Syracuse University

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

H1B Visa Application Classification

CIS-563 INTRODUCTION TO DATA SCIENCE

Prateek Sahu, SUID: 809311241

December 8, 2019

CONTENTS

1 INTRODUCTION

United States economy relies heavily on temporary work visas, like H-1Bs, to hire highly trained, highly skilled employees from abroad[2]. But H-1Bs have been subject to tougher scrutiny, with denial rates on the accretion. The federal government(USCIS) issues only 85,000 visas - known as H-1Bs - annually to businesses in United States, which includes 20,000 for those with advanced degrees from U.S. universities. However, the number of applications are exceeding at prolific rate[9]. The process has come under intense scrutiny during the Trump administration; companies burdened with higher fees and more paperwork when seeking H-1B visas for workers. Denial rates has increased from 13% in 2017 to 33% through the second quarter of this fiscal year[11]

In this project, our aim is to predict the outcome of the H1-B visa status. In terms of Data Science, this is a classic case of multi-class classification problem. We have tried different classification algorithms and compared results from those classification model.

The USCIS claims that the selection process is a lottery, hence it is not clear how the values of the attribute affect the final outcome. Since the number of applicants are increasing day by day, our classification model could be an efficient tool for employers, who are considering to sponsor skilled temporary non-US workers, and individuals, who are seeking visa sponsorship.

2 PRIOR WORK

For starting the project, a notebook on kaggle is referred(placed at the same location the data set is present). The exploratory visualization section in notebook helped in understanding the nature the data set. The other report conducted a detailed data analysis and visualization for H-1B application distribution based on different input features such as location, salary, year and job type [4]. Although they had a prediction algorithm based on K-means clustering and decision trees, they provided prediction accuracies for only a small subset of job types instead of an average one. Overall, this report gave a good insight on the distribution of our data. Literature on how understand the Data set [10] helped in understanding the data more and further on feature selection. The second report which is used as reference used AdaBoost, Random Forest, Logistic Regression and Naive Bayes to predict the visa status [8]. We found some of their pre-processing steps for features quite motivating and implemented them ourselves accordingly before feeding the features into our own machine learning algorithms.

3 METHODOLOGY

In this section, we will discuss about various classification models and other concepts used in the project.

3.1 CLASSIFICATION MODELS

3.1.1 DECISION TREE CLASSIFIER

Decision Tree Classifier is one of the most popular machine learning algorithms used. A decision tree is a tree where each node represents a feature, each link represents a decision and each leaf represents an outcome(categorical or continues value). DecisionTreeClassifier [1] takes as input two arrays: an array X, sparse or dense, of size [n_samples, n_features] holding the training samples, and an array Y of integer values, size [n_samples], holding the class labels for the training samples. After being fitted, the model can then be used to predict the class of samples. DecisionTreeClassifier is capable of both binary (where the labels are [-1, 1]) classification and multi-class (where the labels are [0, ..., K-1]) classification. Common measures of impurity are:

$$GiniH(X_m) = \sum_k p_{mk}(1 - p_{mk}) \tag{3.1}$$

$$EntropyH(X_m) = -\sum_k p_{mk} \log(p_{mk}) \tag{3.2}$$

$$p_{mk} isp_{mk} = 1/N_m \sum_{x_i \in R_m} I(y_i = k) \tag{3.3}$$

3.1.2 RANDOM FOREST CLASSIFICATION

Random forest , like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction [12]

Random sampling of training observations When training, each tree in a random forest learns from a random sample of the data points. The samples are drawn with replacement, known as bootstrapping, which means that some samples will be used multiple times in a single tree. The idea is that by training each tree on different samples, although each tree might have high variance with respect to a particular set of the training data, overall, the entire forest will have lower variance but not at the cost of increasing the bias. At test time, predictions are made by averaging the predictions of each decision tree. This procedure of training each individual learner on different bootstrapped subsets of the data and then averaging the predictions is known as bagging, short for bootstrap aggregating.

