US Visa Application Report

Jingshu Song, M.S.

Limin Tan, M.S.

The George Washington University

1. INTRODUCTION

a. Introduction

With the development of economic and the process of globalization, there is a growing number of people willing to work in US. If foreigners would like to work in US legally, the first step they need to do is to apply visa. Based on the department of labor report, in 2016, the Office of Foreign Labor Certification (OFLC) processed 775,979 employer applications for 1,699,957 positions for temporary and permanent labor certifications. The total number of case adjudication increased by 9 percent from the previous fiscal year. The National Prevailing Wage Center processed 133,242 PWD requests for use in the PERM, H-1B, H-1B1, E-3, and H-2B programs. This was an increase of 2,205 applications over the previous fiscal year, or 2 percent. H-1B applications saw a 5 percent increase in the number of determinations issued in 2016 over 2015. The Department processed 633,943 H-1B applications, certifying 1,257,849 positions. In 2016, 126,143 permanent labor certification applications were adjudicated, a 41 percent increase from 2015. Therefore, with more people applied, the success rate of application is decreasing recent year. Besides that, it takes a long time for application. People also need to pay the application fee to the government and the relevant fee to lawyers who help them. Considering the time and cost, if the application is denied, it is a huge loss for them.

Thus, it is necessary to predict visa application decision and to gain a better understands of the current application situations.

b. Problem Statement

If foreigners want to work in US legally, employers must apply visa for them, and it will take more than six months. However, the government will refuse the application sometimes. Both employees and employers

waste a lot of time and money. Therefore, we would like to create model to predict visa decision based on the employee information.

c. Problem Elaboration

A permanent labor certification issued by the Department of Labor (DOL) allows an employer to hire a foreign worker to work permanently in the United States. In most instances, before the U.S. employer can submit an immigration petition to the Department of Homeland Security's U.S. Citizenship and Immigration Services (USCIS), the employer must obtain a certified labor certification application from the DOL's Employment and Training Administration (ETA). The DOL must certify to the USCIS that there are not sufficient U.S. workers able, willing, qualified and available to accept the job opportunity in the area of intended employment and that employment of the foreign worker will not adversely affect the wages and working conditions of similarly employed U.S. workers. Therefore, applying visa for foreigners is a long and tough process which needs a lot of material and audit. Since the government report of visa application, there are a lot of factors led to the impact on the final decision of application, such as: wage, title, working state, industries, major and so on.

d. Motivation

The motivation of our project is to predict the application decision based on the employers and employees' information, such as: wage, industries, position and employer education. We will sort the feature importance and select the features that have a huge impact on the decision.

e. Project Scope

Our project includes the following part. We will select the target and feature and preprocess missing value. Based on the features, we visualize the current situation of visa application, such as application type, the number of applications. After that, we will preprocess these features, and encode them. Next, logistic regression model, random forest model and decision model will be built, and their performance will be

compared to select the best model. Finally, based on the model we chose, we will sort the features' importance to select the factors that have a huge impact on the application decision.

2. LITERATURE REVIEW

a. Relevant research

Considering the problem type of our project, we mainly researched paper about how to deal with classification problem and improve the accuracy of model, and how to preprocess imbalanced dataset. There are a lot of research papers that authors developed a random forest model to solve classification problem and analyze the impact of the features.

Nitesh (2002) investigated the right way to deal with imbalanced dataset using SMOTE. The paper clearly shows that SMOTE should be done after splitting the train and test dataset. By oversampling before splitting into training and validation datasets, some information will be lost from the validation set into the training of the model. However, because the SMOTE algorithm uses the nearest neighbors of observations to create synthetic data, it still bleeds information. If the nearest neighbors of minority class observations in the training set end up in the validation set, their information is partially captured by the synthetic data in the training set.

