described in two major classes: precordial and peripheral, both of which are divided into subclasses. In relation to the peripheral ones, they can be classified in bipolar and increased unipolar. As described by Willem Einthoven in 1913 [6], those of the bipolar format (three in all) are the derivations that generate the Einthoven Triangle and have the function of measuring the voltage relative to the difference of two electrodes (one positive pole and another negative) that are positioned on parts of the body. This tension is the representation of the momentary activity of the heart and is the means by which the analysis of the behavior and the state of the muscle is performed. It is also worth noting that the derivations themselves represent the different vectors to be observed in order to understand the exam.

The increased unipolar derivations [7], in turn, start to detect the absolute electrical potential, being this, the capacity of an energized body has to perform work or more simply to attract or repel other electrical charges. This comparison is made between a central region of the Einthoven Triangle and its extremes.

The precordial leads, the region in front of the heart and being six in all, complement the 12 leads common to an ECG, they are also described as the absolute electrical potential. These derivations are extracted from regions close to the chest, the positioning of which is described below:

3.2 The myocardial infarction

For the study presented in this work, as well as for correspondence with the data set used for the validation of the project, it was decided to address one of the main and most common cardiovascular problems presented by people when seeking care, the myocardial infarction. This problem is related to the deterioration of cells in a region of the heart. The term infarction refers to cell necrosis, which can be caused in any region of the body, and is caused by a blood clot that stops the blood flow suddenly. The main reason for the occurrence of myocardial infarction is when fatty plaques (usually caused by high cholesterol) cluster in one of the two coronary arteries (arteries that vascularize and nourish the entire heart), causing their obstruction. The condition that precedes the infarction is called ischemia, which is when the tissue itself is already receiving less blood than necessary. Most of the time, the infarction occurs when these plaques rupture, forming clots and interrupting the blood flow, as previously described, which will cause a decrease in the oxygenation of cells in the myocardium [8].

4. PROPOSED SOLUTION

In the view of a need for an specific model to predict anomalies, where in the future the same model can train and predict other comorbidities selected by the user, by simply informing which anomaly to be predicted, the model should be quick and able to assist the doctor to a more accurate detection of the possible deficiencies present in the examination, therefore, the proposed flow of solution is shown in Figure 1. The possibility of the software working together with human service is capable of providing a democratization of the model for more remote areas, as well as help in the reduction of diagnostic errors that are common to occur, even more if analyzed in a big sampling like the number of electrocardiograms done in Brazil. Continuous data feedback, allowing the model to readjust its parameters over a time interval defined by users, will allow the data to be always adapted to reality and updated.

