# Conference Paper Title

Brewton Morais
*DETI*
*Universidade Federal do Ceará*
Fortaleza, Brazil
brewtonlmorais@gmail.com

Lucas Abdalah
*DETI*
*Universidade Federal do Ceará*
Fortaleza, Brazil
lucasabdalah@alu.ufc.br

***Abstract*—An analysis of classification models was done for a set of data with informations related to the request and approvals for research and academic activities. Using MATLAB 2022a, we trained different classification models, statistical and decision trees, such as Logistic Regression, KNN, LDA, CART, etc, making use of their respective confusion matrix as the results demonstrations. One of the goals these tests is to investigate if there are linear relationship between the provided predictors of the data base, in order to optimize the model that could be used by an university or a governamental institution. Finally, we perform a comparison of the results and of the decision making about which model we understand to be the most efficient to the problem.**

***Index Terms*—Classification model, logistic regression, grand application, KNN.**

## I. Introduction

This command is used for a strong suggestion.

This command is used for minor changes suggestion.

Classification models are a class of mathematical models constantly used in problems of assimilating observations of certain events to certain categories that define the problem. Nowadays, these models are considered tools of fundamental importance in the construction of Deep Learning and Machine Learning algorithms. To begin, it's important to evidence the main existing difference between this new class of models and the class of regression models which is the prediction of a qualitative variable instead of a quantitative one. This new class of tools present various practical applications, such as the development of a detection spam filter for emails based on the sender and on the content of the message, the development of classification techniques of a cell belonging to tumors, as beningn or malignant and on the development of a model of credit release for financing.

In addition to these pure classification models applications, there are mixed applications techniques that combine Data Mining techniques with some other types of models to perform a prediction. Some examples of models that use Data Mining practices to improve their results are addressed by Sidropoulos Add reference, such as Web Mining/Search tensor models and Brain Data Analysis.

Meanwhile, in the work developed in this paper, some of the most used routines in the development of classification models were approached, such as Support Vector Machine (SVM), K-Neirest Neighbors (KNN) [1] and Classification and Regression Tree (CART), to train and test a funding model that will separate observations into one of two available groups: Positive Founding and Negative Founding.

KNN (V-B) uses the concept of proximity or distance to make classifications of an individual data point, based on the other points surrounding it. Simply, we assume that data points close to one another up until some threshold belong to the same class. SVM (V-C) is a set of supervised learning used for classification. One of its advantages is its efficience in high dimensional spaces, even when the number of predictors is greater than the number of samples. However, unlike Logistic Regression, it doesn't directly provides probability values. Decision Trees (V-D) are a non-parametric method for classification and regression. In short, the way it works is by efficiently learning decision rules based on the data set, that will predict the final output as being of a class, or another. It's important to state that this method is quite efficient with $N > 2$ classes. Intuitevely, it simply checks some conditions one after another, which are the decision rules, and link the response of those conditions to a certain class at the final step. One good advantage of this method is that it doesn't require much data pre-processing, so it's not necessary to create dummy variables or to normalize the data.

## II. Method

### A. Data set

The data used for the construction of the predictive models consists of 8708 samples of different requests for funding from universities around the world, to finance research, with the outcome being the success or failure of the request. The data set contains samples from the years 2005 to 2008, with a total of 1882 predictors (independent variables). 6633 samples from 2005 to 2007 and 1552 from 2008 were used for model training, and the remaining 518 samples from 2008 were used for testing the obtained model. Predictors can be separeted between continuous, such as the number of successes and failures passed by the "chief investigator", and categorical ones, such as the monetary value of the grant, divided into 17 groups of increasing amounts, and the month of application.

### B. Pre-processing

Initially, the first step is verifying the data skewness, in case there is a strong tendence to the left or to the right, an adequate transformation would be applied in order to remove the skewness. The next step is to scale and center the data

around the mean. It's done so since different predictors can have different scales, and if they're not normalized, models sensitive to the variance would be affected negatively, making it biased to those predictors with the highest values.

Then, what we should do is to verify which predictors have actual importance to the model construction, that is, which of them have a stronger say for the final prediction. We can study this by analysing their correlation. Those with a correlation larger than 0.99, with zero variance or sparse, that is, that have lots of zero values as data, were removed.

Then, the final approach is to verify the linearity of the predictors together with the output. This step is essential, and the reason for this is, once we have analysed this aspect, we can infer if using a linear model is the adequate way of resolving this problem. For example, if there are too much predictos with non-linear relationship with the final output, it makes no sense to insist in linear prediction models.