For a more comprehensive review of random forest to work on the classification problem, we read the paper written by Breiman (2001) and Criminisi et al. (2011). Random forests are built by combining the predictions of several trees, each of which is trained in isolation. Unlike in boosting (Schapire & Freund, 2012) where the base models are trained and combined using a sophisticated weighting scheme, typically the trees are trained independently, and the predictions of the trees are combined through averaging. There are three main choices to be made when constructing a random tree. These are (1) the method for splitting leaves, (2) the type of predictor to use in each leaf, (3) the method for injecting randomness into the trees.

An important milestone in the theory of random forests is the work of Biau et al. (2008), which proves the consistency of several randomized ensemble classifiers. Two models studied in Biau et al. (2008) are direct simplifications of the algorithm from Breiman (2001), and two are simple randomized neighbourhood averaging rules, which can be viewed as simplifications of random forests from the perspective of Lin & Jeon (2006).

We also researched about how to perform accuracy for imbalanced dataset. Stephane and Debra (1997) used precision and recall score to measure. Visualized performance by confusion matrix and ROC curve. A ROC curve plots the true positive rate on the y-axis versus the false positive rate on the x-axis. The true positive rate (TPR) is the recall and the false positive rate (FPR) is the probability of a false alarm.

3. METHDOLOGY

a. Dataset Description

Data covers 2011-2016 application records, and includes information on employer, position, wage offered, job posting history, employee education and past visa history, associated lawyers, and final decision. This dataset consists of 374362 applications described by 153 attributes.

b. Data Collection

Our dataset is downloaded from Kaggle (https://www.kaggle.com/jboysen/us-perm-visas), which is collected by the US department of Labor from 2011 to 2016.

c. Data Preprocessing and/or Feature Engineering

The first step we work on is to select features and targets. We chose application status as our target because our motivation is to predict the application decision. Considering that our dataset has 153 columns, we firstly removed the columns having more than half missing values, so there are 43 columns left. Next, we

looked through the meaning of these 43 columns, we noticed that some features present job posting information, such as: when this job posted, which website or newspapers this job posted on, how many times this position is posted. According to our research, this kind of information has a little relevance about visa application, so we also deleted these columns. There are 22 columns selected as features and they are: class of admission, country of citizenship, decision date, employer num employees, employer state, job info alt combo education experience, job info alt field, job info combo occupation, job info education, job info experience, job info foreign education, job info foreign language required, job info required normal, job info major, job info training, preparer info emp completed, pw amount 9089(wage), pw unit of pay 9089, pw level 9089, pw soc code(occupation code) , pw soc title(position title), pw source name 9089, college univ teacher. These features present the basic information of employees and employers, so it will be helpful to predict visa application decision.

After selection of features and targets, we worked on visualization to present the current situation of visa application from 2011 to 2016. From figure 1, the number of applications increased dramatically from 2011 to 2016, and the number of certified applications also increase a lot, but the number of denied applications keeps a stable situation. Besides that, there are a big difference between the number of certified and denied, so our dataset is an imbalanced dataset which we need to process later.

