**Donors Choose Application Screening Using**

**DonorsChoose Organization Data.**

## ABSTRACT

DonorsChoose.org is an organization that will fund most teachers who are willing to do projects and are in need of funds. This organization was founded in 2000 by a school teacher who empowers all the school teachers from all over the country to send requests for needed materials to express their implementations to help students. DonorsChoose.org once received hundreds of project proposals and needed funding. The existing system has to manually screen of each project proposal before posting it on DonorsChoose.org and funding the project. It tends to be tedious and time-consuming to check and verify every project. To automate this very process is the whole aim of this project. In the same pursuit, we have collected the data from Kaggle, and tried to pre-process data by removing unnecessary information from the data obtained. We have tried to use three different vectorizations (BOW, TFIDF, TFIDFw2v) and tried to train three most suitable machine learning algorithms which could deal a binary class classification problem. We have used (Logistic regression, XGBoost, Naive Bayes). This project aims to analyze all possible combinations and tries to indentify which vectorization and machine learning model to use for the DonorsChoose dataset by analyzing the AUC score and F1 score.

## CC CONCEPTS

Natural Language Processing, Machine Learning, Data Visualization

## KEYWORDS

BOW (Bag of Words) or OHE (one hot encoding), TFIDF (Term frequency and Inverse document frequency), TFIDFw2v (Term Frequency and Inverse document frequency combined with w2v), Logistic regression, XG-Boost, Naïve Bayes, ROC\_AUC(Area under the curve), Confusion matrix, F1 Score.

## 1. Introduction

DonorsChoose is a nonprofit organization located in the United States that allows anyone to contribute directly to public classroom projects. Since 2005, the group has received Charity Navigator's top rating. They declared in January 2018 that they had supported one million projects. At a minimum, one project has already been approved on DonorsChoose in 80 percent of public schools in America. To better improve the DonorsChoose service, the organization has put up the previous data that they have in Kaggle, and expressed a data science problem for that dataset. The description of the problem statement that DonorsChoose has given is as follows,

1. To automate and increase the efficiency of the screening process.
2. To increase the accuracy of choosing the right proposals for funding
3. Use of manpower to for the screening process is to be minimized.

## 2. Data Collection and Preparation

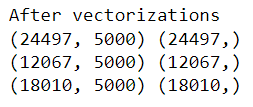
The Dataset is from the organization consisting of various attributes that show the project proposal details with respective teacher details who require funding. We have created the data set where it included attributes such as school\_state, teacher\_prefix, project\_grade\_category, teacher\_number\_of\_previously\_posted\_projects,project\_is\_approved clean\_categories, clean\_subcategories, essay, price is a total of 9 attributes.

Out the of 9 attributes, “project\_is\_approved” attribute is the class variable. Class variable here can take “0” or “1”, where “0” indicates the funding being rejected and “1” indicates that fund has been approved. Our first analysis to be done is check how balanced is the data. The no of “1”s should almost be the same for no of “0”s for the model to learn unbiased.



We have 40% of “0” classified data, and “60%”of “1” classified data. Out of rest “8” features, “5” of them are categorical features, “2” of them are numerical features. The “essay” attribute is the text feature. We have used (one hot encoding) to vectorise the categorical features. We used normalization for the numerical data. For the essay part of the data we used three different types of encoding. (BOW, TFIDF, TFIDFw2v)

Results with **BOW**:

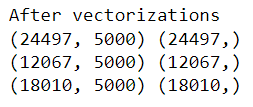


The Bag of words vectorizer yielded 5000 features.

## Results with TFIDF:

## (Term Frequency–Inverse Document Frequency), It is a numerical statistic that Describes how important a word is to a document.

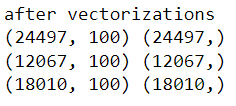
After the TFIDF vectorization the results of the features as follows,



Results with **TFIDFw2v**:

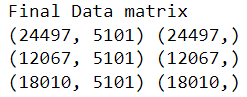
Here we take the” essay” attribute for performing all text encoding vectorization techniques such as a bag of words, TFIDF(Term Frequency Inverse document frequency), and TFIDF word2vector.