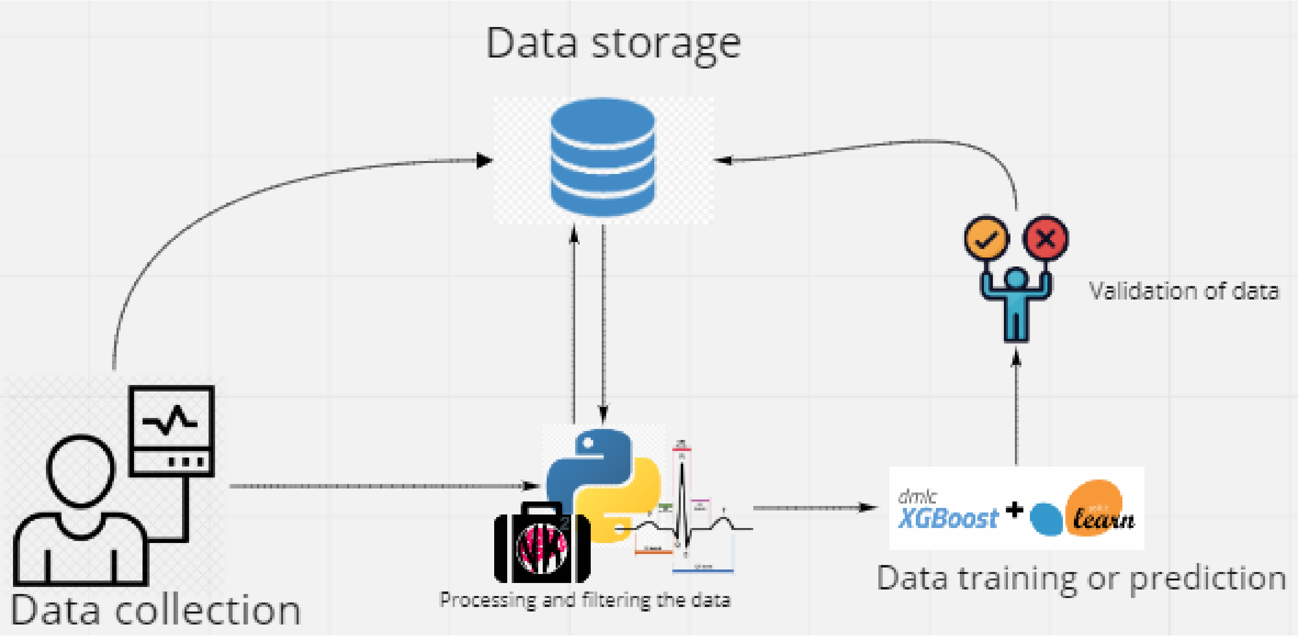


Fig 1: The system flow of the designed study

As mentioned, the data set used has 21837 samples from different people, therefore, the vector used for the data processing will always have the format (n, 1000, 12), being *n* the number of exams extracted from the data sample.

4.1 Data filtering

The treatment of the waves of each exam lead is an essential part of the data filtering and treatment process, since the electrocardiogram signals are signals of low amplitude and sensitive to the person's own muscular activity and at a frequency of 60 Hz. The best way to reduce these interferences is to take great care when making the signal acquisition, especially when preparing the electrodes and using appropriate wiring. In certain situations, noise will be a common and present element, therefore the removal of noise from a signal means to soften the interference caused by an element, which will result in an increase the signal-noise.

The solution for removing this noise is attributed to the digital processing of the signal with noise after it is acquired, however it must be considered that the ECG filtering models act not only on the noise, but also on the ECG signal itself. The best option is one where, after filtering the ECG signals, there is a minimum of distortion of the original exam signal. The simplest techniques are based on finite impulse filters and infinite impulse filters, while more complex techniques are based on Artificial Neural Networks and Wavelet Transform. To carry out the filtering of the signals, the neurokit neuropsychological signal processing library was used [9]. Neurokit consists of a usability package for analysis of clinical data such as photoplethysmography, breathing data and, in the case of the project, electrocardiograms.

To perform the data filtering, the function ecg_clean function available in the library is the main option to use, this function consists of identifying the R wave peaks of each moment of the electrocardiogram.

The R wave corresponds to depolarization of the ventricular walls, that is, myocardial cells are incited, there is a discharge of the electrical forces stored in these cells, that discharge will then depolarize and contract, causing the muscle to contract.

The Neurokit algorithm is able to accurately detect the peak moments of the R wave, which is the instant when the heart contracted, it is also allowed to choose between some different methods of detecting the R peak, such as the Pan-Tompkins method [10], which was used in the project.

The Pan-Tompkins method is widely used in current literature as a benchmark to evaluate the performance of detectors in the QRS complex [11]. The performance of the method is mainly divided into two stages: digital filtering and decision rules. Initially in the filter application process, the signal from a lead of the electrocardiogram passes through a bandpass filter with frequencies between 5 Hz and 15 Hz, these bands are chosen to reduce the influence of external noise sources. In order to obtain greater prominence at the extreme points (maximum and minimum), the signal will be derived, raised to the square and thus integrated with a moving average value, this phase will indicate some possible values such as the R wave peaks. With the possible peaks found, the algorithm then moves on to the second treatment step: the decision rules. The Pan-Tompkins algorithm defines two limit points (thresholds), the upper limit point refers to 50% of the maximum peak found in every filtered signal, while the lower limit point will be 50% of the upper limit value.

