ATLANTIS PRESS

Advances in Social Science, Education and Humanities Research, volume 223

International Conference on Education, Economics and Social Science (ICEESS 2018)

Binary Classification for Teacher Donor's Project

Yunwei Zhang*
Department of Statistics
University of California
Davis, USA

Zibin Zhang
Missile Engineering Department
Army Engineering University
Shi Jiazhuang, China

***Abstract*—Classification always plays an important role in statistical machine learning, which contains both binary classification problems and multi-label classification problems. This article focuses on binary classification models including natural language processing for text objects to help teachers to improve their chances of being funded based on real data sets collected by DonorsChoose.org. Comparing about two natural language processing methods for projects proposals proposed by teachers, we also implement various statistical algorithms on our data sets, aiming to enhance the classification accuracy which can be measured by model accuracy and the area under the curve(AUC). In conclusion, the text objects are important for computer to conduct supervised learning and the length of the proposal and the price column are the crucial features. In addition, the best model will be the LightBGM with AUC 0.77 and accuracy 86%.**

Keywords—binary classification; natural language processing; statistical machine learning models; Python

I. INTRODUCTION

Founded in 2000 by a high school teacher in the Bronx, DonorsChoose.org empowers public school teachers from across the U.S. to request much-needed materials and experiences for their students. Each year, DonorsChoose.org receives hundreds of thousands of project proposals for classroom projects in need of funding. Right now, a large number of volunteers is needed to manually screen each submission before it's approved to be posted on the DonorsChoose.org website. Therefore, using training data to conduct supervised learning that can predict whether or not a DonorsChoose.org project proposal submitted by a teacher will be approved is around the corner.

We have access of the previous data records which have already been split into training and test data sets. As for the training data, the size is of 182080 rows and 16 columns including three different forms: numerical, categorical and text forms. The detail description can be referred to in TABLE I. Moreover, we have one additional resource data set as an aid for supervised learning with 1541272 rows and 4 columns. The detail description can be referred to in TABLE II. As for the test data, we have 78036 rows and 16 columns contains information corresponding to the training set, which we use to test our models.

TABLE I. TRAINING DATA DESCRIPTION

Column Name	Type
Id	Numerical
Teacher_id	Categorical
Teacher_prefix	Categorical
School_state	Categorical
Project_submitted_datetime	Numerical
Project_grade_category	Categorical
Project_subject_categories	Categorical
Project_subject_subcategories	Categorical
Project_title	Natural Language
Project_essay_1	Natural Language
Project_essay_2	Natural Language
Project_essay_3	Natural Language
Project_essay_4	Natural Language
Project_resource_summary	Natural Language
Teacher_number_of_previously_posted_projects	Numerical
Project_is_approved	Target Variable

TABLE II. RESOURCE DATA DESCRIPTION

Column Name	Type
Id	Numerical
Price	Numerical
Quantity	Numerical
Description	Natural Language

II. PRE-PROCESSING OF THE NON-TEXT CONTENT

A. Data Munging

First of all, because of the duplication of Teacher_id, apparently, one teacher might have submitted several proposals, we combine the Teacher_id and create a new column--count_items to represent the number of proposals submitted by one teacher. Secondly, since it only required two project essays after May 17, 2016, we combine "project_essay_1" and "project_essay_2" to be "essay1" before this particular date and the process is similar for another two essay columns. It is natural for us to notice that the length of the proposal title or the summary may affect the decision and therefore, we use bar

plot to check whether it is essential to add the length as new columns or not. The bar plots (Fig. 1, Fig. 2, Fig. 3) shows the impact of the continuous variables.

