FACULTAD DE INGENIERÍA
REVISTA DE LA FACULTAD DE INGENIERIA
www.revistaingenieria.uda.cl

Statistical Machine Learning for Predicting Diabetes Cases in Indigenous Women

Guilherme Michel Lima de Carvalho[1], Jaqueline Lopes Dias[2], Marcos Jardel Henriques[1], Felipe Padula Sanches[3], Oilson Alberto Gonzatto Junior[1], Roseli Aparecida Francelin Romero[3], and Francisco Louzada[1,2]

[1]Programa Interinstitucional de Pós-Graduação em Estatística (PIPGEs) UFSCar-USP (Universidade Federal de São Carlos (DES-UFSCar) e Universidade de São Paulo (ICMC-USP)); *email: jardel@usp.br*
[2]Mestrado Profissional em Matemática, Estatística e Computação Aplicadas à Indústria (MeCAI) do Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (ICMC-USP)
[3]Programa Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC) do Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (ICMC-USP)

ABSTRACT

This work proposes, through machine learning techniques, to analyze a diabetes database: Pima Indians. This database is composed of data that presents results from a group of Native American women living in Arizona, where research was conducted on the main factors related to diabetes. Thus, in this article, this dataset was analyzed using the following techniques: Naive Bayes, Logistic Regression, Support Vector Machine, K Nearest Neighbors, Decision Trees, Random Forest, Perceptron, and Multilayer Perceptron. The results of this work were compared to the findings of several scientific articles that also analyzed the same database.

Keywords: Diabetes, Classifiers, Pima Indian database, Statistical Machine Learning, Data Science.

RESUMEN

Este trabajo propone, a través de técnicas de aprendizaje automático, analizar una base de datos sobre diabetes: los indios Pima. Esta base de datos está compuesta por datos que presentan resultados de un grupo de mujeres nativas americanas que viven en Arizona, donde se llevó a cabo una investigación sobre los principales factores relacionados con la diabetes. Así, en este artículo, se analizó este conjunto de datos utilizando las siguientes técnicas: Naive Bayes, Regresión Logística, Máquina de Vectores de Soporte, K Vecinos más Cercanos, Árboles de Decisión, Bosques Aleatorios, Perceptrón y Perceptrón Multicapa. Los resultados de este trabajo se compararon con los hallazgos de varios artículos científicos que también analizaron la misma base de datos.

Palabras claves: Diabetes, Clasificadores, Base de datos de los indios Pima, Aprendizaje Automático Estadístico, Ciencia de los datos

1 Introduccion

Currently, diabetes is among the top ten diseases that cause the most deaths worldwide [24]. Diabetes is defined as a group of metabolic disorders that exert significant pressure on human health globally [19]. Responsible for triggering several other diseases, diabetes has been widely studied by researchers from all over the planet. Diabetes, also known as diabetes mellitus, is a metabolic disease that increases blood sugar levels for longer periods compared to a person without the disease [7]. There are several symptoms, including excessive thirst, urination, and 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 [23]. As it is a silent disease, it is always necessary to pay attention, especially to cases in the family, as well as to frequently consumed foods.

Diabetes is in the group of chronic diseases, characterized by changes in insulin levels in the blood, whether due to inappropriate production or utilization by the body. The disease is also a risk factor for other conditions, including kidney disease, damage to blood vessels, blindness, and the development of heart disease [22]. Diabetes is a global public health problem, often diagnosed in middle-aged to elderly individuals.

Research that contributes to early diagnosis has helped people achieve a better quality of life. Supervised machine learning techniques can be applied as a means to enhance the diagnostic process. Patients with a high probability of developing the disease can alter their habits to reduce the likelihood of its occurrence, such as adopting better nutrition, engaging in weight loss, and incorporating regular exercise.

