Grant Application Classification Using Linear and Nonlinear Models

Emmanuel Victor Barbosa Sampaio
Department of Teleinformatics Engineering
Federal University of Ceará
Fortaleza, Brazil
emmanuelsampaio@alu.ufc.br

Carlos Alfredo Cordeiro de Vasconcelos Filho
Department of Teleinformatics Engineering
Federal University of Ceará
Fortaleza, Brazil
carlosalfredo@alu.ufc.br

***Abstract*—The problem of verify if a grant application going to be accepted or not represent a challenge for universities around world, which are interested in create a solution to identify the students that going to be accepted by a grant program, resulting in more resources coming for the university. In this context, a possible solution is the application of classification models, that are able to predict the student result in the application program. Given that background, on this work we going to discuss the application of different types of classification models to predict grant application result.**

***Index Terms*—Data analysis, Classification models.**

I. Introduction

An university maybe interest in developed tools to identify the students that has potential to be accepted by programs of grants, in order to optimize the process of select students to be applicant to this kind of programs. In this context, the development of a tool that permits to predict an application as successful or not can represent a useful solution.

Given the previous situation, a possible approach is the use of a set of supervised learning models called classification models. These type of models has as goal perform the prediction of categorical outcome Y, given a set of features X. From [1] these models can be understood as functions that takes a feature set X, which is already divided into classes and perform a prediction of which class an observations in the feature space participate. This can also can be understood as divide the feature space into subspaces, each one representing one class.

In the grants application classification, the categorical outcome is divided between successful and unsuccessful applications and the feature space is composed by the information of the applicant. In this work, in order to contruct the model we going to use a dataset from a Keagle competition, that is also explained in the Chapter 11 from [3]. To develop the model we have access to old application results to use them as training and to test we have a separate class of applications from a different time. For each application there is a set of data related to the applicant and also the application result.

This work going to verify the results of using different classification models approaches in order to identify the models capacity to be assertive in predict successful application at the same time it not allow an unsuccessful application be consider as successful.

The outline of this paper is organized as follows: Section II presents the theoretical background behind the classification modes that were used as well as the methods to evaluate them, Section III presents a description of the dataset variables. Section IV presents a discussion about the results of the models application. We conclude this work in Section V with a final discussion about the results.

II. Analysis Background

From [1] as the value that we aim to predict takes a value in a discrete set we can divide the feature space into subset that are associated to one of the possible values of the prediction. This division is represented by the construction of boundaries, called decision boundaries. When these boundaries are associated with linear functions, we call the classification method a linear method, when the boundaries are defined by nonlinear function we call the classification method nonlinear method.

Against this background, the development of linear and nonlinear models going to verse on how well they split the input space into the different classes, resulting in some models returning smoothly division between the observations in contrast other models going to give us a rough division between them. The rough and smoothly division can be associated with model complexity as well as the model be prone to have overfitting or underfitting.

From the previous consideration, the following subsection going to discuss the linear models, non linear models and how we perform the evaluation of these models.

A. Linear Models for classification

A simple model that was used to perform the prediction was the logistic regression (LR). From [3] the LR uses the fact that the minimization of the sum of the squared residuals in the linear regression also produces maximum likelihood estimates of the parameters when the model residual error follows a normal distribution. This maximum likelihood parameter estimation allow us to make assumptions about the probability distribution of the data. Based on the dataset we generate a model that predicts the probability of an input to belong to all possible classifications, the one with the biggest probability is chosen as the prediction.

Because probability can only go from 0 to 1 we can not use a normal linear regression because we can not guarantee that the result will be in this interval. Therefore we use the concept of odd. If the probability of an event occurring is p the odd of this event happening is $\frac{p}{(1-p)}$. The LR model uses the natural logarithm of the odds as the result of the linear function.

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 x_1 + ... + \beta_k x_k \tag{1}$$

Where k is the number of predictors. Since the odds will have a positive value regardless of the result of the linear function, there is no concern about the range of values that it may produce. Now, we isolate the $p(X)$:

$$p(X) = \frac{1}{1+\exp(-(\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k))} \tag{2}$$

This nonlinear model constrains the probability estimates between 0 and 1 and, from [3], produces linear class boundaries, unless the predictors used in the model are nonlinear versions of the data.