### C. Cross-Validation

Cross-validation consists in a validation technique used to validate the model with the test set, usually taking into account the model flexibility and the mean squared error (MSE). Shortly, it divides the data set into $k$ distinct subsets of size as equal as possible. From these groups, one of them is put aside to be used as validation set, while the model is trained based on the remaining $k-1$ subsets. Once the model is trained, the first removed subset is used as validation as previously stated. Then, the removed subset is restored to the principal set, and the following subset is put aside to perform the same procedure until all of the $k$ subgroups are all used as validation set. This approach improves the model capability of generalization, once it's trained with all the data at dispose, it also makes the error estimation more robust.

This strategy generally serves to indicate which models have a better prediction capability on the test set, since it enables the comparison between the error levels and the variance generated.

It's important to state that if a $k$ is chosen such as it's too small, e.g, $k = 2$ (two subgroups) or too large ($k$ =sample size), we are going to have, respectively, a strongly biased model because we let a lot of data outside the training step, and a potential overfitting issue due the high model complexity, ocurring the model to have a high variance. Thus, to mitigate both of the effects, normally $k = 5, 10$ is employed, since they present an acceptable level of bias and variance.

## III. Model validation performance

One of the most used metrics to measure the performance a classification model is the Receiver Operating Characteristic curve (ROC) and the Area Under the Curve (AUC) correct initials?. The ROC curve is traced in a graphic with the positive ratio in the y-axis and the negative ratio in the x-axis, and each point of the curve is computed varying the classification limiar. Decreasing this limiar makes the model classify more items as positive, increasing the true positive and false positive. Increasing this limiar, causes the inverse effect.

After calculating the ROC curve we can find the area under it. The area under the ROC curve represents how well the model divides the two classes, as close the AUC value is from 1, better the model. However, this kind of analysis shouldn't be applied alone to validate a model's performance, since there is an information loss in the construction of the graphic. Then, the dispersion table, which contains in its principal diagonal the number of true positives and true negatives, and in its secondary diagnoal the number of false positives and false negatives, becomes an interesting analysis complement to the ROC curve.

## IV. Linear Methods

### A. Logistic Regression (LR)

It is statistical model used to determine the probability of an event. Therefora, its values are probabilities, which belong to the $(0, 1)$ interval. The model is defined as:

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}, \quad 0 \leq p(X) \leq 1 \tag{1}$$

This logistic model will aways produce a S-shaped curve, regardless of the value of $X$, getting a relatively precise prediction. After some mathematical astuces, it's possible to come up with the following equations that model the method.

$$\frac{p(X)}{1 - p(X)} = e^{\beta_0+\beta_1 X} \tag{2}$$

$$\log\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \beta_1 X \tag{3}$$

The right term of (2) is called Odds, then the right term of (3) is called log-odds or logistic. The Odds ratio represents the effects of predictor X, on the likelihood that an event will happen.

To adjust the model's parameters would be necessary to use the Maximum-Likelihood technique to perform an estimation of them. However, it's also possible to make use of the Least Squares for the adjust, as well as in the case of the coefficients in a linear regression.

Put Table 1 here

### B. Linear Discriminant Analysis

Instead of directly estimating $P(Y|X)$, a model with the following characteristics will be developed:

- Modeling the distribution of predictors of X separately in each class Y.
- Baye's Theorem to estimate $P(Y = K|X = x)$.
- Normal distribution to describe each class.

Following from these informations, we initiate the model's development directly from the Baye's Theorem:

$$P(Y = k|X = x) = p_k(X) \tag{4}$$

$$p_k(X) = \frac{P(X = x|Y = k)P(Y = k)}{P(X = x)} \tag{5}$$

$$p_k(X) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)} \tag{6}$$

Where $f_k(x)$ represents the probability density function (pdf) of the r.a X of an observation belonging to class K. Thus, instead of directly computing $p_k(X)$ it's possible to simply estimate $\pi_k(X)$ and $f_k(X)$. Then, assuming that the number of predictors is unitary, we can make some affirmations about the form of $f_k(x)$ in order to move on with the LDA method:

$$f_k(x) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{\frac{-(x-\mu_k)^2}{2\sigma_k^2}} \tag{7}$$

Besides, we assume $\sigma_1^2 = ... = \sigma_k^2$. Therefore, it's possible to write $p_k(X)$ as the following:

$$p_k(X) = \frac{\pi_k \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{\frac{-(x-\mu_k)^2}{2\sigma_k^2}}}{\sum_{l=1}^{K} \frac{1}{\sqrt{2\pi\sigma_l^2}} e^{\frac{-(x-\mu_l)^2}{2\sigma_l^2}}} \tag{8}$$