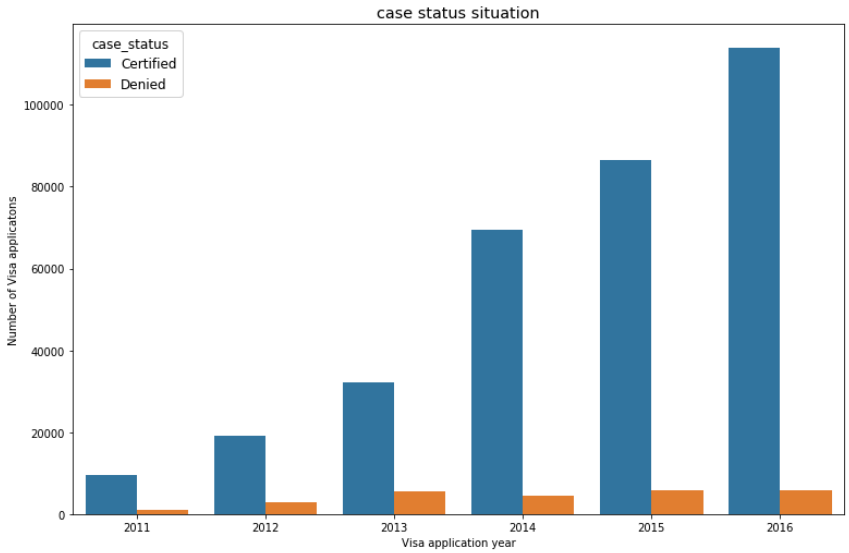

Figure 1

Figure 2 indicates the situation of application type. Our dataset contains more than 50 types of application totally, such as: H1B, F1, J1 and so on, and we just selected the most frequent 5 types to present the application type situation. As you can see, most applicants applied H1b visa, which is the most common working visa in US, and the number of H1B visa application increased a lot these six years. For other application types, the number of them is also increased from 2011 to 2016.

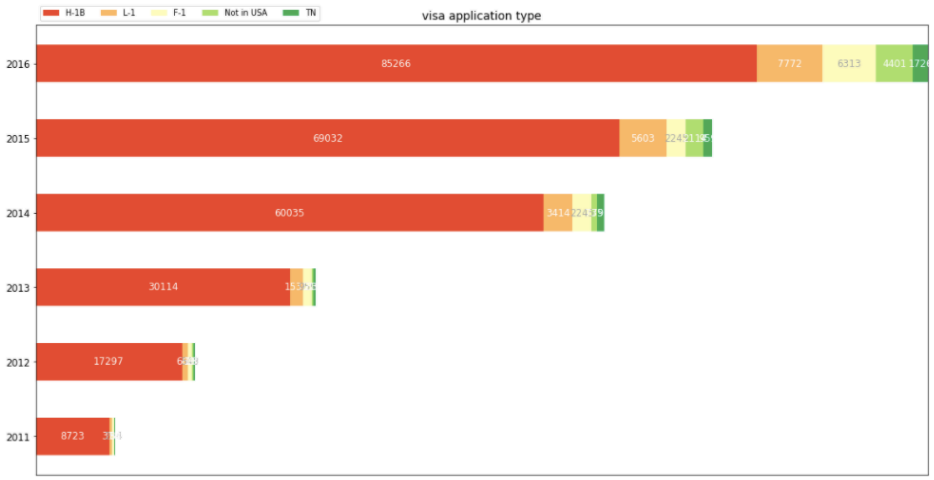

Figure 2

The following two figures (figure 3 and figure 4) show the situation of industries visa application. It is remarkable that top 7 industries are all relevant with computer and electronic industry, and rest of them are most business industry, such as: marketing, accounting, finance and so on, so these 20 industries can be divided into two main industries: computer industry and business industry. Besides that, it needs to be mentioned that the number of software developers' application, top one industry visa application, is three times as many as the top two industry application. From figure 4, we can gain the information about distribution of industries across all visa application. IT industries occupies nearly half of the application and the second one is economic sector (13.4%). This result is the same as the result of top 20 industries visa application.