While the LR is not as efficient as the other classification models, the coefficients of the linear function allow us to make inferential statements about the terms of the model. For example, predictors with bigger coefficients can be identified as more relevant to the classification. Furthermore, the performance of this model can be improved by swapping predictors with better representations of themselves (e.g. swapping x for x^2, e^x or $\log x$).

Another model that was used to perform the prediction was the Linear Discriminant Analysis (LDA). From [2] this model verifies the distribution of the predictors on each class and uses Bayes theorem to perform the prediction of the posterior probability of a observation participate in a class. In this domain, LDA uses that the features X are distributed following a multivariate Gaussian distribution $X \sim \mathcal{N}(\sum, \mu_k)$, where $\sum$ is the covariance matrix of X and μ_k is the mean of the X considering the observation of the class k. The density function $f(x)$ is presented in [2]. Using $f(x)$ in the Bayes formula it is possible to define a function $\delta(x|\sum, \mu_k, \pi_k)$, using the function we classify x in the class k.

B. Nonlinear Models for classification

The nonlinear models differentiate from the previous linear models due to the construction of a decision boundary that is more flexible than the boundary of linear model. This flexibility can cause a model that better divide the feature space set, but as the flexibility increases, the nonlinear model can overffiting, so we need to carefully define the model parameter, for example using cross validation approaches.

Given the nonlinear context, a simple model that can be used is called K-Nearest neighbours. In this model, we verify the K nearest points of a observation in the test set, to do it we use some metrics, and use their labels information to define the class of the observed point. More precisely, from [2], the calculus of the class of an observation, given k as the number of neighbours and $\mathcal{N}_k(x)$ as the set of k nearest observations (x, y) of this observation is defined as follows:

$$\hat{Y}(x) = \frac{1}{k}\sum_{x_i \in \mathcal{N}(x)} y_i \tag{3}$$

To verify if a observation is in $N(x)$ we can use different metrics such as the k lowest values returning from the calculus of the euclidean distance. We can set different number of values for k. In this domain, we can use cross validation in order to define the optimal values for k also verify how this value changes as we modify the metric used to quantify the distances between the observed point and the tranning set point. This tuning process reduce the risk of overffting causing by the increasing in model flexibility due the metric and the number of neighbours as well as the risk of underfitting.

The LDA presented in the last section has as limitation consider the value of the covariance matrix of X in the calculus of $\delta(x|\sum, \mu_k, \pi_k)$, which leads a inflexibility since it not differentiate the distribution of the different classes. For this reason, we can apply a model that consider $X \sim \mathcal{N}(\sum_k, \mu_k)$, where $\sum_k$ is the covariance matrix of X when we consider the observations from the class k. Using this new assuption we can perform the calculus of a function $\delta(x|\sum_k, \mu_k, \pi_k)$, in this approach we consider the class of the observation based on the same condition of the LDA. This model is called Quadratic Discriminant Analysis (QDA), its nonlinearity reduces the bias associated with LDA prediction, otherwise when we consider a large number of predictors, the variance of the model increase.

Another nonlinear Model is the Support Vector Machines (SVM). From [3] if two classes of samples are completely separable, there are an infinite number of linear boundaries that perfectly classify the data. Because of this if we evaluate the boundaries by accuracy there will be an infinite number of solutions. To solve this problem we define a new metric called margin that represents the distance between the classification boundary and the closest training set point. The chosen boundary is the one that maximizes the margin. Suppose we have a two-class problem with class 1 samples having a value of 1 and class 2 samples having a value of 2. We create a function $D(x)$ for witch if $D(x) > 0$, x belong do class 1, otherwise x belong to class 2. For an unknown sample u we can write $D(u)$ as:

$$D(u) = \beta_0 + \beta' u = \beta_0 + \sum_{j=1}^{P} \beta_j u_j \tag{4}$$

Rewriting D(u) to be in terms of each data set point:

$$D(u) = \beta_0 + \sum_{i=1}^{n} y_i \alpha_i x_i' u_j \tag{5}$$

From [3] in the completely separable case the α parameters are zero to all points except the ones on the margin. Because of this, the points that are on the margin are the only ones that matter to the prediction equation and are referred as the support vectors.