With the definition of these two limits, it is possible to use the analysis to determine the distance between two peaks R, respecting an interval of 20% of the signal frequency. According to human physiology, the shortest distance considered between two R peaks is 200 ms, Pan-Tompkins defined that each interval between the peaks can be extracted from the average of the following eight R peaks. With the R peaks located, the algorithm can detect the values referring to waves Q and S, inferring that the location of Q will be 500 ms before peak R and S will be 500 ms after peak R. With this detection, the algorithm can filter the electrocardiogram signal and successfully detect the QRS complex. The Figure 2 shows us how the method detected the R peaks in the data analyzed. As the ecg_clean function described is able to identify approximate values of each segment, wave and interval for each beat measured on the electrocardiogram, noise smoothing is possible as shown in the Figure 3, an individual beat (“Individual Heart Beats”) before signal cleaning, where some instances of waves are generated according to the method and an average value between them is considered the processed value. It can also be seen the original signal in gray, the filtered signal in pink and the peaks of the R wave represented by yellow dots.

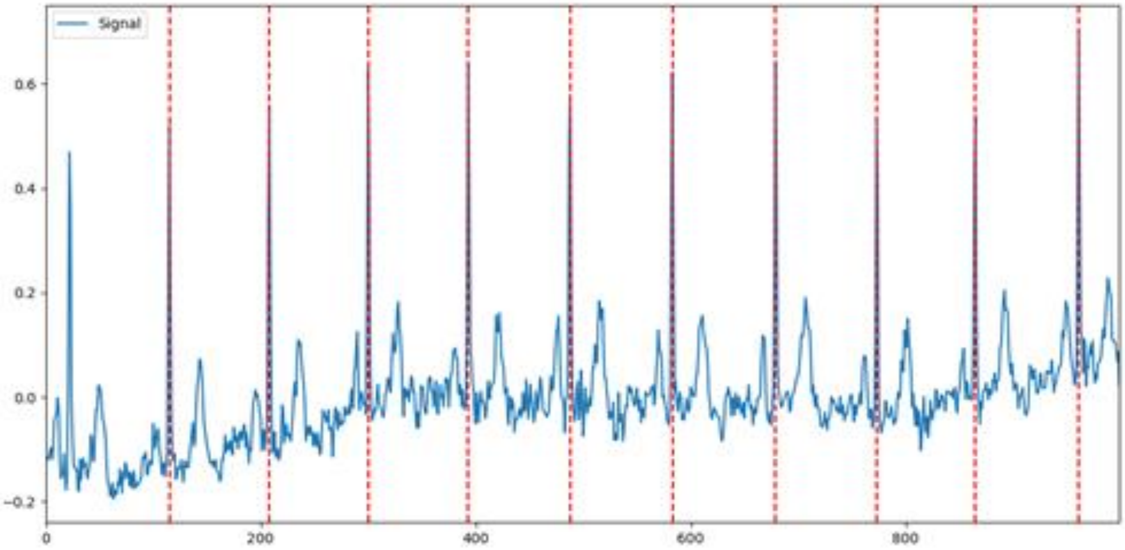


Fig 2: Detection of the R peaks in the data collection

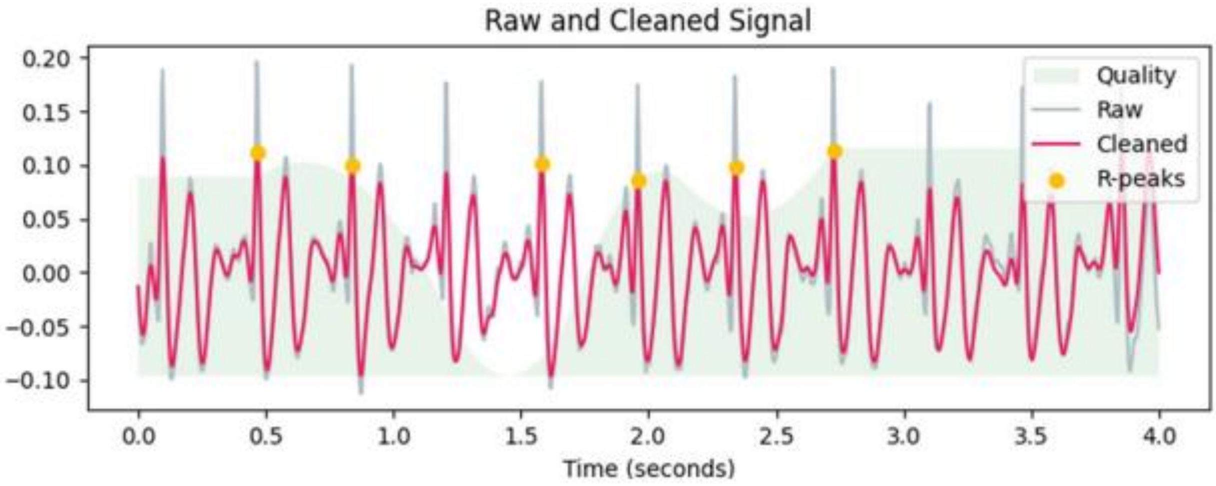


Fig 3: The Raw and cleaned signal of the study data using the Neurokit library

4.2 Appling the data to a Machine Learning Model

The choice of a reliable and efficient algorithm is the starting point for the construction of the desired model in the project, because of this, some points were raised for the decision to use the machine learning framework XGBoost [12]. XGBoost is a framework that uses the gradient boosting technique, to determine the best parameters and error metrics for the model. This technique is used in regression and classification problems, and the boosting is the combination of several models considered weak (with an accuracy below the expected), these models being generally in the shape of a decision tree. Therefore, each weak classifier is trained with a set of data, sequentially and in an adaptive manner, where a base model will always depend on the previous ones, and in the end they are combined in a deterministic way. It is noteworthy that the boosting technique differs from Bagging algorithms (another traditional algorithm), where weak models are trained individually and in parallel, being subsequently combined following a process average, as a kind of voting for the best results.