In this context, diabetes, like other diseases, requires attention ranging from special care to cutting-edge scientific research. In most studies of this nature, the goal is to understand the relationship between one or more variables in a studied condition. Specifically, it is desirable to establish the likelihood of the occurrence of such a condition, given a specific set of symptoms. The presence of a disease in a person is a binary event, and this random variable can be denoted by Y, with assumed values of 0 for the absence of the disease and 1 for its presence. This leads us to classification models.

Binary classification aims to categorize elements in a given dataset into two groups. It attempts to predict, based on the characteristics of individuals, which group each one will belong to. One of the models that comes to mind when working with binary classification is the logistic model. This is because the logistic model is one of the suitable models for estimating the probability of an individual presenting the studied condition or the likelihood of acquiring it [8].

However, there are various binary classification models for addressing 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 discussed in this work, and they 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: the Pima Indians. This database contains results from a group of Native American females living in Arizona, where research was conducted on the major factors related to diabetes [22].

The work is divided as follows: Chapter 1 provides an introduction to the disease and the classifiers, as well as the techniques that will be used in this article. Chapter 2 presents the results of five articles that modeled the same database. Chapter 3 describes the data, presents a descriptive analysis, and outlines the methods used in this article. Chapter 4 presents the results and conclusions.

1.1 Related Works

When conducting a survey of works that have already analyzed the database proposed in this article, a variety of published works are found. Several of them utilize statistical techniques as well as machine learning. Among the works we have reviewed that analyzed the Pima Indians Diabetes database, the study by Kriještorac *et al.* [21] stands out. They achieved favorable results, including good specificities and sensitivities, after transforming the data. The authors employed sev-

eral algorithms, including Random Forest, SVM, J48, and KNN.

Researchers Jhaldiyal and Mishra [16] also explored the Pima Indians Diabetes dataset using two techniques: principal components and SVM. In their analysis using the principal component technique to reduce the number of dimensions and employing error pruning reduction tree to train the data, they achieved an accuracy close to 79%. When working with SVM, also utilizing principal components with the same objective as the first scenario, they attained an accuracy of approximately 97%.

Kordos *et al.* [20], employing the nearest K-neighbor algorithm (K-NN) with appropriate parameter adjustments while working with this database, achieved an accuracy of approximately 77%.

Purnami *et al.* [25] achieved an accuracy of 76.73% in an attempt to predict individuals who would be diabetics in the future. This result was obtained using soft support vector machines (SSVM).

Using a $70\% - 30\%$ split for training and testing to detect diabetes, Jahangeer *et al.* [15] modeled and compared three machine learning techniques: the nearest K-neighbor algorithm ($K - NN$) with $k = 1$ and $k = 3$, Decision Tree-CART, and SVM. Through these different techniques, they found that parameter adjustments are essential in the modeling process. The precision achieved for each technique was as follows: Decision Tree-CART $= 76,60\%$; K-NN with $k = 1$: $64,60\%$; SVM $= 74,21\%$; and K-NN with $k = 3$: $68,79\%$. Similarly, Shirke *et al.* [2], working with SVM and Decision Tree classifiers, achieved precisions of $79,39\%$ and $76,10\%$, respectively.

Christobel *et al.* [5] addressed this problem using cross-validation and the nearest K-neighbor algorithm (K-NN). Through data processing, including 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 variable selection, Hashi *et al.* [12] reduced them by $37,5\%$ and 50%. Through KNN (with $K = 10$ and 11), the accuracy increased from $74,48\%$ to $81,17\%$, and through SVM, the accuracy increased from $77,17\%$ to $87,01\%$.

Also working with KNN, Ster and Dobnikar [29], in the detection of diabetes, achieved an accuracy of $71,90\%$ in ten cross-validation tests. Additionally, they observed that when the parameters were not adjusted, the neural network performed better.

Kandhasamy and Balamurali [17] worked with the classifiers KNN, SVM, and Random Forest, obtaining accuracies of $73,17\%$, $73,74\%$ and $71,74\%$, respectively.

