Binary Classification: Comparative Study of Machine Learning Algorithms in Pima Indians Dataset

Guilherme Michel Lima de Carvalho [*], Jaqueline Lopes Dias[+] and Marcos Jardel Henriques[@]

Abstract— **This work proposes, through machine learning techniques, to analyze a diabetes database: Pima Indians. This database is composed of data that brings results from a group of Native American women who live in Arizona, where research was carried out on the main factors related to diabetes. Thus, in this article this data set was analyzed using the techniques: Naive Bayes, Logistic Regression, Support Vector Machine, K Nearest Neighbors, Decision Trees, Random Forest, Perceptron and Multilayer Perceptron. The missing data were treated using regression techniques. In addition, to deal with unbalanced data, the SMOTE technique was used. The results obtained by these techniques were compared using the following criteria: accuracy, precision, recall and F1 score. Defining the selected models, the best models were KNN, with 3 closest neighbors and Minkonski's distance, (precision of 89%), random forest with number of estimators 600 and Gini method (determination of 93%) and MLP with 1000 hidden layers of 1000 neurons and with activation function (precision of 91%). Finally, combining the best models of this three using ensemble methods, voting classifier and stacking classifier (with a neural network with final estimator), the accuracy of the ensemble methods was 93% and 95%, respectively.**

Keywords: Diabetes, Classifiers, Pima Indian database, Machine learning

I. INTRODUCTION

Currently, Diabetes is among the ten diseases that kill the most in the world [1]. Diabetes is defined as a group of metabolic disorders exerting significant pressure on human health worldwide [2]. Responsible for triggering several other diseases, diabetes has been widely studied by researchers from all over the planet. Diabetes, also called Diabetes Mellitus, is a metabolic disease that increases blood sugar levels, for longer periods of time, when compared to a person without the disease [3]. There are several symptoms and among them are excesses: of giving in, of urine and of hunger. When left untreated it ends up triggering several complications in the individual. There are three main types of Diabetes: Diabetes Mellitus type I, type II and gestational [4]. As it is a silent disease, it is always necessary to pay attention mainly to cases in the family, as well as to foods consumed frequently.

Diabetes is in the group pf chronic diseases, the main characteristic of which is changes in the levels of insulin in the blood. Whether it is inappropriate production or use by the body. The disease is also a risk factor for others, such as kidney disease, blood vessel damage, blindness and heart disease development [5]. Diabetes is a global public health problem, often diagnosed in middle age to elderly people.

Research that contributes to early diagnosis, helped people to have a better quality of life. Supervised machine learning techniques can be applied as a way to improve the diagnosis. Patients who have a high probability of developing the disease can change habits that decrease the chance of developing the disease. How to adopt better nutrition, weight loss and exercise.

In this sense, Diabetes, like other diseases, needs attention ranging from special care to cutting-edge scientific research. In most studies like this, it is desired to know the relationship that one or more variables in a studied condition, more specifically, it is desirable to establish the chance of occurrence of such condition, given a specific image of the symptoms. In the presence of a disease in a person it is a binary event, and this random variable can be denoted by Y, whose assumed values are 0 for the absence of the disease and 1 for the presence. This brings us to classification models.

The binary classification proposes to classify elements of a given data set into two groups. That is, based on the characteristics of individuals, it tries to predict which group each one will belong to. One of the models, which comes to mind, when you want to work with binary classification, is the logistic model. This is because, one of the appropriate models to estimate the probability of an individual presenting the studied condition or the chance of acquiring it is the $Logistic Model$ [6].

However, there are a variety of binary classification models for working with classification problems. Among them, we can mention: Naive Bayes, Logistic Regression, Support Vector Machine, K Nearest Neighbors, Decision Trees, Random Forest, Perceptron and Neural Networks. All of these classification techniques will be addressed in this work, which will be compared using the following criteria: accuracy, precision, recall and F1 score.

Therefore, this work proposes, through machine learning techniques, to analyze a Diabetes database: Pima Indians is a database that brings results from group of females Native Americans living in Arizona, where research was on the major factors related to diabetes [5].

The work is divided as follows: chapter 1 provides an introduction about the disease and the classifiers, as well as, which techniques will be used in this article. Chapter 2 presents the result of five articles that modeled the same database. Chapter 3 describes the data, presents a descriptive

[*]Guilherme Michel is a Msc student in Statistics , Universidade de São Paulo Email: gmichelcarvalho@usp.br

[+] Jaqueline Lopes Dias is Professional Master student, Universidade de São Paulo Email:jldias@usp.br

[@]Marcos Jardel Henriques is a PhD student in Statistics, Universidade de São Paulo. Email: jardel_matematica@hotmail.com

analysis and the methods used in this article. Chapter 4 presents the results and conclusions.

II. Related Works

When carrying out surveys of works that have already analyzed the database that this article proposes to analyze, a range of published works is found. Several of them use statistical techniques, as well as machine learning. Among the works listed by us, which analyzed the Pima Indians Diabetes database, the work of Kriještorac *et al.* [7], which obtained good results, as well as good specificities and sensitivities after transforming the data, draws attention. These authors used several algorithms and among them was: Random Forest, SVM, J48 and KNN.

Researchers Jhaldiyal and Mishra [8] also studied the Pima Indians Diabetes data set using two techniques: main components and SVM. When analyzing the data using the main component technique, to reduce the number of dimensions and, using the error pruning reduction tree to train the data, they obtained an accuracy close to 79%. When working with SVM, also using main components, with the same objective of the first situation, they reached an accuracy of approximately 97%.