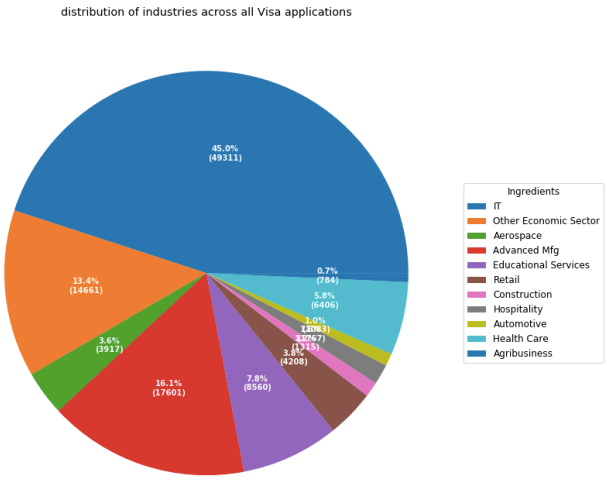

Figure 3

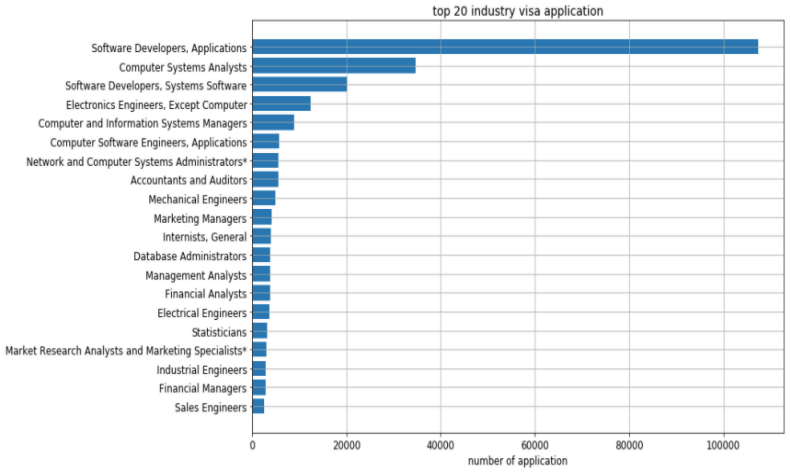

figure 4

The figure 5 presents the education level of applicants. There are nearly 10 thousand applicants having master's degree, which is the most in the all education level, and the second one is bachelor's degree. The number of applicants with master's degree and bachelor's degree don't have a big difference. However, these is a huge gap between the bachelor's degree and the doctorate degree, which is the third one in the all education level. Master's applicants are more than seven times than the doctorate's applicants.

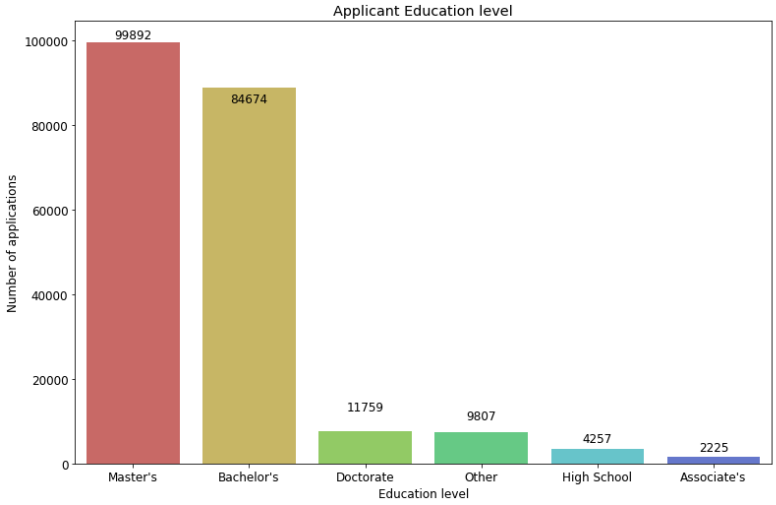

Figure 5

The figure 6 presents the geographical information about visa application. New York city has the most visa applicants and the second one is college station, which is easy to understand because most IT and business companies are located in these two cities. Figure 7 indicates the applicants' citizenship. The most applicants are from India, the second one is China and the third one is South Korea. It also needs to be emphasized that the number of applicants from India is five times compared with the Chinese applicants.