We can create a non-linear boundary by substituting the linear cross product by a kernel function.

$$D(u) = \beta_0 + \sum_{i=1}^{n} y_i \alpha_i K(x_i', u_j) \tag{6}$$

Where K(:,:) is a function of two vectors.

C. Models Evaluation

In the context of classification models a error that can occur is the miss classification of an observation which means put a observation from a class into other class. A initial metric that we can consider is the accuracy of a model that can be represented as the number of correct classifications divided by the number of observations in the dataset.

The accuracy metric gives as on the number of right classifications of the model, but in a binary context for example we can not determine the number of elements from one class that was wrongly classified in other. To define a better description, we can study the number of right and wrong classifications associated with each class, which means for a class k verify the number of observations from class k that were classified as k and the number of observations from other classes that were classified as class k. In a binary classification context, the literature [2, 3, 1] uses a biologically notation, they call positive and the other negative. The number of observations that are correctly classified as positive are called true positive, and as negative are called true negative. When we perform a wrong classification of a positive class, we call false positive, and a wrong classification of a negative class, we call false negative. These four values, can be placed in a matrix that is called confusion matrix, where the principal diagonal is fitted by the true positive and true negative and the other diagonals by the other metrics, considering each line related to a class.

Using the information of the confusion matrix and having a specific event as positive an other as negative we can define some metrics to evaluate the model. In this work domain we going to consider as positive the successful application and as negative the unsuccessful application. So we can use two metrics precision and recall. Recall is the number of right predictions of successful applications divided by the number of real truly successful application, and precision is the number of right predictions of successful applications divided by the number of predictions of successful applications.

D. Using Classification Models For Feature Selection

Classification trees has as goal define a set of rules that are capable to determine set of points that aggregate the highest number of points of one determined class. In this context, the definition of the rules are based on the values of feature variables and the optimization process looks to reduce the number of wrong region placement, which means avoid to produce regions where the number the missclassification is high. In order to metric this a useful metric is the Gini index, that uses the proportion of observations of a class in a region p_k and measure the total variance for the classes using:

$$G = \sum_{i=1}^{K} p_k(1 - p_k) \tag{7}$$

From [2] Ginni is small when all p_k are close to 1 or 0, which means have a concentration of least classes as possible per region, so we aim to reduce for each region its Ginni index. Since this regions are defined by the splitting, Ginni can be used as metric to qualify a splitting.

Given the notion of Ginni index, the feature importance is associated with the reduction of the Ginni index calculated to with a specific splitting caused when we used that feature to control the splitting. We can use this feature importance notion to identify features that are most useful be used in other classification models in order to reduce the dimension of the feature space.

III. Exploratory Analysis Of the data

The dataset contains information about 8190 applications for grants in the traning dataset. From all them 4387 applications are unsuccessful and 3803 are successful. In this domain each contains 1784 different information about the applicant that can be used as predictors for the model. For this activity we used a reduced set of predictors defined by [3], that contains only 252 features associated with the applicant, from [3] this reduced set was taken due to the presence of missing values and to high correlation between features, this reduced set does not suffer was already pre-processed to reduce the impact of the points.

The majority of these features coming from encoding a categorical non numerical data, which means the translation of a non numerical categorical data into a numerical categorical data. In this dataset, the encoding process created binary features, that assumes 1 when the feature is associated with the application and 0 when the feature is not associated.

A. Features Relation With The Outcome Class.

In this reduced set we started by verify the impact of the month in the application result. From 1 we can verify that the in January and August, we have a clear distinction between the number of accepted and not accepted applications. Given these two months, we can verify that in January, from all 525 applications only 45 were unsuccessful. Contrastingly, in August the number of unaccepted applications are represents a great percentage of all applications in that month 1013 from 1377 were unaccepted.

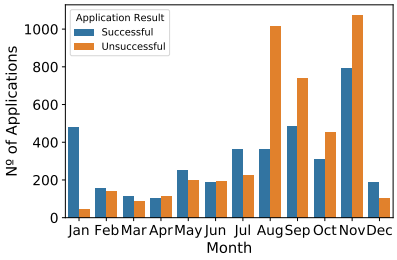


Fig. 1. The Application result per months in 2008. We can visualize that the January is a month were the number of successful application is far from the result. And in August, the majority of the applications are falling.