Shankaracharya et al. [28], using the logistic regression technique and working with eight variables, achieved an accuracy of $79,17\%$. However, the accuracy increased slightly to $80,21\%$ when the authors worked with only the four most important variables.

2 Seccion II

2.1 Pima Indians Dataset

This database originates from the National Institute of Diabetes and Digestive and Kidney Diseases. Its primary objective is to diagnostically predict whether a patient has diabetes based on specific diagnostic measurements included in the database. Several constraints were applied when selecting instances from a larger database. Notably, all included patients are females, at least 21 years old, of Pima Indian heritage, residing in or near Arizona, USA, and the data was collected in 1990. The database comprises various medical predictor variables and one target variable labeled as "Outcome." Predictor variables encompass the number of pregnancies, BMI, insulin level, age, and other relevant factors. In Table 1 it is possible to check the symbols and names of the variables and the average value of each of them.

Table 1: Predictor variables - Pima Indian diabetes database

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

Fonte: The authors

Motivating the application of prediction techniques for this population is the slow and gradual onset of the disease [13]. Traditional diagnostic methods, such as the plasma glucose test, can take up to 10 years to identify the disease [26]. The dataset was chosen because it is widely used to test classification systems, making it easier to compare the results obtained with other models already proposed for the diagnosis of diabetes.

2.2 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 values in 376 out of 768 observations in the Pima database. Due to the high proportion of values, the removal of these observations would represent a significant loss of data. A technique to correct missing values was adopted, a variant of the closest neighbors approach, implemented by the Impute library in Python. From an array with an initially applied function, we obtain the complete array. Regression of the k nearest neighbors is applied to the weighted average to find the imputed value.

The data were standardized by centralizing the mean and scaling component-wise to unit variance.

In this simple visualization, we note that the blood pressure and BMI variables have some null values. However, these variables cannot be null in practical terms. Thus, these null values actually represent missing values. We create a plot to illustrate this intuitively.

A 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. Regarding age and weight, the disease often develops more in people in middle age and overweight individuals. The group is relatively young, with an average age of 33 years, a minimum age of 21 years, and a maximum age of 80 years.

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

Table 2: Quantity of missing observations in Pima Indian Database

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

Fonte: The authors

In Figure 2 and Table 2, we observe a significant amount of missing data in our database. To address this issue, various techniques can be applied. A common approach is to replace the missing values with the mean value of the column. Alternatively, a more sophisticated method involves using a regression technique. In this article, we choose to use KNN for replacing the missing data. The next subsection will elaborate on what KNN entails.

Another consideration in the exploratory analysis is that the Pima Indian Database is unbalanced. We have more observations for Class 0, indicating individuals without diabetes. This is illustrated in the following figure.

To address the issue of imbalance observed in the dataset, we employ a technique called Synthetic Minority Over-sampling Technique (SMOTE). This approach aims to rectify data imbalance by creating synthetic samples based on the existing data. For more details, refer to Ref. [3].

2.3 Feature Selection

We first look at the correlation matrix of the covariables. We can perform this visualization in Figure 4.

The most correlated variables are SkinThickness and BMI, with a Pearson correlation of $0,65$. In this database, there is not much correlation between the features, and the feature space has low dimensionality. Therefore, we use all the variables in the modeling.

2.4 Classification Models

2.4.1 Naive Bayes

Consider that we have a sample of independent observations in a training set, let's 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.

2.4.2 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. Since we use a series of independent variables, it is a multivariable problem. The Multiple Logistic Regression model is employed for cases of regression with more than one explanatory variable.

According to Agresti (2007) [1]; Hosmer and Lemeshow (2013) [14] 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

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

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

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

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 [1].

2.4.3 Support Vector Machine

Developed in the $1990s$, Support Vector Machines (SVM), initially designed only for binary problems, ended up becoming a robust classifier for a range of situations. Consequently, it became a reference in statistical machine learning. SVM is based on hyperplane concepts, which require a good understanding of algebra [6].