After some algebraic manipulations, it's possible to conclude that classifying one observation to a given class is equivalent to classify one observation to a given class such that the the linear discriminant function $\sigma_k(x)$ is larger:

$$\sigma_x = x\frac{\mu_x}{\sigma^2} - \frac{u_k^2}{2\sigma^2} + \log(\pi_k) \tag{9}$$

However, in practical situations, it's not always possible to know the parameter's values, then LDA approximates the Bayes classifier by the following expressions:

$$\hat{\mu_k} = \frac{1}{n_k} \sum_{i=k}^{K} x_i \tag{10}$$

$$\hat{\sigma^2} = \frac{1}{n-K} \sum_{i=k}^{K} \sum_{i=k}^{K} (x_i - \hat{\mu_k})^2 \tag{11}$$

$$\pi_k = \frac{n_k}{n} \tag{12}$$

Put Table 2 here

## V. Non-Linear Methods

### A. Quadratic Discriminant Analysis (QDA)

The QDA method is a non-linear variant of the LDA. The main difference between QDA and LDA lies in the fact that the covariance matrix of each class is different onr from another:

$$X \sim \mathcal{N}(u_k, \sum_k) \tag{13}$$

The discriminant function for this method is, after some algebraic astuces:

$$\sigma_k(x) = -\frac{1}{2}(x-\mu_k)^T \sum_{k}^{-1}(x-\mu_k) - \frac{1}{2}\log|\sum_k| + \log(\pi_k) \tag{14}$$

$$\begin{aligned}\sigma_k(x) = &-\frac{1}{2}x^T \sum_{k}^{-1} x + x^T \sum_{k}^{-1} \mu_k - \frac{1}{2}\mu_k^T \sum_{k}^{-1} \mu_k - \\ &- \frac{1}{2}\log|\sum_k| + \log(\pi_k)\end{aligned} \tag{15}$$

Thus, the QDA may be summarized as a way of computing $\sum_k$, $\mu_k$ and $\pi_k$ in order to use the discriminant equation in the classification of a X observation into a class in which the discriminant has the larger absolute value.

Therefore, the QDA method is mainly recommended when there is a sufficiently large data set so the statements about the variance don't be a need or when it is not possible to sustain the statement abour the unicity of the covariance matrix. Hence, differently of the LDA method, the decision region of QDA is described by a non-linear curve.

### B. K-Nearest Neighbors (KNN)

Generally in practice, informations about the conditional distribution of Y given X are not available. That being the case, the Baye's classification method works only as a comparison tool to other more easily applicable practices. From this need of developing methods that don't rely on previous statements about the form of the boundary of decision, the KNN routine takes place.

Shortly, KNN is a model such that its result is define by a "voting", where each "vote", is the amount of K samples closer of elements surrounding the analysed point which belongs to a certain class. The class having the larger amount of points, or votes, wins.

Something important to define in KNN methods is how the distances would be measured between the samples, given that the classifiers are not in lenght units. Starting from this premise, considering the vector $\mathbf{x} = [p_{x_1}, ..., p_{x_n}]$, and $\mathbf{n} = [p_{n_1}, ..., p_{n_n}]$, where $\mathbf{x}$ the vector that represents the sample to be classified and $\mathbf{v}$ is the neighbor vector to be computed the distance. Therefore, the most traditional way of computing the distance, is using the Euclidian Distance:

$$d = \sqrt{\langle (x-v), (x-v) \rangle} \tag{16}$$

Of course there are a lot of other ways of computing the distance between the samples. In the training phase, the best results were obtained by using the Spearman Distance computation method, where the vector measures are taken into account and represented by $\bar{\cdot}$, and it is given as:

$$d = 1 - \frac{\langle (x-\overline{x}), (v-\overline{v}) \rangle}{\langle \sqrt{\langle (x-\overline{x}), (v-\overline{v}) \rangle}, \sqrt{\langle (x-\overline{x}), (v-\overline{v}) \rangle} \rangle} \tag{17}$$

This distance was used in a test with 50 neighbors, together with the use of weights that considered the inverse of the

square distance, and in order to not harm its performance, the data weren't standardized, being the second more well succeeded than the previous and also than the pure distance. However, by means of reference, the results depicted in table 3 were computed without weights and with Euclidian Distance.

Since KNN does not classify the samples by probabilities, one way of making it possible is to make the probabilities related to votes of each sample. This is done by the following equation. which defines the conditional probabilities, with $K \in \mathbb{Z}^+$ referent to the neighbors and an observation $x_0$ in a set $N_0$ and the function $I(\cdot)$ returns 1 if the sample $y_i \neq j$ and 0 otherwise.

$$P(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in N_0} I(y_i = j) \tag{18}$$

Finally, to conclude the KNN algorithm the Baye's Rule must be applied to classify the observations $x_0$ into the class with the highest probability of occurence. It's interesing to notice that for $K$ close to 1, the decision boundary is extremely flexible, corresponding to a classifier with low polarization (low bias) and with high variance. Meanwhile, for a large K, the exact opposite is true.