Kordos *et al.* [9], using the nearest K-neighbor algorithm (K-NN), when working with this database, after good parameter adjustments, it obtained an accuracy of approximately 77%.

Purnami *et al.* [10], achieved an accuracy of 76.73% in an attempt to predict people who in the future would be diabetics. This result was obtained using soft support vector machines (SSVM).

Using a ratio of $70\% - 30\%$ to conduct training and testing, with the intention of detecting Diabetes, Jahangeer *et al.* [11], modeled and compared three machine learning techniques : nearest K-neighbor algorithm (K-NN), with k = 1 and k = 3; Decision Tree-CART and SVM. With that, they were able to see that the parameter adjustments are essential in the modeling. Through these different techniques, the following precision was reached: Decision Tree-CART = $76,60\%$; K-NN com K = 1: $64,60\%$; SVM = $74,21\%$ e K-NN com K = 3: $68,79\%$. In this same sense, Shirke *et al.* [12], working with the SVM and Decision Tree classifiers reached the precision are 79.39% and 76.10% respectively.

Christobel *et al.* [13], worked on this problem using cross-validation and the nearest K-neighbor algorithm (K-NN). Through data processing, normalization, dimensioning and imputation, they achieved an accuracy of 74.74% through the use of cross-validation twice. When they performed cross-validation ten times, they reached an accuracy of 71.84%. An accuracy of 73.59% was obtained using kNN.

Performing selection of variables Hashi *et al.* [14] and reduced them by 37.5% and 50%. And through KNN (with K = 10 and 11) the accuracy increased from 74.48% to 81.17%; through SVM the accuracy was increased from 77.17% to 87.01%.

Also working with KNN, Ster and Dobnikar [15], in the detection of Diabetes, they reached an accuracy of 71.90% testing cross-validations ten times. In addition, they realized that when the parameters were not adjusted, the neural network performed better.

III. Material and Methods

A. Pima Indians Dataset

This database is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the database is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the database. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage, living in or near Arizona, USA, conducted in 1990. The database consists of several medical predictor variables and one target variable, Outcome. Predictor variables includes the number of pregnancies the patient has had, their BMI, insulin level, age, and so on.

Name	Description	Mean
NP	Numbers of times pregnant	3.8
GP	Plasma glucose concentration after 2 h	12.9
BP	Diastolic blood pressure (mmHg)	69.1
SKIN	Triceps skinfold thickness (mm)	20.5
INSULIN	Two-hour serum insulin	79.8
BMI	Body mass Index (weight (kg)/height m)	32
DPF	Diabetes pedigree function	0.5
AGE	Age of patient (years)	33.2

TABLE I: Predictor variables -Pima Indian diabetes database

Motivating the application of prediction techniques for this population is the slow and gradual onset of the disease [16]. Traditional diagnostic methods plasma glucose test can take up to 10 years to identify the disease [17]. The data set was chosen because it is widely used to test classification systems, making it easier to make comparisons between the results obtained with other models already proposed for the diagnosis of diabetes.

B. Pre-processing and Exploratory Analysis

Some observations have a value of zero, such as age and insulin, indicating that the database has some data reporting problems. Missing data are recorded as zero value in 376 of 768 observations in the Pima database. Due to the high proportion of values, the withdrawal of these observations would represent a great loss of data. It was adopted as a technique to correct missing values, a variant of the approach of closest neighbors, implemented by the Impute library (python). From a array with an initial applied function, we have the complete array. Regression of the k nearest neighbors is applied on the weighted average to find the imputed value.

The data were standardized, centralizing the mean and and component wise scale to unit variance.

In this simple visualization we note that the blood pressure and BMI variables have some null values. But, this variables

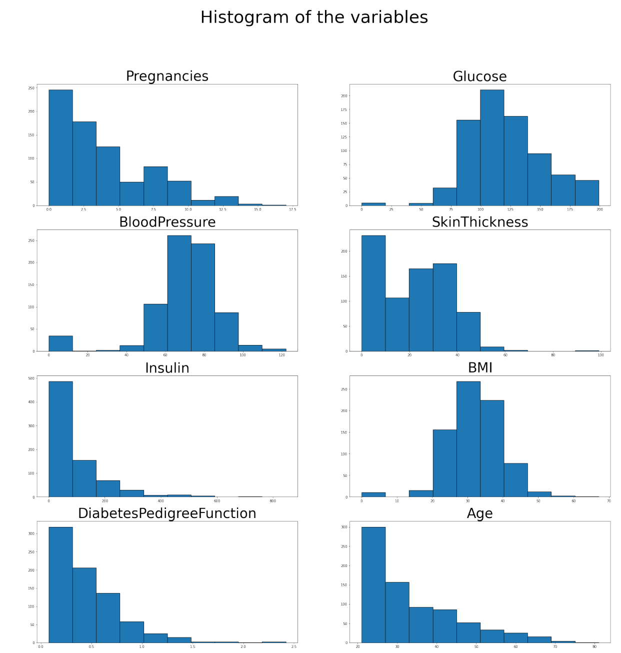

Fig. 1: Histogram of the variables in Pima Indian Database

cannot be null in practical terms. So, this null value represents in fact a missing value. We make a plot to see this in a intuitively way.