Thus, when opting for the gradient boosting technique and the dependence of the models on its previous one, combining this with one or more arbitrary error metrics, the objective then is to create a chain of non-ideal models, where the objective is to minimize the error of the previous model, using a loss function. The beginning of the technique consists of generating an initial model, using a simple approximation to the real values, which will return us a residue, being the difference between what was predicted and the real values. This residue will then be used for a prediction by a second fragile model, which will generate a new residue and this sequence will be persisted a number of times specified by the user. That is, the minimum difference between the predicted value and the actual value. The final model will be the sum of all adjustments of all weak models.

The training and interpretation of data by the framework used is, perhaps, the primordial step in every data processing flow, since it will define the fixed values of all parameters that will be called in every forecast that a new data is exposed. That is, as long as there is no call for re-training the model, the same file with the best parameters generated by the training will be used in the prediction of any new data entered in the system.

The initial approach after loading the processed data is to use the Label Encoding technique [13], this technique assumes that multiple machine learning algorithms cannot operate directly with alphanumeric values, being required that all variables entered are numeric only. This designation serves more as an implementation to increase the efficiency of the models than a limitation of them in itself, however, if an attempt to predict data sets with alphanumeric values is made, the algorithm will not accept such an imposition. Thus, the application of the Label Encoding technique will convert all alphanumeric columns of the data to a numerical representation. As the vectors extracted from the electrocardiogram waves are already fractional numerical variables, the only column that will make this approach necessary is the response column, containing the diagnosis of each electrocardiogram. This treatment where the numerical values were applied to the column referring to the diagnosis: 0 for normal and 1 for myocardial infarction.

Another important technique used was the normalization of the values of the numerical columns, in the work performed, the MinMaxScaler technique [14] was used in the sklearn library [15]. The main idea behind this approach is that different variables, because they have specific scales, end up not contributing equally to the model. Thus, when using MinMaxScaler, each column will be treated and normalized to a range from 0 to 1, that is, the minimum value found in the column will be set to zero, as well as, the maximum value found in the column will be transformed to 1 and the values within the column.