Other analysis was the result per day of application, we verified that applications that occurs on Sunday had more chances to be acceptable compare to the other day of the week. In 2 we can visualize this as well as the number of application at Friday and Saturday are greater than applications in the other days, and here the number of Unsuccessful applications is greater than the number of successful ones.

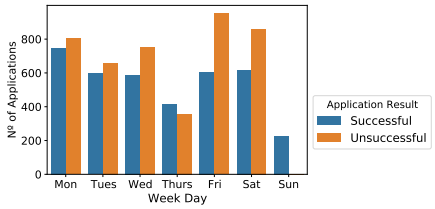


Fig. 2. The difference between the days application results.

We also verified that the band associated with each application. We verified that the Unknown bands are related with the majority of the application as well as the highest value of odd we compare to the other bands, this is possible to visualize when we compare the number of successful and unsuccessful application per band of contract, that are presented in 3 the unknown contract is associated with most of the application and it is also related to the greatest number of unsuccessful results. Considering the other bands, A has the greatest ratio of successful application.

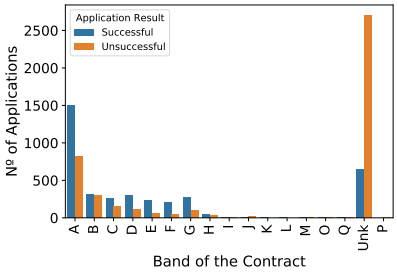


Fig. 3. The difference between the application results for each band contract.

We also verified how the Number of successful application changes given the categorization of the sponsor. From Fig.4, the sponsor associated with the label 10A is associated with the greatest percentage of applications in this dataset, but the greatest ratio between successful and unsuccessful is related to 30B and the greatest ratio between successful and unsuccessful is related to 50B.

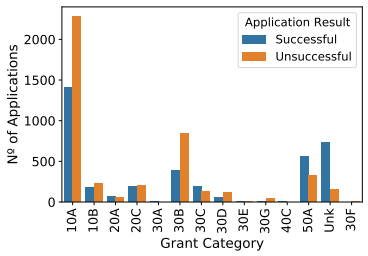
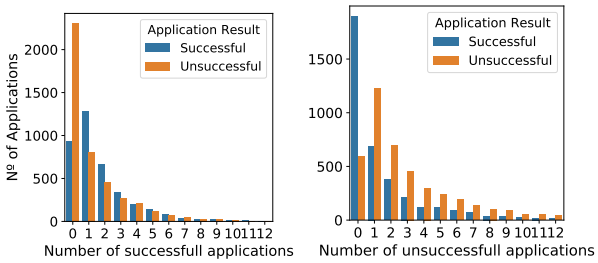


Fig. 4. The difference between grant category.

Another variables that we visualize was the variables associated with the number of previous successful and unsuccessful application of the applicant. In Fig.5 it is presented a bar chart that shows the number of successful and unsuccessful applications for each number of previous positive or negative application. From the Fig. 5b as the number of previous unsuccessful applications increase the result tends to be unsuccessful. From Fig.5a it is possible to visualize as the number of previous successful application increase the result of the applications tends to be successful.



(a) (b)

Fig. 5. In 5a the number of successful and unsuccessful applications considering the number of previous successful results. In 5b the number of successful and unsuccessful results considering the number of previous unsuccessful result.

IV. Classification of the Applications

In this section we going to use different classification models to perform the classification of the application results. In this context, we going to apply both linear and nonlinear models and evaluating them using the metrics showed in II. Given the evaluation we going to be able to compare the models verifying how their right and miss classifications are situated into the two evaluated classes.

Before performing the model creating a important process is to define the features, from Section III we already understand that there exists features in the feature space that are highly associated with successful and unsuccessful application while order features not have the same impact in the grant application result. So we can perform feature selection in order to get a lower set of the original features in the feature space.

A. Feature Selection

In order to reduce the number of variables available in the dataset we aim to apply the calculus of the feature importance. In order to improve the quality of the feature importance we going to use a ensemble method that groups different classification trees called random forest, from [2] the feature importance going to be calculate as the total amount that the Gini index is decreased by splits over a given feature, averaged over all trees.