After vectorization results,

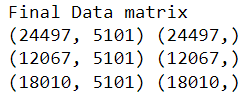


We have transformed them into one similar data frame. When finally merged, the “Essay” text encoded attribute with the remaining numerically or categorically encoded attributes, The results are as follows,

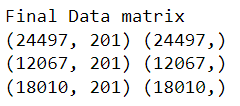
Final BOW vectorized data shape:



Final TFIDF vectorized datashape:



Final TFIDF\_W2V vectorized datashape:



## 3. Prediction Models and Results

We have applied three different models to build the prediction system. We use Naive Bayes, XGBoost, and Logistic Regression for checking the performance on “Bag Of Words” and “TFIDF (Term Frequency Inverse document frequency)”, TFIDF (Term Frequency Inverse document frequency) word2vector.

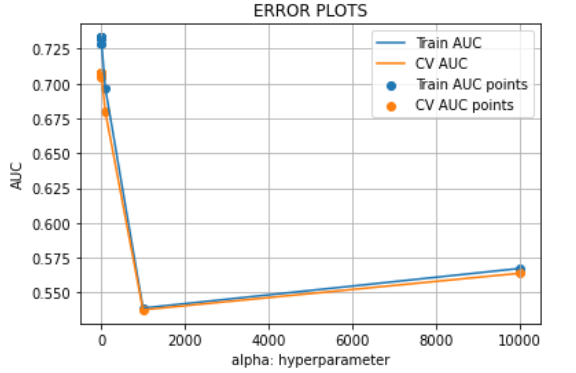
Since we notice intuitionally noticed the conditional independency we planned on using Naive Bayes. The reason we wanted to use Logistic Regression is because the range of its loss function is between 0 to 1, which best suits the binary classification problem. We wanted to implement a decision tree, We used Xtreme Gradient Boosted Decision tree to check the results.

## Naïve Bayes implementation:

We tried to implement Grid search for the initialization of hyperparameters. The hyperparameter in Naïve bayes algorithm is alpha.

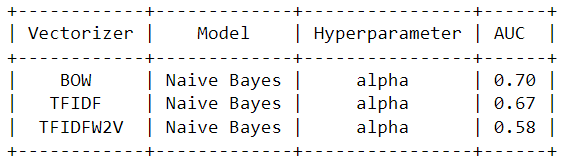
Alpha lies in the range, [[0.00001,0.0001,0.001,0.01,0.1,1,10,100,1000,10000]]

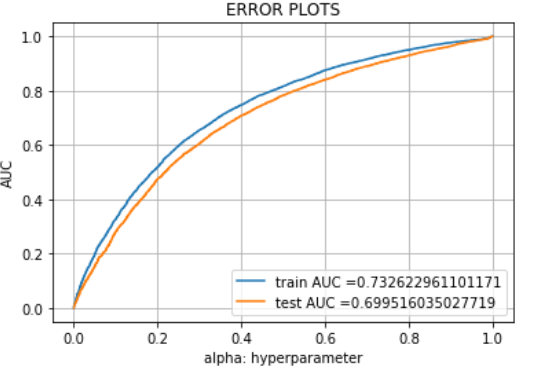
The ROC\_AUC curve to decide upon which parameter value is as follows,



Alpha value chosen is “1”

## Results of Naïve Bayes are as follows:



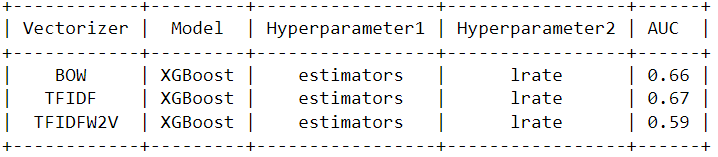


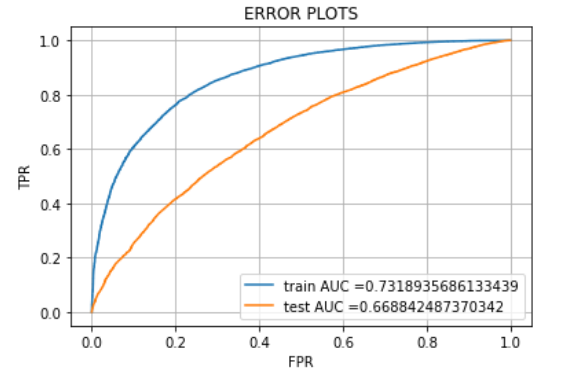
BOW vectorization gives the best accuracy for Naïve Bayes.

## XGBoost implementation:

In the XGBoost implementation we have done Grid search on the hyperparameters, with range of values  
“estimators” = [5,10,50, 75, 100]  
“l\_rate” = [0.0001, 0.001, 0.01, 0.1, 0.2, 0.3].  
After the Grid search the values turned out to be estimators = 100, l\_rate = 0.01 for all the three vectorized data.