After all data formatting, a separation of the total data between training data (used only to feed the training model) and test data is used, these data will act as a kind of new insertion, that is, this portion of data never seen by model training, will be predicted and based on evaluation metrics, the model's reliability and accuracy rates can be . The recommendations for creating the models are always to use 70% to 90% of the data for training the model, and the rest, for the test data, which will guarantee how accurate the model developed is. The column with the answers is also separated from the rest of the data columns. The division between training and test occurs through the specification of the data used, following the recommendation of Physionet (the separation was made in this way), therefore the relationship between training and test was 88, 8%. With the data processed and separated between training and testing, data training was carried out. For this step, an empty XGBoost model is called and defined with which parameters and their intervals will be tested by the model. This step of selecting the parameter range is essential for the tuning technique, used to find the best model.

4.3 Parameter tuning

An important step in the development and improvement of a machine learning model is the correct definition of the best available parameters. The combination of parameters using values at specific intervals can be fundamental for the definition of an excellent model for a poor quality model, since most of the parameters have considerable sensitivity. Tuning can be done in a few different ways. The technique can be performed manually, where the developer manually changes the model parameters and runs the algorithm several times until he finds the best scenario, or, a test algorithm can be used, to optimize and automate that scenario.

There are two main tuning algorithms: GridSearchCV [16] where the algorithm will test all possible combinations of the values and RandomSearchCV [17],which was used in the work, mainly because it suppresses the problem of multiple combinations and processing time, its working methodology is to perform random combinations through a number of predefined iterations in development, so the best results serve as a guide for choosing hyper parameters.

In conjunction with RandomSearchCV, in order to determine which reduced intervals would be more assertive in relation to broader intervals, that is, extreme values were first tested for each parameter and based on the response of the best model found, a filter itself. An example of this is the learning_rate parameter, which by definition receives values in the range 0 to 1, so initially it is applied values in RandomSearchCV close to the extreme 0, intermediate values like 0.5 and values close to the extreme 1.

Having determined which of the main values has the best initial result, it can specify the search for values close to the best initial, if the initial value of the best result was 0, it can specify the search in close values such as 0.05, 0.1 or 0.15 and if it is the intention, you can refine this filter again in order to limit the search for RandomSearchCV in the interval that verified the best result values. To understand how RandomSearchCV takes into account the evaluation of the model based on the smallest error found, the concept of cross validation and the error metric used by roc_auc_ovr must be better understood. A common problem in data science is the problem of variance, where the accuracy of the model in one test data can be totally different from the accuracy obtained in another test data, using the same model generated by the algorithm. The solution to this problem is to use a Cross Validation algorithm [18], this technique aims to fragment the training data in K plots, with K being the variable initially defined as the number of fragmentations that the model will accomplish.

The version of cross validation used in the work is the StratifiedKFold [19], having as a differential of the other cross validation algorithms the fact of maintaining in all the fragmentations of the training data, the original relationship between the majority and minority class in all divisions. This will ensure an equal balance between all divisions of the data, ensuring homogeneous training throughout the process. To measure the results of the fragmentation that represents the test in each of the training iterations performed, it was used as scoring, a parameter referring to the strategy for evaluating the cross validation fragmentation test performance, it was the roc_auc_score, this metric concerns the ROC curve [20].

A ROC curve traces the relationship between the rate of true positives vs. the rate of false positives at different thresholds of classification.