To perform this calculus we used the training dataset and a model with 1000 trees that has as maximum deep a value of 5. From the feature importance analysis we select to use in the classification models 8 variables that are related to the top 8 highest values of feature importance for the Random Forest model tested. In Figure 6 we show the resultant feature importance for 20 variables of the dataset. It is interesting to visualize that the odds analysis in the section III highlighted variables that have great feature importance for the model such as Unknown Brand value in the Brand analysis, in Fig. 3 and January in Fig.1.

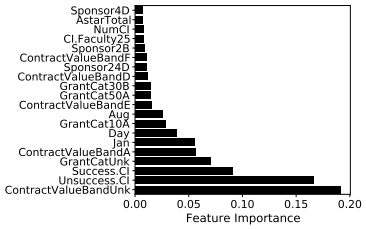


Fig. 6. Feature importance measured using Ginne index reduction caused by each feature divided by the number of trees in the random forest model.

B. Logistic Regression

After define the feature set we them started to use models to perform the prediction. The first tested model was the logistic regression model which was explained in II. We started by using a non penalized model, which is represented by (2). Using the mentioned variables we found a model which has a accuracy of 0.83. To get a more deeply understand about the model performance, we verified the confusion matrix associated with the predictions, that is presented in Fig. 7. We observed that the a value of recall of 0.87 that is greater than the value of precision that is 0.77, this occurs because the model has more unsuccessful application that were predicted to be successful than successful that were predicted as unsuccessful

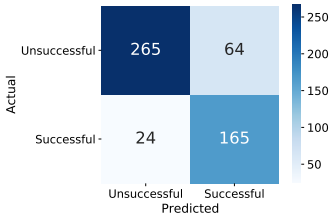


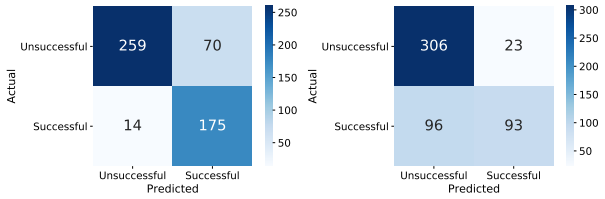
Fig. 7. Confusion matrix from the logistic regression.

C. LDA and QDA

Another technique that we can apply to perform the classification is the discriminant analysis, considering both linear and quadratic models. For both models we going to use the same set of features defined in the feature selection subsection.

Using LDA we defined a model that reaches a accuracy of 0.837. Calculating the confusion matrix, Fig.8a. We identify that LDA model reduced the number of successful classifications that were classified as unsuccessful compare to logistic regression confusion matrix presented in 7, which cause the recall increase to 0.91. In order hand the precision reduced, for LDA 0.71, because the number of truly unsuccessful that were classified as successful increase.

Different from the LDA, QDA returned an accuracy 0.77 in the test set. When we verified the confusion matrix for the QDA, Fig.8b, we verified that for this model we increase the precision, for QDA is 0.80, because we reduce in the number of truly unsuccessful that were consider to be successful, in other hand the recall decreases considerably to 0.49, because due to the classification of truly successful application as unsuccessful.



(a) (b)

Fig. 8. In 8a the confusion matrix of the LDA. In 8b the confusion matrix of QDA.

D. K Nearest Neighbours

We also verified the behavior of K Nearest Neighbours model, for different number of neighbours and for different distance metrics. To define the neighbours number we used a K-fold cross validation approach, explained in Chapter 5 from [2], considering a k fold of size 10. We calculate the average accuracy of cross validation for different numbers of the neighbours using three different distance metrics, the Euclidean distance, Manhattan distance, sum of the distance on each component of the two points, and Chebyshev distance, it is the maximum value of the modules of difference between the points coordinates. The result is presented in Fig.9. From the analysis of the cross validation, the best result was found using Manhattan distance with a number of 9 neighbours.

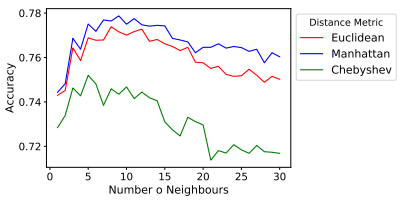


Fig. 9. The accuracy changing as we modify the number of neighbours.