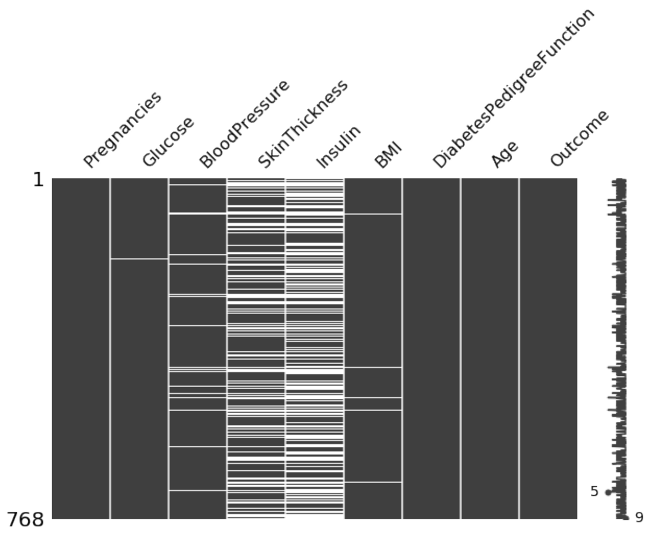

Fig. 2: Missing values for the variables

The family history of diabetes increases the chances of developing the disease. Gestational diabetes is also positively associated with the risk of developing type II diabetes in the following years. With respect to age and weight, often people in middle age and overweight people, the disease develops more. The group is young, has an average age of 33 years, a minimum age of 21 years and a maximum age of 80 years.

In the figure 2, the missing data are represented with a white line.

	Qt Missing Data
Glucose	5
BloodPressure	35
SkinThickness	227
Insulin	374
BMI	11

TABLE II: Quantity of missing observations in Pima Indian Database

We note in the figure 2 and table 2 there is a lot of missing data in our database. To deal with this we have to use some technique. A common way is to replace the missing value to the mean value of the column. Another way, more sophisticated, is to replace the value using a Regression technique. In this article we choose to use the KNN to replace the missing data. In the next subsection we will explain what is KNN. Another consideration in the exploratory analysis is that, the Pima Indian Database is unbalanced. We have more observations of the Class 0, that is, without diabetes. We show this in the following figure.

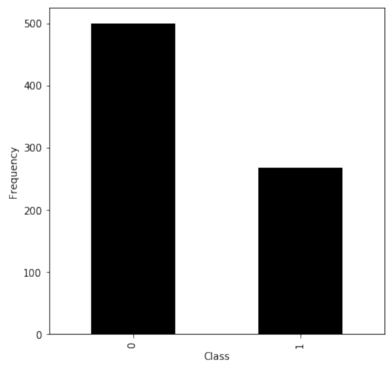

Fig. 3: Quantity of observations in each class

To overcome the problem of imbalance observed in the data set we use a technique called Synthetic Minority Over-sampling Technique(SMOTE),this approach is used to remove imbalance in the data by creating samples using the current data, see Ref. [18] for more details.

C. Feature Selection

We first look at the correlation matrix of the covariables.

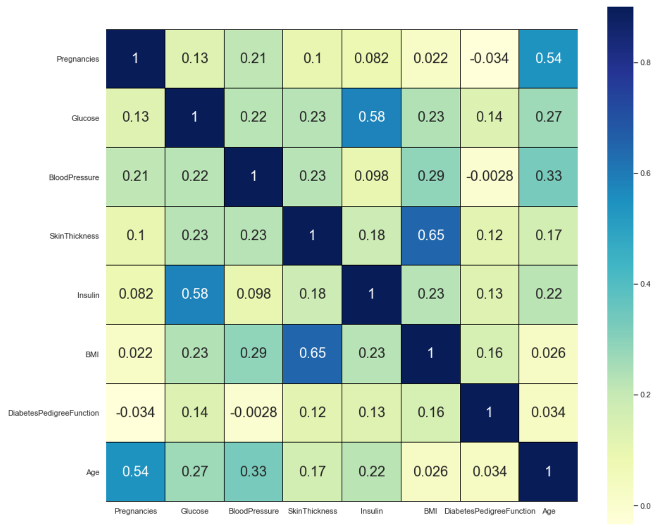

Fig. 4: Correlation Matrix

The most correlated variables are SkinThickness and BMI, with 0.65 pearson correlation. So in this database do not have to much correlation between the features and the feature space have a low dimension, therefore we use all the variables in the modelling.

D. Classification Models

1) Naive Bayes: Consider we have a sample of independent observations in a training set, lets say: $(\mathbf{x_1}, Y_1), (\mathbf{x_2}, Y_2), \cdots, (\mathbf{x_n}, Y_n) \sim (\mathbf{X}, Y)$. Where, $\mathbf{x_1}, \cdots, \mathbf{x_n}$ are the d dimensional vectors of features, and $Y_1, \cdots, Y_n$ are the target, in this case, Y_i are binary variables. Our goal is to build a function to predict new observations, based on this sample.