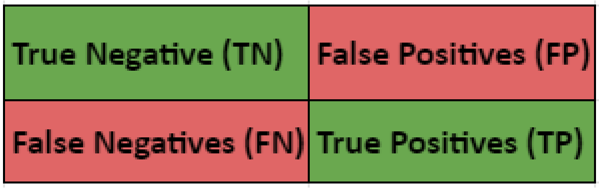
Thus, in an attempt to simplify the analysis of the ROC curve, the metric AUC (“area under the ROC curve”) emerged, which is nothing more than the way to summarize the ROC curve in a single value, aggregating all ROC entries, calculating then the “area under the curve”. The value of the AUC ranges from 0.0 to 1.0 and the divisor between the class is 0.5. Thus, any value above this limit, the algorithm classifies as one class and below the other. The value of AUC is directly proportional to the correctness rate of the model, that is, the higher the value of AUC, the greater the accuracy of the algorithm.

4.4 Training the model

The training itself is performed using the XGBoost fit command, in this step, the training data are passed as lists, which will be used as the basis for designating the relationships. As iterations are performed during the tuning technique and RandomSearch, these iterations will be grouped with the rounds of iterations of the fit call, therefore, if 50 predefined iterations are defined for random selection of parameters according to the tuning technique, the model will be trained 50 times and in those 50 times, it will only stop its training until a defined number of consecutive iterations show no improvement in the evaluation of the error metric. At the end of the training, the result list will store the training information, and the best model found will be accessible, assigning it to the variable containing its values.

5. RESULTS

The best model found in the training process is then used to perform the validation of predictions, which through command .predict (variable where the best model is stored), there was the prediction of all file called X_test, containing all vector information of the 12 leads of the patients' exams, however, does not contain the answer (in the case if the exam was considered normal or with myocardial infarction). Finally, the values of each patient can be saved by creating two columns in a dataset, which is referenced through the patient's unique id. The continuous refinement of the parameters must be carried out based on the metrics used to find the best hit rates for the model. Therefore, two main metrics were used as a basis for analyzing the model's quality: confusion matrix[21] and the AUC value ("area under the ROC curve"), which is the format used to summarize the ROC curve in a single value. The confusion matrix is one of the most used metrics in the current classification models, being one of the most used terms in the Machine Learning area today and having its origin attributed to the term contingency table. Figure 4 represents one of the ways to visualize it.



True Negative (TN)	False Positives (FP)
False Negatives (FN)	True Positives (TP)

Fig 4: The confusion matrix

This matrix allows for a much-discussed approach when developing a learning model: the exchange between bias and variance. This exchange corresponds to the dilemma of adjusting the parameters of the model to seek an increase in the number of correct positives, however, this tends to cost in an increase in the number of false positives. Thus, some factors must be taken into account when adjusting the model's correctness, for example: When analyzing a fictitious data set with 1000 samples, of which only 5 are from really sick people, which would be more relevant : Do you correctly predict these 5 sick people but detect 5 more of them as false positives, or not detect any false positives but only detect 3 sick people?

Likewise, when analyzing a dataset of 1000 samples where there is an attempt to predict whether an animal is a dog or a cat, there is no distinct importance between the classes, thus, using the model that has the better accuracy rate. In the work developed, it was sought to find a model in which the test data respected values above 80% both in the calculation of sensitivity and specificity. An initial analysis was made with some different error validations, as shown in Table 1.

Table 1 – Confusion Matrix of initial parameters

Approach	Sample Size	Error used	Confusion Matrix
Initial parameters	1000	roc_auc_score	57 5 12 12
Initial parameters	1000	precision	57 5 13 11
Initial parameters	1000	f1_score	59 3 11 13

With a prediction of the myocardial infarction class close to 50% for all techniques initially analyzed, some attempts were made to refine the data in order to improve the final hit percentages. The approach then taken was to use the signal filter, as described previously. With the filtered data, the model's threshold [22] was also observed. The threshold is the decision threshold value on the ROC curve, that is, values below this number will be classified as one class and values above this value will be classified as the other class. The common threshold value is 0.5, associated with the 50% value of a probabilistic range of 0-100%, however, contrary to the empirical sense, the threshold is a problem-dependent. Therefore, depending on the type of model created for a data set, better results can be achieved using different values. For example, a value of 0.4 instead of the common 0.5. To analyze the best models found, some samples of the scale_pos_weight parameter were used, which is the weight between the majority class and the minority class of data. Tables 2 and 3 show several tests of the confusion matrices and their correctness rates for the best results found, for different threshold values. The weights used between classes for each of the tables were 1.25 and 0.8 respectively and all of them used a sample size of 3000 samples and the roc_auc_score.