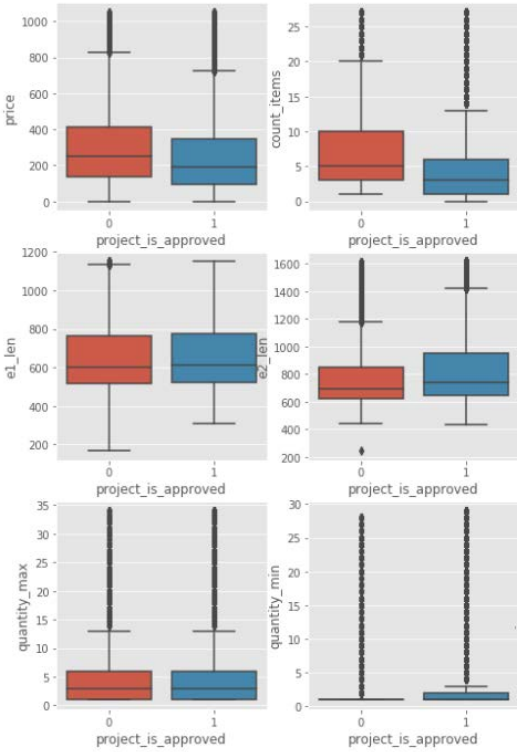

Fig. 1. Influence of continuous variables.(part1)

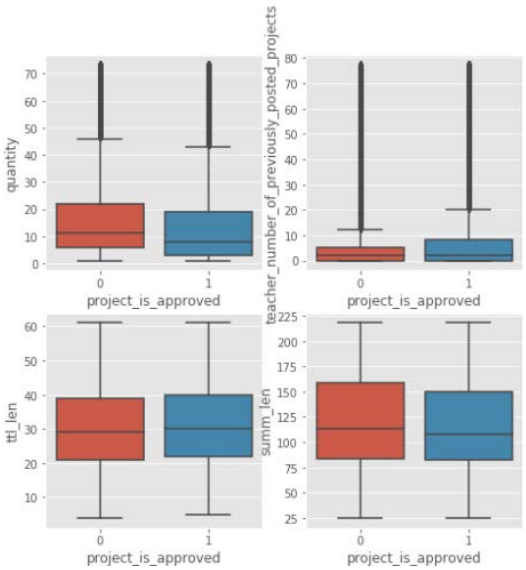

Fig. 2. Influence of continuous variables.(part2)

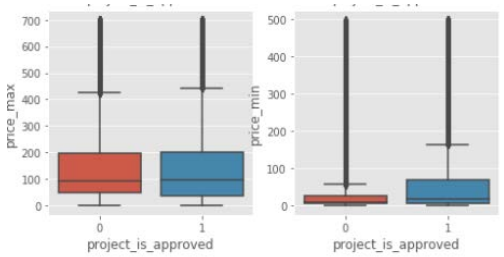

Fig. 3. Influence of continuous variables.(part3)

We observe that whether the project is approved or not, "ttl_len" and "summ_len"do not have apparent difference, so we drop them.

B. Feature Selection

We calculate the correlation between all those features and notice that "quantity_max" and "price_max" are highly correlated with other features so we decide to drop those two columns.

III. NATURAL LANGUAGE PROCESSING FOR TEXT DATA

TF-IDF is a classic way to convert text information. Since the resulting matrix of TF-IDF is large, we use Truncated SVD to reduce the dimension of the sparse matrix. 1000 SVD components are used in our models. In this paper, we also use an intuition method to deal with text content.

A. Vectorization by TF-IDF only (method 1)

1) TF-IDF is a text feature extraction algorithm used for text data. It is impossible for us to select features directly from the text data and therefore, we use The Bag of Words representation in Python sklearn package to represent the corpus of documents with a matrix one row per document and one column per token occurring in the corpus based on the feature defined as each individual token occurrence frequency,as in [5].

2) Truncated SVD performs linear dimensionality reduction by means of truncated singular value decomposition (SVD). However, it is different with principal component analysis (PCA) since it does not center the data before computing the singular value decomposition, which also indicates that it can work with scipy.sparse matrices efficiently, which is suitable for use in text information feature transformation,as in [6].

B. Intuition method for text content (method 2)

In order to reduce the dimension of TF-IDF matrix, we want to select features that have the most influence on the classification before doing TF-IDF.

- We first split the text into two groups based on whether this proposal is approved or not.
- Then, stemming and lemmatization are used on the corpus. Stemming usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes,as in [2]. Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to

remove inflectional endings only and to return the base or dictionary form of a word,as in [3].