We can use the following Risk function to train the model:

$$R(g) = P(Y \neq g(X)) \tag{1}$$

So, we have to maximize:

$$g(x) = \underset{c \in \{0,1\}}{argmax} P(Y = c|\mathbf{x}) \tag{2}$$

Now, we have to estimate $P(Y = c|\mathbf{x})$, for this let's consider the Bayes theorem:

$$P(Y = c|\mathbf{x}) = \frac{f(\mathbf{x}|Y = c)P(Y = c)}{\sum\limits_{s \in \{0,1\}} f(\mathbf{x}|Y = s)P(Y = s)} \tag{3}$$

To get the estimate for $P(Y = c|\mathbf{x})$ we have to estimate everything in the right side of the (4) equation. The estimates for $P(Y = c)$ for $c \in \{0, 1\}$ are easily to get, using the proportion of this classes in the sample. The Naive-Bayes method assume that, for every $c \in \{0, 1\}$, $f(\mathbf{x}|Y = c)$ can be factored:

$$f((\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_d})|Y = c) = \prod_{j=1}^{d} f(\mathbf{x_j}|Y = c) \tag{4}$$

By assuming that distributions are independent. So, we can estimate every $f(\mathbf{x_j}|Y = c)$ by assuming for example:

$$X_j|Y = c \sim N(\mu_{j,c}, \sigma^2_{j,c}) \tag{5}$$

And the estimates for $\mu_{j,c}$ and $\sigma^2_{j,c}$ are given by maximum likelihood. Therefore, we have:

$$\hat{P}(Y = c|\mathbf{x}) = \frac{\hat{f}(\mathbf{x}|Y = c)\hat{P}(Y = c)}{\sum\limits_{c \in \{0,1\}} \hat{f}(\mathbf{x}|Y = s)\hat{P}(Y = s)} \tag{6}$$

Finally we have:

$$g(x) = \underset{c \in \{0,1\}}{argmax} \hat{P}(Y = c|\mathbf{x}) \tag{7}$$

And the decision rule can be: $g(x) = 0$ if $\hat{P}(Y = 0|\mathbf{x}) \geq \hat{P}(Y = 1|\mathbf{x})$ and $g(x) = 1$ otherwise.

2) Logistic Regression: In the logistic model, we use the values of a series of independent variables to predict the occurrence of the dependent variable (disease or condition). Thus, all variables considered in the model are controlled among themselves. As we use a series of independent variables, it is a multivariable problem. The regression model Multiple logistics is used for the case of regression with more than one explanatory variable.

According to Agresti (2007) [19]; Hosmer and Lemeshow (2013) [20] the general model for logistic regression with multiple Explanatory variables are defined as follows. Denote k predictors for an answer binary Y by $x_1, x_2, \ldots, x_k$. The model for the *odds log* is given by

$$\begin{aligned} \mathrm{logit(P(Y = 1))} &= \ln\left[\frac{\pi(x)}{1 - \pi(x)}\right] \\ &= \alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k, \end{aligned}$$

The parameter β_i refers to the effect of x_i in the *odds log* when $Y = 1$, controlling the others $x's$. In addition, perhaps the most important interpretation of the logistics model is in the calculation of the **odds** ($Odds$ - O) and the **odds ratio** ($OddsRatio$ - OR), which allows to quantify the magnitude of association between the explanatory variables and the response variable. *Odds* are given by

$$\frac{\pi(\boldsymbol{x})}{1 - \pi(\boldsymbol{x})} = \exp\left(\alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k\right).$$

This exponential relationship provides an interpretation for $\beta's$: The product by $\exp(\beta_i)$ for each unit plus x_i. See that when $\beta_i = 0$, $\exp(\beta_i) = 1$, therefore, the odds does not change when x_i changes [19].

3) Support Vector Machine: Developed in the 1990s, support vector machines - SVM, initially designed only for binary problems, ended up becoming a good classifier for a range of situations. With that, it became a reference when it comes to statistical machine learning. SVM is based on hyperplane concepts, which requires good knowledge of algebra [21].

Mathematically speaking, a definition of a hyperplane is not very complex and can be represented as:

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 = 0$$

What can be generalized to:

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p = 0$$

Thus, all points of the vector $\mathbf{x} = x_1, x_2, \cdots, x_p$, which satisfy the equation, will belong to the hyperplane.

Otherwise,

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p < 0$$

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p > 0$$

With that, we just need to calculate the sign of the equation, to know where the point under analysis is.

A classic example of how to increase the dimensionality of data is by transforming a two-dimensional space into a three-dimensional one

$$f(x_1, x_2) = (x_1^2, \sqrt{2}x_1 x_2, x_2^2)$$

When the limit of separation of the vectors is, even if approximately, linear, the SVM will present good results. When linearity is bad, just increase the dimensions to solve this problem.

The following are some ways to increase these dimensions:

- Linear kernel

$$K(\mathbf{X}, \mathbf{X}') = \mathbf{X} * \mathbf{X}'$$

When using a linear kernel, we will have results equal to the results obtained by SVM.

- Polynomial kernel

$$K(\mathbf{X}, \mathbf{X}') = (\mathbf{X} * \mathbf{X}' + c)^d$$

For $d = 1$ and $c = 0$, we will have results equal to SVM and the linear kernel. However, when $d > 1$ the linearity will be increased, proportionally to its growth.

- Gaussian Kernel (RBF)

$$K(\mathbf{X}, \mathbf{X}') = exp(\gamma ||\mathbf{X} - \mathbf{X}'||^2)$$

The value of $gamma$ controls the value of the kernel. The higher the value of $gamma$, the greater the flexibility of the model.

These are four of many other ways to increase dimensionality to solve the linearity problem. Each with its advantages and disadvantages, where, its hyperparameters can be found through cross-validation. Deciding which kernel to use will depend a lot on the situation you have to resolve. However, the Gaussian Kernel (RBF) is the most suitable for having only two hyperparameters to be optimized, and it is very flexible as to the classifier complexity.