Table 2 – Values for an weight of 1.25

Approach	Confusion Matrix		Hit Rate
0.5 threshold	179	17	91.33%
	29	41	58.57%
0.3 threshold	152	44	77.55%
	15	55	78.57%
0.15 threshold	104	92	53.06%
	5	65	92.86%

Table 3 – Values for an weight of 0.8

Approach	Confusion Matrix		Hit Rate
0.5 threshold	151	16	90.42%
	15	40	72.73%
0.3 threshold	128	39	76.65%
	6	49	89.09%
0.15 threshold	86	81	51.5%
	2	53	96.36%

Using a sample of 10,000 tests, a high rate of accuracy can be found in the test data, which has never been seen by the model, above 80% in normal exams and also above 80% in exams with incidence of heart attack. myocardium, however, understanding that in the health area, the impact of false negatives is much more serious than the incidence of false positives, the model also obtained results close to 90% of correctness in the myocardial infarction class, using other threshold values as shown in Table 4.

The results, according to Table 5, show an accuracy (general hit rate of the model) of 0.823 (82.3%) with a true positive rate (total hit of the class with incidence of myocardial infarction) of 0, 84 (84%). The model also had a reasonable false positive rate of 0.185 (18.5%). The general precision of the model counting false positives was 0.687 (68.7%) while the metric f1, a combination of precision and the rate of true positives, which indicates the general quality of the model, of 0.756 (75.6%).

Table 4 – Final metrics of the model

Metric	Value
True positive rate	0.84
True negative rate	0.185
Precision	0.6865
Accuracy	0.823
F1-Score	0.756

6. CONCLUSION

During the work, the steps taken in the development of a myocardial infarction prediction model in 12-lead electrocardiograms were presented. From the theoretical detailing of the technologies used to their more technical specifications, it was opted to use of open source solutions that are widely used in projects today. In relation to the model developed, it was possible to obtain a hit rate greater than 80% both in the prediction of normal exams and in the prediction of the class in which myocardial infarction was present, and in some cases, hit rates higher than 75% of normal exams and almost 90% in relation to exams with myocardial infarction, considering that in the health area there is a tendency to prefer a greater number of false positives if this implies in a few diagnostic errors in the disease (false negatives).

A weighting factor in the data was also the use of random selection of parameters (the RandomSearchCV technique), within predefined intervals. This technique of RandomSearchCV was used for code optimization reasons. However, it does not guarantee exactly the best global parameters. The approach of using confusion matrices to detect the best threshold is an interesting approach for the sake of visualizing the results and how the model advances or reduces as the threshold value is increased or decreased.

The general values of the model and the results found satisfy the research developed, where you can find hit rates in the infarction class, higher than 80, 90 and up to 95%, also understanding that the increase in the hit rate of the class with the condition, it tends to decrease the correct rate of normal exams.

6.1 Future works

Future work for this research mainly includes increasing the granularity of the parameters and also looking for better hit rates. Another future project would be the use of the same system process for other comorbidities, as well as the integration of the project created with smartwatches, which with technological advances are already able to perform electrocardiogram exams.

7. REFERENCES

[1] Richens, J.G., Lee, C.M. and Johri, S., 2020. Improving the accuracy of medical diagnosis with causal machine learning. *Nature communications*, *11*(1), pp.1-9.

[2] Lehoux, P., Sicotte, C., Denis, J.L., Berg, M. and Lacroix, A., 2002. The theory of use behind

telemedicine: how compatible with physicians' clinical routines?. *Social science & medicine*, *54*(6), pp.889-904.

[3] Liu, Y., Jain, A., Eng, C., Way, D.H., Lee, K., Bui, P., Kanada, K., de Oliveira Marinho, G., Gallegos, J., Gabriele, S. and Gupta, V., 2020. A deep learning system for differential diagnosis of skin diseases. *Nature Medicine*, *26*(6), pp.900-908.