- Finally, we count the word frequency and define the new frequency ratio. We define the frequency ratio by using the difference between the number of the word shown in approved proposal and the number of the word shown in rejected proposal divided by the total number of the word occurred, as in (1).After calculating the frequency ratio, We select 10000 words with the largest frequency and only vectorize these words.

$$R(w) = \frac{\#\text{ in approved text} - \#\text{ in rejected text}}{\#\text{ in all text}} \quad (1)$$

IV. MODEL SELECTION

There are several binary classification models that we can apply to the transformed matrix from the raw training and resource data sets such as Logistic Regression, KNN, Ridge Regression, SVM, Random Forest, XGBoost and LightGBM,etc. Considering that the data sets are large, we avoid using SVM and Neural Network because of the computation complexity. We compare Logistic Regression, KNN and Random Forest models on the matrix obtained by TF-IDF vectorization only method for the text data concatenated with other numerical and categorical variables. As for the sparse matrix obtained by intuition method, we compare three classification methods:Logistic Regression, Random Forest and LightGBM.

A. Model Introduction

1) Logistic Regression is typically used when we have binary dependent outcomes,as in [7]. Therefore, since we want to predict whether the teacher actually gets the donates or not, which can be treated as 0,1 variables where 0 means they fail to get the money and 1 is the opposite, it is reasonable to try logistic model. (Also, notice that logistic model can also be used when we have multi-class outcomes but clearly, this is not our case here.) The core methodology behind logistic regression is also gradient descent, however, unlike OLS, the coefficients usually do not have a closed-form solution. (i.e.we can not actually get the formula for a certain β does not include other β's when we have higher dimension.)

2) A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the data set and use averaging to improve the predictive accuracy and control over-fitting,as in [5]. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement if we use bootstrap method (by default).

3) Python sklearn KNN classifier is the classifier implementing the k-nearest neighbors vote,as in [6]. We have different methods of calculating the distance such as the Chebyshev distance, Mahalanobis distance, Bhattacharyya distance. Also, it is efficient and is not sensitive to outliers.

4) LightBGM is a gradient boosting framework that uses tree based learning algorithms. It has much higher efficiency and the training speed is faster than traditional classification models. Also, the memory needed is lower and the accuracy is better. In addition, it is GPU learning supported and capable of handling large-scale data sets, which means it can also be used for large-scale image data analysis,as in [1].

B. Model comparison for method 1

Comparison of those three different models is shown in the figure below.

TABLE III. MODEL COMPARISON FOR METHOD 1

Method	Parameter	Accuracy	AUC
Logistic Regression	Lbfgs	0.85	0.723
KNN	N_neighbors=70	0.85	0.640
Random Forest	Max_depth=7 Max_depth=13	0.85	0.704 0.691

We can clearly see that the accuracy of those three models are the same but the AUCs are slightly different. This figure shows the best parameter chosen in Logistic Regression model and KNN model separately and for the Random Forest model, different maximum depth of the tree has different AUC. Therefore, it is essential to try various parameters in the model to locate the best one for classification.

C. Model comparison for method 2

Comparison of those three different models is shown in the figure below.

TABLE IV. RESOURCE MODEL COMPARISON FOR METHOD 2

Method	Parameter	Accuracy	AUC
Logistic	Lbfgs	0.85	0.73
Random Forest	N_tree=80,max_depth =85,entropy	0.85	0.72
LightGBM	Num_boost_round=150	0.86	0.77

As for the accuracy of those models, the result is similar with method 1--they are all the same. However, by choosing the best parameter used for the model, it can be observed that the best AUC is 0.77, which is much higher than others.

V. MODEL EVALUATION

Model evaluation is an essential part of model development process. Not only does it help with choosing the best model that represents our data most, but also contributes to getting the best understanding about how well the chosen model works on the test data, i.e. the potential data sets in the future,as in [8]. Generally speaking, model evaluation can be divided into two sections: classification evaluation and regression evaluation. In this paper, we focus our study on the classification evaluation.

A. *Accuracy*

Accuracy in statistics means the proportion of the total number of predictions that are correct. It can be obtained from the confusion matrix. Both true positive rate (sensitivity) and true negative rate (specificity) contribute to accuracy,as in [4].