4) K-Nearest Neighbours: The K-nearest neighbors (KNN) technique is a method of imputing and classifying data through the nearest neighbor. This method was proposed by Fukunaga and Narendra [22]. The method selects K objects with profiles similar to a missing observation Y_i through a distance measurement, such as Euclidean distance. After calculate all distances, one of the K observations closest to $_i$ what value occurs most often (for categorical data), or the average weighted of these values (for quantitative data), to fill in the missing value Y_i. This process does not change the variance or the average of the data set.

Two things must be defined to work with the KNN method: the metric of the distance calculation to be used and the size of *K*.

The metrics are chosen according to the problem, and the most used in calculating the distance between two points are:

- Euclidean Distance

$$D_E(p, q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

- Minkowski distance

$$D_M(p, q) = \left(\sum_{i=1}^{n} |p_i - q_i|^r \right)^{1/r}$$

- Chebyshev Distance

$$D_C(p, q) = \max_i (|p_i - q_i|)$$

In any case, $p = (p_1, \ldots, p_n)$ e $q = (q_1, \ldots, q_n)$ are two n-dimensional points and in the equation $D_M(p, q)$, r will be a chosen chosen constant.

The KNN method is often used to work with classification problems. Let's exemplify through the Euclidean distance, calculating the distance between two vectors.

$$e = \sqrt{\sum_{j=0}^{n} (X_{i,j} - X_{i+n,j})^2}$$

Given a point in space, KNN seeks to find the closest neighbors through the set of vectors:

$$\underset{X_{i,j} \subset X}{\arg\min} \sum_{i=0}^{K} \sqrt{\sum_{j=0}^{N} (X_{i,j} - t_i)^2}$$

Where: X, K and N are the neighbors and number of elements of the vector, and $X_{i,j}$ are the lines of the matrix.

What the vector will return $K_n = \{x_1, x_2, \cdots, x_n\}$ such that the sum of their distances is as short as possible, before the instance t.

5) Decision Trees: Decision tree is a supervised machine learning algorithm used in problems of classification and regression. Transforming a complex problem into multiple simpler problems, the algorithm is a strategy of dividing to win. For classifier a new item obtained from the tree is explained using set of rule taken on attribute values of training data. Through a simple algorithm to classify a new item, through of root nodes the instances with different features. The algorithm follows between questions and answers until reaches a leaf, that will be used as class for this case. The solutions to minor problems are combined into one tree, solving a more complex problem[23].

A decision tree can be easily and effectively converted into a set of rules, mapping from the root node to the leaf nodes step by step. Based on the concept of entropy and information gain [24]. We need to calculate two types of entropy using frequency tables as follows:

Entropy of of the target:

$$Entropy(T) = \sum_{i=1}^{c} -P_i log_2 P_i$$

and entropy using the frequency table of two features:

$$Entropy(T, X) = \sum_{c \in X} P(c)E(c)$$

The gain in information measures the reduction in entropy for each division:

$$Gain(T, X) = Entropy(T) - Entropy(T, X)$$

The objective is to maximize the gain.

Other classification criteria common is Gini

$$Gini(T) = \sum_{i} P_i(1 - P_i)$$

6) Random Forest: Random forest algorithm uses multiple learning algorithms seeking the best predictive performance. "The Random forest classifier creates multiple decision tree from randomly selected subset of training database. Aggregates the votes from different decision trees to decide the final class of test objects"[25]. It has two parts: Tree bagging and from tree bagging to random forest. Let n be the number of cases in sample at random of training set with replacement. The sample will be the training set of tree. We select the node that best split the node. There is no pruning, each tree is grown to the largest extent possible [26].

7) Perceptron: Rosenblatt's Perceptron is a simple Neural Network that can be composed by one single neuron, or several neurons arranged in a single layer. Is an algorithm for supervised learning of binary classifiers. The perceptron was invented in 1957 by Frank Rosenblatt [27]. The idea behind the Perceptron is to compose one or more output neurons, each of which is connected to several inputs units by means of connections that are characterized by weights [28]. One illustration of this model is presented bellow:

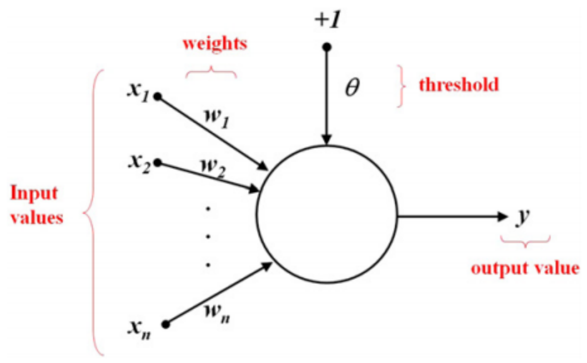

Fig. 5: Single Neural Perceptron, adapted from [28]

The output value is calculated as follows:

$$y = f(\sum_{i=1}^{n} w_i x_i + \theta) \tag{8}$$

The function f is called activation function and is given by:

$$f(x) = \begin{cases} 1, & \text{if } x > 0 \\ -1, & \text{otherwise} \end{cases} \tag{9}$$

In the Perceptron model we can use different penalty functions in regularization. Here we explore three: L1 norm, L2 norm and Elasticnet. For more details of regularization in Perceptron see Ref. [29].

8) Multilayer Perceptron: An Multilayer Perceptron(MLP) is an extension of Perceptron. In fact, MLP consists at least three layers of nodes. The following figure, adapted from [30] is showed an example of MLP.