[4] Li, Q., Rajagopalan, C. and Clifford, G.D., 2014. A machine learning approach to multi-level ECG signal quality classification. *Computer methods and programs in biomedicine*, *117*(3), pp.435-447.

[5] Nicolau, J.C., Polanczyk, C.A., Pinho, J.A., Bacellar, M.S.D.C., Ribeiro, D.G.L., Darwich, R.N., Ribeiro, A.L.P., Dunda, M.M.E., Germiniani, H., França, F.F. and Saraiva, L., 2003. Diretriz de interpretação de eletrocardiograma de repouso. *Arquivos Brasileiros de Cardiologia*, *80*, pp.1-18.

[6] Barold, S.S., 2003. Willem Einthoven and the birth of clinical electrocardiography a hundred years ago. *Cardiac electrophysiology review*, *7*(1), pp.99-104.

[7] Neto, M.M.R., 1948. Conceito e valor das derivações unipolares dos membros. *Revista de Medicina*, *32*(169-172), pp.55-73.

[8] White, H.D. and Chew, D.P., 2008. Acute myocardial infarction. *The Lancet*, *372*(9638), pp.570-584.

[9] Makowski, D., Pham, T., Lau, Z.J., Brammer, J.C., Lespinasse, F., Pham, H., Schölzel, C. and Chen, S.A., 2021. NeuroKit2: A Python toolbox for neurophysiological signal processing. *Behavior Research Methods*, pp.1-8.

[10] Pan, J. and Tompkins, W.J., 1985. A real-time QRS detection algorithm. *IEEE transactions on biomedical engineering*, (3), pp.230-236.

[11] Álvarez, R.A., Penín, A.J.M. and Sobrino, X.A.V., 2013. A comparison of three QRS detection algorithms over a public database. *Procedia Technology*, *9*, pp.1159-1165.

[12] XGBOOST: XGBoost Documentation. [S. l.], 2021. Available at: <https://xgboost.readthedocs.io/en/latest/>. Acessed in: 16 de mar. de 2021.

[13] LABELENCODER: Encode target labels with value between 0 and n_classes-1. [S. l.], 2021. Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html#sklearn.preprocessing.LabelEncoder>. Acessed in: 16 de mar. de 2021.

[14] MINMAXSCALER: Transform features by scaling each feature to a given range. [S. l.], 2021. Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html>. Acessed at: 15 de mar. de 2021.

[15] SCIKIT-LEARN: Simple and efficient tools for predictive data analysis. [S. l.], 2021. Available at: <https://scikit-learn.org/stable/index.html>. Acessed at: 15 de mar. de 2021.

[16] GRIDSEARCHCV: Exhaustive search over specified parameter values for an estimator. [S. l.], 2021. Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html>. Acessed in: 15 de mar. de 2021.

[17] RANDOMIZEDSEARCHCV: Randomized search on hyper parameters. [S. l.], 2021. Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html>. Acessed in: 15 de mar. de 2021.

[18] Shaikh, R., 2018. Cross Validation Explained: Evaluating estimator performance. Towards Data Science. Available at: <https://towardsdatascience.com/cross-validation-explained-evaluating-estimator-performance-e51e5430ff85> Acessed at: 16 de mar. de 2021.

[19] STRATIFIEDKFOLD: Stratified K-Folds cross-validator. [S. l.], 2021. Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html>. Acessed at: 15 de mar. de 2021.

[20] Bradley, A.P., 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern recognition*, *30*(7), pp.1145-1159.

[21] Provost, F. and Kohavi, R., 1998. Glossary of terms. *Journal of Machine Learning*, *30*(2-3), pp.271-274.

[22] GOOGLE DEVELOPERS. Classification: Thresholding. [S.I]. 2020. Available at: <https://developers.google.com/machine-learning/crash-course/classification/thresholding> Acessed at: 23 de mar. 2021.