Mathematically speaking, the 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 determine 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 yield good results. When linearity is inadequate, simply increasing the dimensions can 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 address the linearity problem. Each has its advantages and disadvantages, and their hyperparameters can be determined through cross-validation. The choice of which kernel to use will depend largely on the situation you need to resolve. However, the Gaussian Kernel (RBF) is the most suitable because it has only two hyperparameters to be optimized and is very flexible in terms of classifier complexity.

2.4.4 K-Nearest Neighbours

The K-nearest neighbors (KNN) technique is a method for imputing and classifying data based on the nearest neighbors. This method was proposed by Fukunaga and Narendra [11]. The method selects K objects with profiles similar to a missing observation Y_i through a distance measurement, such as Euclidean distance. After calculating all distances, one of the K observations closest to Y_i is chosen, and its value occurs most often (for categorical data) or is the weighted average of these values (for quantitative data), filling in the missing value Y_i. This process does not change the variance or the average of the dataset.

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

The metrics are chosen according to the problem, and the most commonly 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:

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

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.

2.4.5 Decision Trees

A decision tree is a supervised machine learning algorithm used in classification and regression problems. Transforming a complex problem into multiple simpler problems, the algorithm employs a strategy of dividing to conquer. To classify a new item, the algorithm explains it using a set of rules derived from attribute values in the training data. The process involves navigating through the tree nodes based on different features until reaching a leaf, which is used as the class for the given case. The solutions to smaller problems are combined into one tree, addressing a more complex problem [18].

Converting a decision tree into a set of rules is a straightforward and efficient process, involving mapping from the root node to the leaf nodes step by step. This conversion is guided by the principles of entropy and information gain [4]. The calculation involves determining two types of entropy using frequency tables, as outlined below:

Entropy 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

$$Gini(T) = \sum_i P_i(1 - P_i)$$

2.4.6 Random Forest

The Random Forest algorithm employs multiple learning algorithms to optimize predictive performance. According to Tigga [30], the Random Forest classifier generates numerous decision trees from randomly selected subsets of the training database. It then aggregates the votes from these diverse decision trees to determine the final class for test objects. This process consists of two main parts: Tree bagging and the transition from tree bagging to a Random Forest.

Let $'n'$ represent the number of cases randomly selected with replacement from the training set. This sample becomes the training set for each tree. The algorithm selects the node that best splits the node, and notably, there is no pruning applied; each tree is grown to its maximum extent [17].

2.4.7 Perceptron

Rosenblatt's Perceptron serves as a straightforward Neural Network, capable of being configured with a single neuron or multiple neurons organized in a single layer, as illustrated in Figure 5. It functions as an algorithm designed for supervised learning of binary classifiers. Originally proposed by Frank Rosenblatt in 1957 [27], the Perceptron concept involves the creation of one or more output neurons. Each of these neurons is connected to several input units through connections characterized by weights [31]. An illustration of this model is presented below:

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. [10].

2.4.8 Multilayer Perceptron

An Multilayer Perceptron(MLP) is an extension of Perceptron. In fact, MLP consists at least three layers of nodes. To build this illustration, we adapted [9], as shown in Figure 6. In other words, this is an example of MLP.

The learning processes occurs in MLP by changing connection weights, using backpropagation. For more details see Ref. [31]. 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:

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

2.4.9 Ensemble methods

Ensemble methods use several classifiers to obtain better predictive performance than each classifier alone. In this work, we use two ensemble methods: the Voting Classifier/Majority Rule and the Stacking Classifier. The Voting classifier is straightforward to understand. Suppose, for a certain classification problem, we have three different classifiers. The voting classifier is a way to combine these three rules into one classifier. A common way to combine them is to use the mode. The stacking classifier is similar, but the rule to get the final prediction is not only simply the mode; instead, every classification rule is passed to a new classifier, and this meta-classifier is used to get the final prediction.