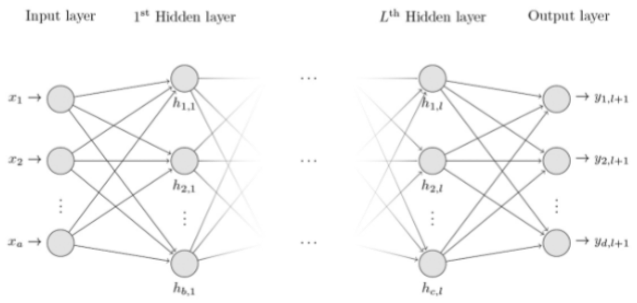

Fig. 6: MLP example, adapted from [30]

The learning processes occurs in MLP by changing connection weights, using backpropagation. For more details see Ref. [28]. In the case of MLP we can use several activation functions. Here we use logistic, identity, tanh and relu. This functions are:

Logistic:

$$f(x) = \frac{1}{1 + e^{(-x)}} \tag{10}$$

Identity:

$$f(x) = x \tag{11}$$

Tanh:

$$f(x) = tanh(x) \tag{12}$$

Relu:

$$f(x) = max(0, x) \tag{13}$$

9) Ensemble methods: Ensemble methods use several classifiers to obtain better predictive performance than each classifier alone. In this work we use two ensemble method: Voting Classifier/Majority Rule and the Stacking Classifier. The Voting classifier is very simple to understand, suppose for a certain classification problem we have three different classifications, the voting classifiers is a way to combine this three rules in one classifier. A common way to combine is to use the mode. The stacking classifier is similar, but the rule to get the final prediction is not only simple the mode, but every classification rule is passed to a new classifier and this meta classifier is used to get the final prediction.

10) Performance Metrics: For evaluate our models we choose four different metrics: Accuracy, Recall, Precision and F1-Score. The choice of the evaluation metrics influences the firm that measures the performance of machine learning algorithms, different performance metrics were adopted to evaluate the algorithms. First we have to define the confusion matrix, it a kind of contingency table with two dimensions. Based on the confusion matrix, the relationship between true and false positives and negatives. Where we seek to minimize false negative and positive results.

- Accuracy: number of correct predictions made by the model. That is, the proportion of the true positives and negatives in relation to all the predictions of the confusion matrix. Applied to classes with proportions of data are nearly balanced.
- Recall: proportion of true positives in relation to those who actually tested positive. To minimize false negatives, the measure will approach
- Precision: discover the proportion of true positives in relation to those that are true and false positives.
- pandas: used for data analysis and manipulation tool.
- f1-score: a unique score to represent precision and recall, based on the harmonic average between the two metrics.

TABLE III Confusion Matrix

		True diagnosis		
		Positive	Negative	Total
Predicted	Positive	TP	FP	$TP + FP$
	Negative	FN	TN	$FN + TN$
	Total	$TP + FN$	$FP + TN$	N

In terms of Confusion Matrix these metrics are defined by:

Accuracy

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{14}$$

Recall

$$Recall = \frac{TP}{TP + FN} \tag{15}$$

Precision

$$Precision = \frac{TP}{TP + FP} \tag{16}$$

F1-Score

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{17}$$

E. Implementation:

We use python to implement all of this methods. The following libraries are required:

- NumPy: used for linear algebra and other operations.
- seaborn: used for statistical data visualization.
- matplotlib: used for plot and data visualization.
- pandas: used for data analysis and manipulation tool.
- scikit-learn: used for all the classification methods and preprocessing methods.
- Impyute: used for deal with missing data imputation.
- imblearn: used for deal with imbalanced data.

IV. Experiments

A. Description of experiments and performance of classifiers

For each model we describe in the Section of material and methods we try a few experiments. In the next subsections we show the results for some experiments. Each measure for performances of the models is an average of 10-fold cross validation. The database we use in each experiment is the database with replacement of the miss values for a value given by a regression(We use the package impyute) and we balanced the data using SMOTE technique, which is implemented in the package imblearn.

1) Naive-Bayes: First we use the simple model Naive-Bayes, defined before.

	Accuracy	Recall	Precision	f1-Score
Naive Bayes	0.76	0.74	0.760417	0.759904

2) Logistic Regression: The results for logistic regression are:

	Accuracy	Recall	Precision	f1-Score
Logistic Regression	0.8	0.78	0.800481	0.79992

3) Support Vector Machine: We use SVM Classifier with differents kernels, the results are showed in the following table:

	Accuracy	Recall	Precision	f1-Score
SVM with poly Kernel of degree 2	0.68	0.70	0.680288	0.679872
SVM with poly Kernel of degree 3	0.78	0.90	0.797114	0.776786
SVM linear	0.79	0.76	0.791048	0.789811
SVM with rbf Kernel	0.87	0.94	0.877397	0.869360

4) K-Nearest-Neighbors: We use different configurations of KNN, with 3 or 5 Nearest Neighbors and the Minkowski distance with different values of p. The results are showed in the following table:

	Accuracy	Recall	Precision	f1-Score
KNN with 3nn, distance metric with p=1	0.89	1.00	0.909836	0.888653
KNN with 3nn, distance metric with p=2	0.88	1.00	0.903226	0.878247
KNN with 5nn, distance metric with p=1	0.84	1.00	0.878788	0.835796
KNN with 5nn, distance metric with p=2	0.84	0.98	0.868924	0.836801

5) Decision Trees: We choose two decision tree, one with gini criterion and the other with entropy.The maximum number of features to consider when looking the best split was the total number of features in the database.