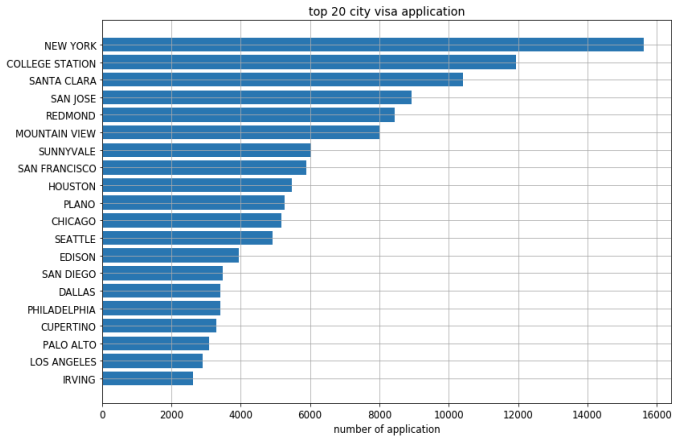

Figure 6

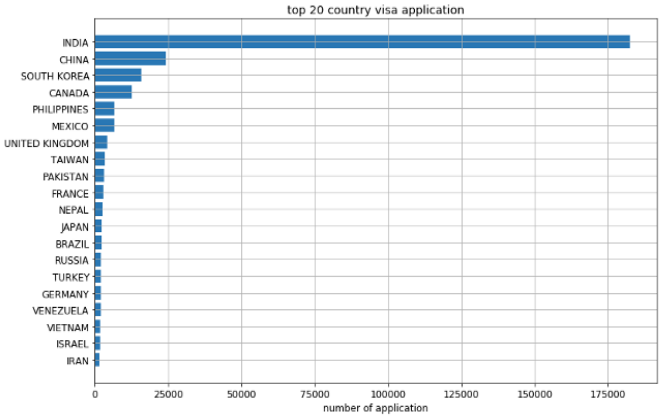

Figure 7

After visualizing the basic situation of visa applications, we need to deal with missing value, which is an important step of preprocessing data. Based on the features we selected, we sum the missing value of these features firstly to present the basic situation of data. Some features like decision date has really low percentage of missing value, while some features like class of admission has a lot of missing values. We mainly used three types to solve the problem of missing values. First, for the categorical features, such as: country of citizenship, education level and major, we will use 'None' to replace missing value because these kinds of features are hard to insert the missing values. Second, we will check the values of some features, for example, when checking the value of job info training, we noticed that most value is Y, there are only 10 percentage of value is N, so for these features which has a big difference between two values, we will use mode to replace missing values. Finally, some particular methods of processing missing value will be used on some features. For example, we use mean to replace the missing value of wage and use median to insert the missing values of the number of employees.

After solving the problem of missing values, we worked on preprocessing each feature for encoding and building model. For example, we changed the format of decision date where we only keep the year of decision date and removed the month and day. For the pw amount 9089 (wage), their unit is different. They are counted by year, month, day or hour, so we processed this feature with pw unit of pay to unify the counting unit and the amount of wage will be counted by year. Besides that, we also removed the unnecessary punctuation of the pw soc code and title columns.

Another essential step what we did is to reduce the level of some categorical features, such as: country of citizenship, major, title and so on. Especially, like the column of major, the number of unique values is more than 500. If these features are not processed, then the subsequent one hot encoding process won't work, and the model is unable to run because of the excessive memory consumption. In order to avoid this situation, it is important to reduce the level of them. First, for each category features, we sum the number

of the unique value of it. We can notice that the number of unique values for citizenship, major, title, soc name and soc code is more than 100, so we calculate the percentage of values in this feature. Then, for the values' percentage is lower than 0.5%, we will name these values as other in order to reduce the level of it. By using this way, the level of categories is greatly reduced, and it is able to work on one hot encoding, for example, for the title column, the number of unique values is reduced from more than 500 to 156.

After the all preprocessing, we worked on splitting train and test dataset to prepare for the model by using sklearn model selection library to split. After splitting, the training set has 284939 samples, and the testing set has 71234 samples.

The last step we did for the preprocessing is to handle the problem of imbalanced dataset. We chose smote to preprocess it. Smote performs certain operations to create additional training data real data. Operations such as rotation and tilt are disturbing natural methods of training data. Smote generate comprehensive examples in a less application-specific way, running in "functional space" rather than "data space." Minorities are over-sampled by taking each minority sample and introducing a comprehensive example along the line to join the network segment of any / all closest neighbors in the k minority categories. Depending on the amount of oversampling required, the neighbors in the k nearest neighbors are chosen randomly. This leads to select random points along the line segment between two specific features. This method effectively forces minority decision-making areas to become more general, so we think it is more reasonable to use smote for oversampling.

d. Data modeling and visualization

For our project, we mainly used four different models to predict the visa application decision: logistic regression, random forest, decision tree and extra trees. In our report, we will introduce these four models first, and talk about their application in our dataset.