2.4.10 Performance Metrics

To evaluate our models, we choose four different metrics: Accuracy, Recall, Precision, and F1-Score. The choice of 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, which is a kind of contingency table with two dimensions. Based on the confusion matrix, we assess the relationship between true and false positives and negatives, aiming 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.

Table 3: Confusion Matrix

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

- 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.

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

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

The evaluation and diagnostic metrics of the models used in this work were calculated according to the Table 3 and the equations 14, 15, 16 e 17

2.5 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.

3 Experiments

3.1 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.

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.

4 Sección III

4.1 Results

In a brief visualization of the data, which were worked on in this scientific article, it can be seen how the set of information treated here behaves through the histograms in Figure 1.

In Figure 3 we can understand the frequency of diabetics and non-diabetics between the sick and non-sick groups. In other words, note that approximately one third of individuals have diabetes, while two thirds have not yet developed the disease.

Continuing, you have to measuring accuracy, recall, precision, and F1-score across 8 different algorithms reveals the advantages and disadvantages

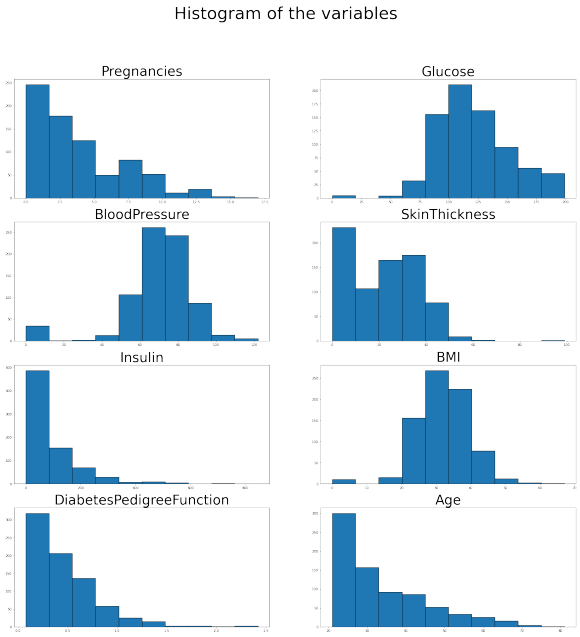

Figure 1: Histogram of the variables in Pima Indian Database

Table 4: Comparison of Classification Models through Metrics: Accuracy, Recall Precision and F1-Score

	Accuracy	Recall	Precision	f1-Score
Naive Bayes	0.76	0.74	0.760417	0.759904
Logistic Regression	0.8	0.78	0.800481	0.79992
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
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
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
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
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
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
Voting Classifier	0.93	0.934343	0.98	0.929825
Stacking Classifier	0.95	0.95018	0.94	0.949995

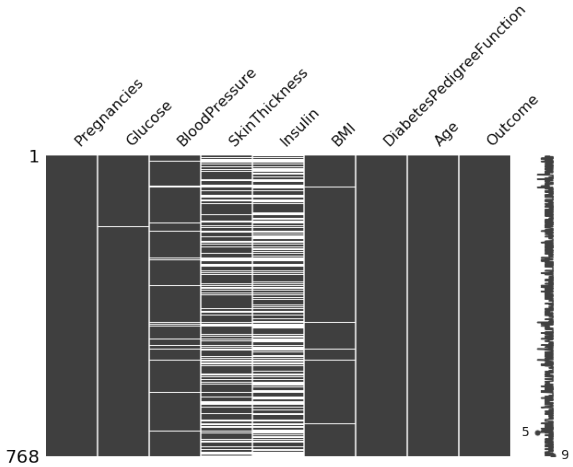

Figure 2: Missing values for the variables
Fonte: The authors

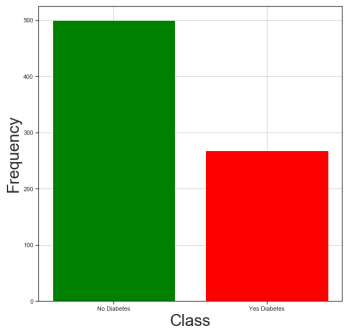

Figure 3: Quantity of observations in each class
Fonte: The authors

Figure 4: Correlation Matrix
Fonte: The authors

of different methods. Since the database was previously balanced, accuracy will be adopted as a measure for comparing the evaluated models.