	Accuracy	Recall	Precision	f1-Score
Decision Tree with gini criterion	0.83	0.94	0.846784	0.827918
Decision Tree with entropy criterion	0.81	0.86	0.813131	0.809524

6) Random Forest: For the random forest classifier we test two criterion, gini and entropy. In addition, we choose the number of estimators as 100 and 600. The results for this classifier are showed in the following table:

	Accuracy	Recall	Precision	f1-Score
Random Forest, n_estimators=100, Gini	0.90	0.92	0.900641	0.899960
Random Forest, n_estimators=100, Entropy	0.91	0.94	0.911481	0.909919
Random Forest, n_estimators=600, Gini	0.93	0.96	0.931554	0.929937
Random Forest, n_estimators=600, Entropy	0.92	0.96	0.922705	0.919872

7) Perceptron: We use the Perceptron with different penalty functions and the rate of learning 0.01 and $max\_iterations = 1000$. The penalty functions used was l1 norm, l2 norm and elasticnet.

	Accuracy	Recall	Precision	f1-Score
Perceptron with l1 penalty	0.67	0.46	0.706411	0.654776
Perceptron with l2 penalty	0.63	0.46	0.646992	0.618989
Perceptron with elastic net penalty	0.63	0.46	0.646992	0.618989

8) Multilayer Perceptron: We use 1000 hidden layers with 1000 neurons for MLP Classifier and differents activation functions, the results are showed in the following table:

	Accuracy	Recall	Precision	f1-Score
MLP with (1000,1000)(Activation logistic)	0.82	0.82	0.820000	0.820000
MLP with (1000,1000)(Activation identity)	0.81	0.80	0.810124	0.809981
MLP with (1000,1000)(Activation tanh)	0.91	0.98	0.918197	0.909557
MLP with (1000,1000)(Activation relu)	0.89	0.96	0.897797	0.889458

9) Ensemble Methods: We use meta-classifiers including Voting Classifier and Stacking Classifier. For do this we choose the best tree models defined before and perform the ensemble classifier. The models are: MLP with 1000 hidden layers and 1000 neurons each layer and $tanh$ activation, Random Forest with 600 number of estimators and Gini criterion and KNN with 3 nearest neighbors and Minkoski distance with $p = 1$. For the Stacking Classifier we use a MLP with the final estimator with 1000 hidden layers and 1000 neurons. The results are showed in the following table:

	Accuracy	Recall	Precision	f1-Score
Voting Classifier	0.93	0.934343	0.98	0.929825
Stacking Classifier	0.95	0.95018	0.94	0.949995

B. Results

Measuring accuracy, recall, precision and f1-score in 8 different algorithms, advantages and disadvantages of different methods are presented. Once database were previously balanced, accuracy will be adopted as a measure of comparison between the models evaluated.

All algorithms were tested considering all variables and applying 10-fold cross-validation, due to their efficiency in small data sets. Previous studies [7] that explored different combinations of variables in the same data set, achieved better results using all variables. From the results obtained it can be concluded that the ensemble methods (Stacking Classifier and Random Forest) and neural networks (MLP), present a better performance. Considering different weaker algorithms proved to be more powerful in both accuracy and precision for the classification of diabetes for the Pima database. Ensemble Methods Stacking Classifier, get the highest accuracy to be 95% and Random Forest with 600 estimators and considering gini decision criteria (accuracy 93%).

The SVM classification model using the RBF kernel function obtained a better accuracy performance (87%) when compared with its linear application (79%) and polynomial of degree 3 (78%), a similar result found by other experiments [Krije]. Obtaining better results than Decision Tree with gini criterion, Logistic Regression and Naive Bayes algorithms, as in previous experiments [11]. However, the results were lower than the KNN considering the three closest neighbors (89%).

One intersting thing to observe is that, the KNN classifier give to us a very high recall measure, which means, this method don't give any false negative in three configurations tested. Decision Tree model uses Gini criterion and entropy criterion, the first being the one with the highest accuracy (83%) and precision (84%). Obtaining a low performance, surpassing only Naive Bayes (76%) and logistic regression (80%).

V. Conclusions

In this article we review the main current techniques that can be used for binary classification using the Pima Indian Database. We use techniques to deal with the missing values and imbalanced classes. To deal with missing data we use a regression technique to replace them. Besides that, to deal with imbalanced data we use the SMOTE technique [18]. We compared several classifiers, Naive-Bayes, Logistic Regression, SVM, KNN, Decision Tree,Random Forest, Perceptron and Multilayer Perceptron using different hyperparameters. Considering the models separately the best models are KNN, with 3 nearest neighbors and Minkonski distance,(89% accuracy), Random forest with 600 number of estimators and Gini criterion(93% accuracy) and MLP with 1000 hidden layers of 1000 neurons and $tanh$ in activation function(91% accuracy). Finally we combined this tree best models using ensemble methods, voting classifier and stacking classifier(with a neural network in final estimator), the accuracy of the ensemble methods was 93% and 95%, respectively.

VI. Source Code

All the code used in this paper can be access in the following link: `https://github.com/gmichelcarvalho/Project`.