Add table 3

Add table 4

### C. Support Vector Machine (SVM)

In its simplest form, SVM separates points of two different classes using a single hyperplane with the Statistical Learning Technique, both developed by Vapnik (1995) and (1998) add reference. In the more complex versions, an hyperplane is built in $\mathbb{R}^n$, for a $n > 2$, for the class split, the increase of $n$ leaves the problem with a non-linear solution. In general, it's used the expression in (19) to define the decision boundary, i.e, the hyperplane for the separation of classes in $\mathbb{R}^2$.

$$D(u) = \beta_0 + \sum_{j=1}^{P} \beta_j \mu_j = \beta_0 + \sum_{i=1}^{n} y_i \alpha_i x_i' u, \qquad \alpha_i \geq 0 \tag{19}$$

Add table 5

### D. Decision Tree

This method is quite robust and it comes from a simple principle: divide the approach into levels and attributes to build a cost function. To facilitate the visualization, it's interesing to use a binary tree. In general, the approach is given by decision levels, and for each level an attribute is chosen such that it better splits a data set according to the class.

Add figure 1

For each new chosen attribute, new branches are created for the values present in the list of all possible attributes values. The way this is implemented depends on the algorithm used. In our work, we opted for the CART algorithm. The attributes choice in this model is made based on the Gini Criterion, which deals with node impurity, defined in Add Reference here by Breiman (1984) as:

$$\sum_{j \neq i} p(i|t) p(j|t) \tag{20}$$

In short, we can assume that the criterion associates an object randomly selected from a node to the class $i$ with probability $p(i|t)$. Meanwhile, $p(j|t)$ refers to the estimated error probability that the item belongs to class $j$, such that the $p(j|t)$ is the Gini Indice.

The Gini Criterion works with one class at a time, aiming to separate the more frequent class objects from the remaining objects in each test.

The well detailed explanation it's really important, since one of the factors that may influence the precision and the time consumed to classify is the number of nodes present. Look at the tables (6) and (7) cite tables, there are represented the results with trees of 10 and 100 nodes, respectively. The number of nodes didn't have much influence in the confusion matrix.

Add table 6

Add table 7

## VI. Results and Discussion

### A. Training results

During the training phase of all the models proposed, we applied the cross-validation with $k = 10$ subgroups, allowing protection against overfitting. To verify the results on the training set, we look at figure (2) and table (8). From the figure, we can conclude that non-linear methods, such as SVM and the Decision Trees with 100 nodes, have better performance when compared to simpler methods, such as Decision Trees with 10 nodes and Linear Methods.

However, this pattern does not hold on the test set, since the linear models obtained a quite good generalization capacity, making them as efficient as the non-linear models.

### B. Conclusion

Among the test results obtained, there were no relevant differences between the results of the models, with an exception of KNN, which had the worst result, but still, not bad. A probable cause of this performance of KNN, it's that there are probably a data skewness, since the "nature" of KNN, will preferentially classify the data located at the side containing more elements, the evidence to it is in the table (4), since there is a good classification performance in one of the classes, what suggests that this class is more numerous, distorcing the remaining class classification.

This similarity on the results between linear and non-linear methods, means that the classes "success" and "failure" are linearly or approximately linearly separable. Therefore, it favors the choice of simplest models of less computational cost. According to the generated test, the best model was the Quadratic SVM. On the other hand, that performance was only superior in one hit in comparison with the LDA model. The latter can be trained with a lot less time and generate equivalent results.

Like KNN, other models had more considerable errors in one of the classes. The Logistic Regression had a correct rate smaller in the class ”success” than in the ”failure” class. Meanwhile, in the decision trees algorithms, the situation is inverted. The algorithms presenting a better trade-off in the class classification were the LDA and the Quadratic SVM, with a difference around two percentual points between the correct classification rates.

Having raised these considerations, we conclude that the model with the best trade-off is the LDA, as its results with respect to corrected classified rates and time of training are better than the results of other traditional methods. Its corrected classified rate was not discrepant between classes, besides being equiparable to the non-linear method with the best performance on the test set. The only caution would be if it was necessary to often remodel the methods as more data were to be added. To this specific situation, the Decision Tree with 10 nodes would be recommended, in order to be more agile in the training phase with a large data set.

## REFERENCES

[1] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2022. [Online]. Available: https://www.R-project.org/