All algorithms were tested considering all variables and applying 10-fold cross-validation, chosen for its efficiency in small datasets. Previous studies, such as Kevric et al. [21], explored different combinations of variables in the same dataset and achieved better results using all variables. From the results obtained, it can be concluded that ensemble methods (Stacking Classifier and Random Forest) and neural networks (MLP) present better performance, as can be seen in Table 4. Considering different weaker algorithms, they proved to be more powerful in both accuracy and precision for the classification of diabetes in the Pima database. The Stacking Classifier, among ensemble methods, achieved the highest accuracy at 95%, and the Random Forest with 600 estimators and considering the Gini decision criteria reached an accuracy of 93%.

The SVM classification model using the RBF kernel function obtained better accuracy performance (87%) compared to its linear application (79%) and polynomial of degree 3 (78%), a result consistent with other experiments [**?**]. These results were better than those of the Decision Tree with the Gini criterion, Logistic Regression, and Naive Bayes algorithms, as found in previous experiments [15]. However, the results were lower than those of KNN considering the three closest neighbors (89%). An interesting observation is that the KNN classifier provided a very high recall measure, indicating that this method did not produce any false negatives in the three configurations tested. The Decision Tree model using the Gini criterion outperformed the entropy criterion, achieving the highest accuracy (83%) and precision (84%). Nevertheless, it showed relatively low performance, surpassing only Naive Bayes (76%) and logistic regression (80%).

Table 5 presents the accuracy results obtained in this study through the nine classification methods used. Additionally, the table includes the values of the highest accuracy obtained by the works cited in this article. The ten referenced articles in this work primarily employed classification techniques such as Logistic Regression, SVM, KNN, Decision Tree, and Random Forest. Considering all the references that worked with Decision Tree, KNN, and

Table 5: Comparison of results (accuracy)

Methods	Our	[16]	[20]	[25]	[15]	[2]	[5]	[12]	[29]	[17]
Naive Bayes	0.76									
Logistic Regression	0.8									
SVM with poly Kernel of degree 2	0.68									
SVM with poly Kernel of degree 3	0.78	0.97*		0.7673**	0.7421	0.7939		0.8701		0.7374
SVM linear	0.79									
SVM with rbf Kernel	0.87									
KNN with 3nn, distance metric with p=1	0.89									
KNN with 3nn, distance metric with p=2	0.88		0.77		0.6879		0.7474	0.8117	0.7190	0.7317
KNN with 5nn, distance metric with p=1	0.84									
KNN with 5nn, distance metric with p=2	0.84									
Decision Tree with gini criterion	0.83				0.7660***	0.7610				
Decision Tree with entropy criterion	0.81									
Random Forest, nestimators=100, Gini	0.90									
Random Forest, nestimators=100, Entropy	0.91									0.7174
Random Forest, nestimators=600, Gini	0.93									
Random Forest, nestimators=600, Entropy	0.92									
Perceptron with l1 penalty	0.67									
Perceptron with l2 penalty	0.63									
Perceptron with elastic net penalty	0.63									
MLP with (1000,1000)(Activation logistic)	0.82									
MLP with (1000,1000)(Activation identity)	0.81									
MLP with (1000,1000)(Activation tanh)	0.91									
MLP with (1000,1000)(Activation relu)	0.89									
Ensemble Methods: Voting Classifier	0.93									
Ensemble Methods: Stacking Classifier	0.95									