References

[1] W. H. Organization, "Diabetes," 2020.
[2] e. a. Kavakiotis, I., "Machine learning and data mining methods in diabetes research," *Procedia Computer Science*, 2017.
[3] D. P. Cury, R. Okamoto, S. Tumelero, and M. C. Madeira, "Comparação de índice glicêmico e gordura corporal entre portadores de diabetes, praticantes e não praticantes de atividade física," *EFDeportes.com, Revista Digital. Buenos Aires*, vol. 16, no. 157, 2011.
[4] U. Masharani and M. S. German, "Pancreatic hormones and diabetes mellitus," in *Greenspan's basic & clinical endocrinology* (D. G. Gardner and D. Shoback, eds.), ch. 17, New York: McGraw-Hill Medical, 9th ed., 2011.
[5] K. J. Kriještorac M., Halilović A., "The impact of predictor variables for detection of diabetes mellitus type-2 for pima indians," *Advanced Technologies, Systems, and Applications IV -Proceedings of the International Symposium on Innovative and Interdisciplinary Applications of Advanced Technologies*, 2019.
[6] G. S. G. A. S. J. M. P. de., "Utilização de estratificação e modelo de regressão logística na análise de dados de estudos caso-controle.," *Revista Saúde Pública*, vol. 29, no. 4, pp. 283–289, 1995.
[7] M. Kriještorac, A. Halilović, and J. Kevric, "The impact of predictor variables for detection of diabetes mellitus type-2 for pima indians," *Springer*, vol. 83, 2020.
[8] T. Jhaldiyal and P. K. Mishra, "Analysis and prediction of diabetes mellitus using pca, rep and svm," *International Journal of Engineering and Technical Research (IJETR)*, vol. 2, no. 8, pp. 164–166, 2014.
[9] M. Kordos, M. Blachnik, and D. Strzempa, "Do we need whatever more than k-nn?," *Proceedings of the 10th International Conference on Artificial Intelligence and Soft Computing (Springer-Verlag)*, vol. I, pp. 414–421, 2010.
[10] S. W. Purnami, A. Embong, and J. M. Zain, "A new smooth support vector machine and its applications in diabetes disease diagnosis," *Journal of Computer Science*, vol. 5, no. 12, pp. 1003–1008, 2009.
[11] S. Jahangeer, M. Zaman, M. Ahmed, and M. Ashraf, "An empirical comparison of supervised classifiers for diabetic diagnosis," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 1, pp. 311–315, 2017.
[12] M. S. Barale and D. T. Shirke, "Cascaded modeling for pima indian diabetes data," *International Journal of Computer Applications*, vol. 139, no. 11, pp. 1–4, 2016.
[13] Y. A. Christobel and P. Sivaprakasam, "Improving the performance of k-nearest neighbor algorithm for the classification of diabetes dataset with missing values," *IJCET*, vol. 3, no. 3, p. 155–167, 2012.
[14] E. K. Hashi, S. U. Zaman, and R. Hasan, "Developing diabetes disease classification model using sequential forward selection algorithm," *Int. J. Comput. Appl.*, vol. 180, no. 5, pp. 1–6, 2017.
[15] B. Ster and A. Dobnikar, "Neural networks in medical diagnosis: comparison with other methods," *Proceedings of the International Conference on Engineering Applications with Neural Networks*, p. 427–430, 1996.
[16] Y. Hayashi and S. Yukita, "Rule extraction using recursive-rule extraction algorithm with j48graft combined with sampling selection techniques for the diagnosis of type 2 diabetes mellitus in the pima indian dataset," *Informatics in Medicine Unlocked*, 2016.
[17] H. R. and H. N., *Essential endocrinology and diabetes*. Wiley-Blackwell, 2006.
[18] K. H. L. Chawla, Nitesh. Bowyer and W. Kegelmeyer, "Synthetic minority over-sampling technique," *Elsevier*, 2018.
[19] A. Agresti, *An Introduction to Categorical Data Analysis*. Second Edi ed. New Jersey: JohnWiley Sons, 2007.
[20] S. S. R. X. HOSMER, D. W. J.; LEMESHOW, *Applied Logistic Regression*. 3a. ed. New Jersey: [s.n.], 2013.
[21] C. Cortes and V. Vapnik., "Support vector machine.," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
[22] K. Fukunaga and P. Narendra, "A branch and bound algorithm for computing k-nearest neighbors," *IEEE transactions on computers*, vol. 100, no. 7, pp. 750–753, 1975.
[23] e. a. Katti Faceli, *Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina* . Grupo Editorial Nacional, 2011.
[24] K. P. T. S. Choubey, D. K., "Performance evaluation of classification methods with pca and pso for diabetes," *Network Modeling Analysis in Health Informatics and Bioinformatics*, 2019.
[25] N. P. Tigga and S. Garg, "Prediction of type 2 diabetes using machine learning classification methods," *Procedia Computer Science*, 2020.
[26] J. Kandhasamy and S. Balamurali, "Performance analysis of classifier models to predict diabetes mellitus," *Procedia Computer Science*, 2015.
[27] F. Rosenblatt, "The perceptron - a perceiving and recognizing automaton," *Cornell Aeronautical Laboratory.*, 1957.
[28] L. Vanneschi and M. Clastelli, "Multilayer perceptron," *Elsevier*, 2018.
[29] Z. et al, "Regularized structured perceptron: A case study on chinese word segmentation, pos tagging and parsing," *aclweb*, 2019.
[30] C. et al, "Multilayer perceptron architecture optimization using parallel computing techniques," *Plus one*, 2017.