The reason why we chose the logistic regression because is a method for estimating binary response variables based on one or more independent variables. The logic function can be written as an independent variable, which is to predict the probability of the dependent variable. It can be seen from the logic function that only the linear relationship between independent variables is considered, so the model cannot capture any non-linear relationship between features. One of the problems of logistic regression is that the coefficients are unbounded and can take extreme values, making other features useless and the model uncertain. Several ways to regularize appointments. In our project, we used the logistic regression method implemented in scikit-learn, where the sum penalty is the regularization parameter. If the feature size needs to be reduced, L1 regularization is very useful because it can produce more zero coefficients. Due to the imbalance of classes in the data set, we also plan to adjust the class weight parameter available in the logistic regression method.

Random forest is a common model when solving the classification problem, so we chose it. It combines the results of many basic models (weak learners), can deal well with categorical variables and outliers, and can handle nonlinear relationships between elements. We also chose the decision tree, but decision trees tend to overfit data. Therefore, we make adjustments to avoid overfitting: n estimators (number of trees), min samples split (minimum data points to split), min samples leaf (minimum data points of leaves). We also planned to adjust max features (the maximum number of features used for merging trees) and bootstrap (whether to use bootstrap samples when generating trees). The reason for bootloader is because our dataset is imbalanced, and the bootloader can be used to reduce the representation of the enhanced class of data.

The last model is extra trees. It can merge multiple trees and use random subsets of features to decompose sequences. In summary, by default, bootstrap (False) is used to plant multiple trees, so we can sample the dataset without replacement, and segmentation based on random splits between random subsets of features selected on each officer. Considering there two main differences, we chose this model for our project.

4. RESULT AND ANALYSIS

	Precision score	**Recall score**
Logistic Regression	**0.924**	**0.936**
Random Forest	**0.933**	**0.942**
Extra Trees Classifier	**0.931**	**0.940**
Decision Tree	**0.920**	**0.919**

Table 1

Through the training of our four models: Logistic Regression, Random Forest, Extra Trees Classifier, Decision Tree, we can notice from the table 1 that the performance of random forest is the best, because its precision and recall are the highest. Based on this result, we perform hyperparameter tuning on random forest. We generated a final model with above list of tuned parameters. The output of this tuned model came just about 0.1% higher in accuracy vs the untuned model. Through hyperparameter tuning we found the best parameters. The number of estimators is 240, the min samples splits is 5, the min samples leaf is 1, the max feature is auto, the boot strap is True.

We also worked on the confusion matrix. From figure 8, based on the result of the recall, we can know that out of all the positive classes we have predicted correctly, and from precision result, we can know that out of all the positive classes we have predicted correctly, 92.5% are actually positive.

```
---------- confusion matrix ------------
          Predicted_0  Predicted_1
Actual_0        26403         2153
Actual_1         2300        27537
---------- accuracy ------------
0.9237408593495796
---------- classification report ------------
              precision    recall  f1-score   support

           0      0.920     0.925     0.922     28556
           1      0.927     0.923     0.925     29837

    accuracy                          0.924     58393
   macro avg      0.924     0.924     0.924     58393
weighted avg      0.924     0.924     0.924     58393
```

Figure 8

5. CONCLUSION

a. Conclusion

In the conclusion part, we will summarize the whole project. After loading data, we first select target and features. The target is application status. When selecting the features, we remove the columns with more than half missing values and select them by meaning. After that, we worked on the data analysis to present the basic application information, such as: the citizenship of applicants. We used None to replace the missing values of category features and use mode/median to replace the numerical features. Next, we reduced the level of some categorical features and worked on encoding. Then, we split the training and testing dataset, and use SMOTE to deal with imbalanced dataset.