Using the model with the best result, we verified the model performance in the test set. The accuracy of the model was 0.79. The confusion matrix found is presented in Fig.10. When we verify the precision and recall we verify that precision is greater than the recall, 0.72 of precision and 0.70 of recall. When we compare to other models, we verify that this model different from LDA and Linear Regression, reduces the number of truly unsuccessful that are classified as successful.

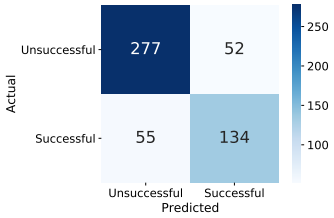


Fig. 10. Confusion for the KNN.

E. Support Vector Machines

Another model that we going to use in this work is Support Vector Machine. In this work we going to use two different Kernels functions in order to calculate the cost function as well as a $L2$ regularization to reduce the risk of overffitng. Given this background, we going to use a linear kernel, Equation (8), and radial basis function as kernel (RBF), Equation (9).

$$K(x, x') =< x, x' > \tag{8}$$

$$K(x, x') = \exp(-\gamma \sum_{j=1}^{k} (x_{ij} - x'_{ij})^2) \tag{9}$$

We found using cross validation that the RBF kernel result in a model with a highest accuracy when we compare to the model that uses linear kernel, as presented in Fig. 11, when we consider $\gamma = 0.003$, this γ value is calculated by the implementation using the division of 1 by the multiplication of number of features by the variance of the features.

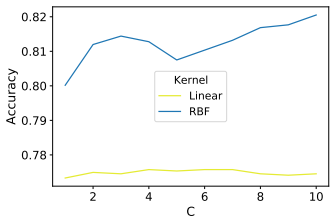


Fig. 11. The accuracy of the SVM using two different kernels for different values of the regularization coefficient C.

Using them the RBF kernel we search to verify the value of C that improve as much as possible the value of the accuracy in the cross validation. The result is presented in 12. So now from this cross validation result we define a model with $\gamma = 0.003$ and $C = 100$ that can be applied in the test set in order to define the classes of the data.

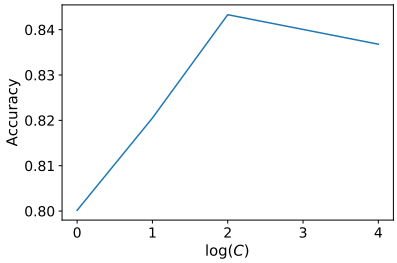


Fig. 12. Finding the best value for the regularization coefficient C.

Using the model the accuracy found in the test was 0.84, with a precision of 0.74 and a recall of 0.86, resulting in the confusion matrix presented in Fig.13. This model reduced the number of successful applications that were predicted as unsuccessful and increase the number unsuccessful application that were classified as as successful.

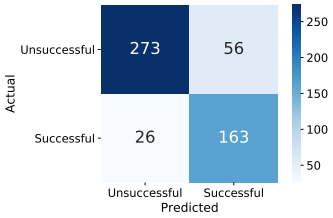


Fig. 13. Confusion matrix of SVM model.

V. Conclusion

As a result of the analysis, we identified that the LDA resulting in a model with the highest value of recall, which is associated with the reduced number of error in classifying the truly successful applications as successful, at the same time this model has the greatest number of truly unsuccessful application that were predicted to be successful which reduced the precision of the model. The model with the highest accuracy, SVM, reduced the number of right predictions of truly successful applications, at the same time, compared to the LDA result, SVM predicted a lower number of truly unsuccessful applications as successful. Considering these two models, SVM and LDA, the best that we used in this we can say that LDA works as a filter that let more unsuccessful application pass as it increases the number of truly acceptable application pass, in order hand, SVM produced a more restrictive filter, that can let more successful application as not accepted at the same time that reduces the acceptance of unsuccessful applications.

References

[1] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. New York, NY, USA: Springer New York Inc., 2001.

[2] Gareth James et al. *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014.

[3] Max Kuhn and Kjell Johnson. *Applied predictive modeling*. New York, NY: Springer, 2013.