*SVM with PCA

**Soft Support Vectormachines (SSVM)

***Decision Tree-CART

Fonte: The authors

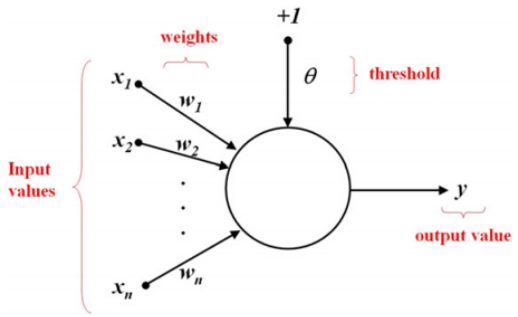

Figure 5: Single Neural Perceptron, adapted from [31]

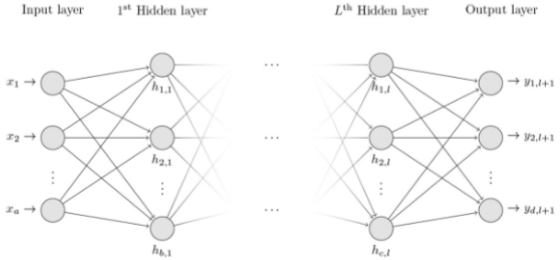

Figure 6: MLP example, adapted from [9]

Random Forest, our article demonstrated superior accuracy. Among the six works that utilized KNN, only one, [16], achieved a higher accuracy than the values presented in this article.

5 UNIDADES

5.1 Conclusions

In this article, we review the main contemporary techniques applicable to binary classification using the Pima Indian Database. We employ strategies to address missing values and handle imbalanced classes. To tackle missing data, a regression technique is employed for replacement. Additionally, for imbalanced data, the SMOTE technique is utilized [3]. We conduct a comparative analysis of various classifiers, including Naive-Bayes, Logistic Regression, SVM, KNN, Decision Tree, Random Forest, Perceptron, and Multilayer Perceptron, using different hyperparameters. When evaluating the models individually, the top performers are KNN with 3 nearest neighbors and Minkowski distance (89% accuracy), Random Forest with 600 estimators and Gini criterion (93% accuracy), and

MLP with 1000 hidden layers of 1000 neurons and tanh activation function (91% accuracy). Finally, these three best models are combined using ensemble methods, namely the voting classifier and stacking classifier (with a neural network in the final estimator), resulting in ensemble accuracies of 93% and 95%, respectively.

6 Source Code

All the code used in this paper can be accessed in the `https://github.com/gmichelcarvalho/SCC5948_Ciencia-de-dados/blob/master/Projeto.ipynb`.

References

[1] Agresti, A. (2007). *An Introduction to Categorical Data Analysis*. Second Edi ed. New Jersey: JohnWiley & Sons.

[2] Barale, M. S. and Shirke, D. T. (2016). Cascaded modeling for pima indian diabetes data. *International Journal of Computer Applications*, 139(11):1–4.

[3] Chawla, Nitesh. Bowyer, K. H. L. and Kegelmeyer, W. (2018). Synthetic minority over-sampling technique. *Elsevier*.

[4] Choubey, D. K., K. P. T. S. (2019). Performance evaluation of classification methods with pca and pso for diabetes. *Network Modeling Analysis in Health Informatics and Bioinformatics*.

[5] Christobel, Y. A. and Sivaprakasam, P. (2012). Improving the performance of k-nearest neighbor algorithm for the classification of diabetes dataset with missing values. *IJCET*, 3(3):155–167.

[6] Cortes, C. and Vapnik., V. (1995). Support vector machine. *Machine learning*, 20(3):273–297.

[7] Cury, D. P., Okamoto, R., Tumelero, S., and Madeira, M. C. (2011). Comparação de índice glicêmico e gordura corporal entre portadores de diabetes, praticantes e não praticantes de atividade física. *EFDeportes.com, Revista Digital. Buenos Aires*, 16(157).