We chose four models: Logistic Regression, Random Forest, Extra Trees Classifier and Decision Tree. Among these four models, the classification result of Random Forest is the best, and we worked on hyperparameter tuning for the random forest model. Finally, we list the feature importance to overview which features contribute to the decision. The occupational code, prevailing wage, the total number of employees and job title largely determine the decision. This result is not surprising, because according to

US policy, there will be a certain tendency for people with income and specific job nature. The two features of Indians and software engineers account for a very important proportion in visa applications.

b. Project Limitation

One of the project limitations is to select the features. Our data set has 154 columns, which is diverse and extraordinary for data collation and advance. We read the description of all 154 columns to delete columns based on their meaning, which is not appropriate. If we delete some columns by mistake, it will cause some errors and misleading. Besides that, we deleted some column whose missing values are more than half, which may cause some important information to be deleted. Also, there are many ways to lose data, outliers and converted to digital data. We just tried several typical methods to deal with missing values and imbalanced data. There is no horizontal comparison to choose the best method.

Another limitation is the parameter search for the model, because the amount of data is too large, we cannot search for the optimal parameter for each model. Therefore, we only searched the optimal parameters for the most accurate model.

c. Further Research

It is still available for us to improve the final result because the tuned final model made no significant improvement over the untuned model. The precision score for the untuned random forest model is 0.93 and the score for tuned random forest is 0.94. There could be more ways we could improve the score, particularly by selecting a subset of features using the ranking obtained from observing feature importance ranking and then performing the same exercise we did as describe above.

As we mentioned in the limitation, we didn’t preprocess all columns, and the method of selecting features are needed to improve. Some columns with more missing values may contain valuable information. In

future research, we can do some feature engineer work on all columns to further improve the method of selecting features, so we can improve the accuracy of the model. Besides that, we can also try other models, like XG boosting, to view the result.

6. REFERENCE

Bradley, A. P. (1997). The Use of the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms. Pattern Recognition, 30(6), 1145–1159.

Chawla, N., Bowyer, K., Hall, L., & Kegelmeyer, P. (2000). SMOTE: Synthetic Minority Over-sampling TEchnique. In International Conference of Knowledge Based Computer Systems, pp. 46–57. National Center for Software Technology, Mumbai, India, Allied Press

Drummond, C., & Holte, R. (2000). Explicitly Representing Expected Cost: An Alternative to ROC Representation. In Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 198–207 Boston. ACM.

Kubat, M., & Matwin, S. (1997), Addressing the Curse of Imbalanced Training Sets: One Sided Selection. In Proceedings of the Fourteenth International Conference on Machine Learning, pp. 179–186 Nashville, Tennesse. Morgan Kaufmann.

Misha Denil, David Matheson, Nando de Freitas(2014), Narrowing the Gap: Random Forests In Theory and In Practice.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall and W. Philip Kegelmeyer (2002), SMOTE: Synthetic Minority Over-sampling Technique, Journal of Artificial Intelligence Research 16 (2002) 321–357.

N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer (2011), SMOTE: Synthetic Minority Over-sampling Technique, Journal Of Artificial Intelligence Research, Volume 16, pages 321-357.

R. Genuer, J.-M. Poggi, and C. Tuleau.(2008), Random Forests: Some Methodological Insights. arXiv:0811.3619

R. Genuer, J.-M. Poggi, and C. Tuleau-Malot.(2010), Variable selection using random forests. Pattern Recognition Letters, 31:2225–2236.

R. Diaz-Uriarte and S.A. de Andres.(2006), Gene selection and classification of microarray data using ´ random forest. BMC Bioinformatics, 7:1471–2105.

Sotiris Kotsiantis, Dimitris Kanellopoulos, Panayiotis Pintelas(2010), Handling imbalanced datasets: A review, Educational Software Development Laboratory Department of Mathematics, University of Patras, Greece

Safa ABDELLATIF1 and Mohamed Ali BEN, HASSINE(2018), ARCID: A new approach to deal with imbalanced datasets classification, Theory and Practice of Computer Science, pp.569-580

Sotiris Kotsiantis, I.D. Zaharakis(2006), Machine learning: A review of classification and combining techniques.