B. *AUC*

An ROC curve (receiver operating characteristic curve) is a curve showing the performance of the classification model at all classification thresholds. The x-axis representing the false positive rate and the y-axis representing the true positive rate,as in [4].

AUC stands for the area under the ROC curve. And therefore, it is easy to get the core that the large the area under the ROC curve, the better the model performs, given that the probability that we achieve the correct classification is larger.

C. *Examples for this study*

For this Teacher Donor's project, we show the ROC curve and the AUC for method 2 using Logistic Regression and LightBGM models, separately, as in Fig. 4 and Fig. 5.

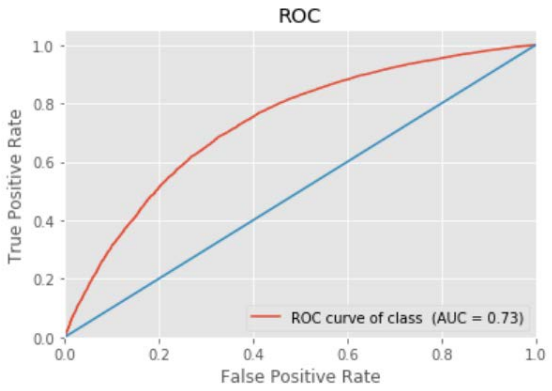

Fig. 4. ROC curve of Logistic Regression using method 2.

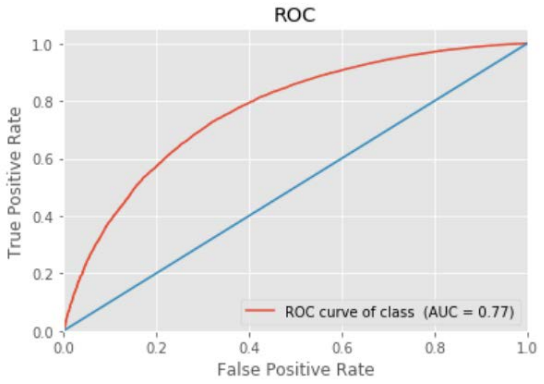

Fig. 5. Comparison of three models for method 2.

The blue line in the figure is the diagonal line representing the points that have the same x value and y value. The red line is the ROC curve for the model.

VI. CONCLUSION

A. *Non-text content*

For all the non-text content features, the length of the essay, named "e1_len" and "e2_len" in our figure, and the price of the item required, named "price_max" and "price_min" in our figure, are the crucial features which play an important role in the model building.

B. *Model assessment*

The intuition method of dealing with the text content is better than the original TF-IDF only method. Compared with the highest AUC 0.77 obtained with LightBGM by the intuition method, the largest AUC achieved by TF-IDF only method is 0.723 when using Logistic Regression model.

As for the difference between models within method 1, Logistic Regression shows the best performance while KNN shows the worst.

As for method 2, all the AUCs are larger than 0.7 and the difference is not too much, which also, in generally, better than method 1 no matter which model is used. Interestingly, as the most traditional model, Logistic Regression model is accurate than Random Forest but not comparable with LightBGM.

REFERENCES

[1] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree." In Advances in Neural Information Processing Systems (NIPS), pp. 3149-3157. 2017.

[2] Mehta, Manish, Rakesh Agrawal, and Jorma Rissanen. "SLIQ: A fast scalable classifier for data mining." International Conference on Extending Database Technology. Springer Berlin Heidelberg, 1996.

[3] Shafer, John, Rakesh Agrawal, and Manish Mehta. "SPRINT: A scalable parallel classifier for data mining." Proc. 1996 Int. Conf. Very Large Data Bases. 1996.

[4] Christopher D. Manning, Prabhakar Raghavan , Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press, Cambridge, England, 2008.

[5] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, "Scikit-learn: Machine Learning in Python," JMLR 12, pp. 2825-2830, 2011.

[6] Lars Buitinck , Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, "API design for machine learning software: experiences from the scikit-learn project," 2013.

[7] Trevor Hastie, Robert Tibshirani, and Jerome Friedman, "Elements of Statistical Learning," Springer, 2017.

[8] Unpingco José, "Python for Probability, Statistics, and Machine Learning," Springer, 2016.