[8] de., G. S. G. A. S. J. M. P. (1995). Utilização de estratificação e modelo de regressão logística na análise de dados de estudos caso-controle. *Revista Saúde Pública*, 29(4):283–289.

[9] et al, C. (2017). Multilayer perceptron architecture optimization using parallel computing techniques. *Plus one*.

[10] et al, Z. (2019). Regularized structured perceptron: A case study on chinese word segmentation, pos tagging and parsing. *aclweb*.

[11] Fukunaga, K. and Narendra, P. (1975). A branch and bound algorithm for computing k-nearest neighbors. *IEEE transactions on computers*, 100(7):750–753.

[12] Hashi, E. K., Zaman, S. U., and Hasan, R. (2017). Developing diabetes disease classification model using sequential forward selection algorithm. *Int. J. Comput. Appl.*, 180(5):1–6.

[13] Hayashi, Y. and Yukita, S. (2016). 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*.

[14] HOSMER, D. W. J.; LEMESHOW, S. S. R. X. (2013). *Applied Logistic Regression*. 3a. ed. New Jersey: [s.n.].

[15] Jahangeer, S., Zaman, M., Ahmed, M., and Ashraf, M. (2017). An empirical comparison of supervised classifiers for diabetic diagnosis. *International Journal of Advanced Research in Computer Science*, 8(1):311–315.

[16] Jhaldiyal, T. and Mishra, P. K. (2014). Analysis and prediction of diabetes mellitus using pca, rep and svm. *International Journal of Engineering and Technical Research (IJETR)*, 2(8):164–166.

[17] Kandhasamy, J. P. and Balamurali, S. (2015). Performance analysis of classifier models to predict diabetes mellitus. *Procedia Computer Science*, 47(C):45–51.

[18] Katti Faceli, e. a. (2011). *Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina* . Grupo Editorial Nacional.

[19] Kavakiotis, I., e. a. (2017). Machine learning and data mining methods in diabetes research. *Procedia Computer Science*.

[20] Kordos, M., Blachnik, M., and Strzempa, D. (2010). Do we need whatever more than k-nn? *Proceedings of the 10th International Conference on Artificial Intelligence and Soft Computing (Springer-Verlag)*, I:414–421.

[21] Kriještorac, M., Halilović, A., and Kevric, J. (2020). The impact of predictor variables for detection of diabetes mellitus type-2 for pima indians. *Springer*, 83.

[22] Kriještorac M., Halilović A., K. J. (2019). 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*.

[23] Masharani, U. and German, M. S. (2011). Pancreatic hormones and diabetes mellitus. In Gardner, D. G. and Shoback, D., editors, *Greenspan's basic & clinical endocrinology*, chapter 17. New York: McGraw-Hill Medical, 9th edition.

[24] Organization, W. H. (2020). Diabetes.

[25] Purnami, S. W., Embong, A., and Zain, J. M. (2009). A new smooth support vector machine and its applications in diabetes disease diagnosis. *Journal of Computer Science*, 5(12):1003–1008.

[26] R., H. and N., H. (2006). *Essential endocrinology and diabetes*. Wiley-Blackwell.

[27] Rosenblatt, F. (1957). The perceptron - a perceiving and recognizing automaton. *Cornell Aeronautical Laboratory.*

[28] Shankaracharya, D. O., Samanta, S., and Vidyarthi, A. S. (2010). Computational intelligence in early diabetes diagnosis: a review. *Rev. Diabet. Stud.*, 7(9):252–262.

[29] Ster, B. and Dobnikar, A. (1996). Neural networks in medical diagnosis: comparison with other methods. *Proceedings of the International Conference on Engineering Applications with Neural Networks*, page 427–430.

[30] Tigga, N. P. and Garg, S. (2020). Prediction of type 2 diabetes using machine learning classification methods. *Procedia Computer Science*.

[31] Vanneschi, L. and Clastelli, M. (2018). Multilayer perceptron. *Elsevier*.