Library used to implement: XGBoost  
Classifier used : XGBClassifier

Results of XGBoost are as follows:  


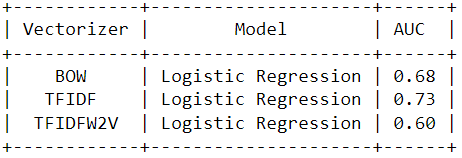


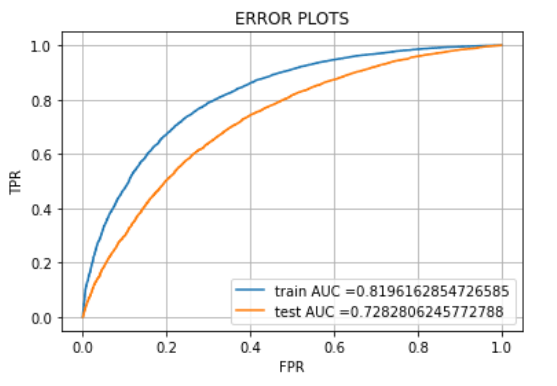
XGBoost with TFIDF Vectorization gives best accuracy.

**Logistic Regression:**

The module is imported from SkLearn. There are no explicit initializations done to this model. We went with default values given in the Sklearn.

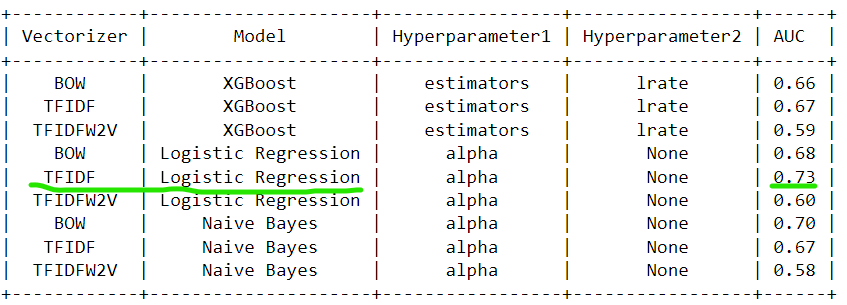
The Results of logistic regression are as follows:





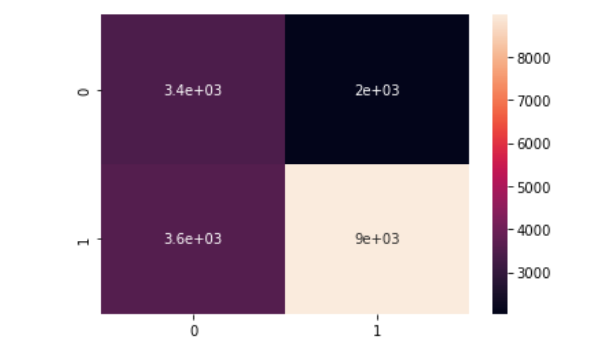
## 4. Discussions and Conclusions

Depending on the produced results using various algorithms such as Naïve Bayes, XGBoost, Logistic regression on different vectorized data such as BOW (Bag of Words),TFIDF (Term Frequency Inverse document frequency), TFIDF (Term Frequency Inverse document frequency), word2vector, realized that, Vectorizing the donor choose dataset to TFIDF and using a Logistic Regression algorithm to predict the fund approval has highest accuracy.



With these results, we will be able to generate a decent DonorChoose.org Application Screening model to predict the best project proposal.

**Confusion matrix for Logistic Regression using (TF-IDF):**

****

F1\_SCORE = 0.76

TP = 9e+03

TN = 3.4e+03

FP = 2e+03

FN = 3.6e+03

## 5. Future Work

The project can be further tested upon accuracy by implementing Support Vector Machine, and K-NN algorithms and also trying to Upsample or DownSample the data to see which of these could give a better accuracy over the DonorsChoose dataset.

## Acknowledgments

We sincerely thank Dr Zhang, for her constant support and guidance, She is the inspiration to explore the depths of the Data Mining for making this project work.

## References

* <https://www.donorschoose.org/>
* <https://www.kaggle.com/c/donorschoose-application-screening>
* <https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/>
* <https://machinelearningmastery.com/extreme-gradient-boosting-ensemble-in-python/>