Random Subsets of features for splitting nodes The other main concept in the random forest is that only a subset of all the features are considered for splitting each node in each decision tree. Generally this is set to sqrt(n_features) for classification meaning that if there are 16 features, at each node in each tree, only 4 random features will be considered for splitting the node.

3.1.3 K-Neighbours Classification

kNN is considered among the oldest and most commonly used non-parametric classification algorithms. To classify an unknown example, the distance (using some distance measure e.g. Eculidean) from that example to every other training example is measured. The k smallest distances are identified, and the most represented class in these k classes is considered the output class label. The value of k is normally determined using a validation set or using cross-validation.

In the classification setting, the K-nearest neighbor algorithm essentially boils down to forming a majority vote between the K most similar instances to a given âĂIJunseenâĂİ observation. Similarity is defined according to a distance metric between two data points. A popular choice is the Euclidean distance given by

$$d(x, x') = \sqrt{(x_1 - x')^2 + ... + (x_n - x'_n)^2} \tag{3.4}$$

but other measures can be more suitable for a given setting and include the Manhattan, Chebyshev and Hamming distance. More formally, given a positive integer K, an unseen observation x and a similarity metric d, KNN classifier performs the following two steps:

- It runs through the whole dataset computing d between x and each training observation. WeâĂŹll call the K points in the training data that are closest to x the set A. Note that K is usually odd to prevent tie situations.
- It then estimates the conditional probability for each class, that is, the fraction of points in A with that given class label. (Note I(x) is the indicator function which evaluates to 1 when the argument x is true and 0 otherwise)

$$P(y = j|X = x) = 1/K \sum_{i \to A} I(y^{(i)} = j) \tag{3.5}$$

Finally, our input x gets assigned to the class with the largest probability.

3.2 Handling Unbalanced Class

To put in simple words a data-set is imbalanced if the classes are not approximately equally represented. In our case also classes are highly imbalanced figure 4.1 clearly shows that with respect to CERTIFIED case status other classes are almost negligible. If the data set is imbalanced the model will be biased towards major class. Below we will discuss some of the common technique to handle unbalanced data. **Undersampling**: In this method

we downsize the actual data set in such a way that classes become balanced by reducing the majority class numbers. However, reducing the size would actually restrict data for the model to feel. The other technique is **Oversampling**: This method uses synthetic data generation to increase the number of samples in the data set. Two of the most common oversampling techniques are SMOTE and ADASYN.

3.2.1 SMOTE

If we have to explain SMOTE in brief then it First finds the n-nearest neighbors in the minority class for each of the samples in the class . Then it draws a line between the the neighbors an generates random points on the lines. [13]

3.2.2 ADASYN

ADASYN is an improved version of Smote. After creating the sample it adds a random small values to the points thus making it more realistic. In other words instead of all the sample being linearly correlated to the parent they have a little more variance in them i.e they are bit scattered. [14]

After referring to the above cited papers on SMOTE and ADASYN, we used ADASYN in our project since it was found better for multi-class classification.

4 Results and Findings

4.1 Pre-Processing and Feature analysis

The data set used is listed in Kaggle [3] by the name "H1B Prediction Case Status prediction for H1B Visa Applications(2015-2018)". The data is well divided into year-wise CSVc. We are

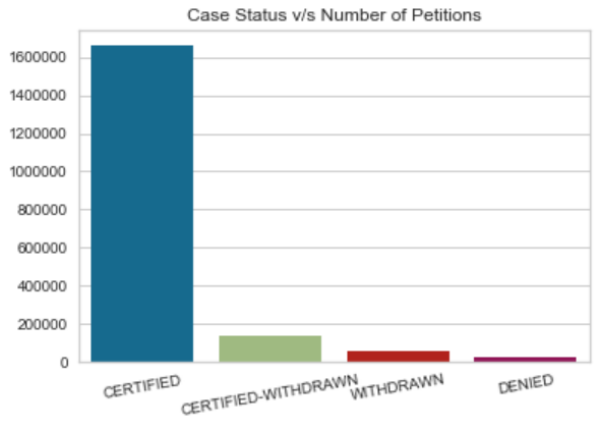
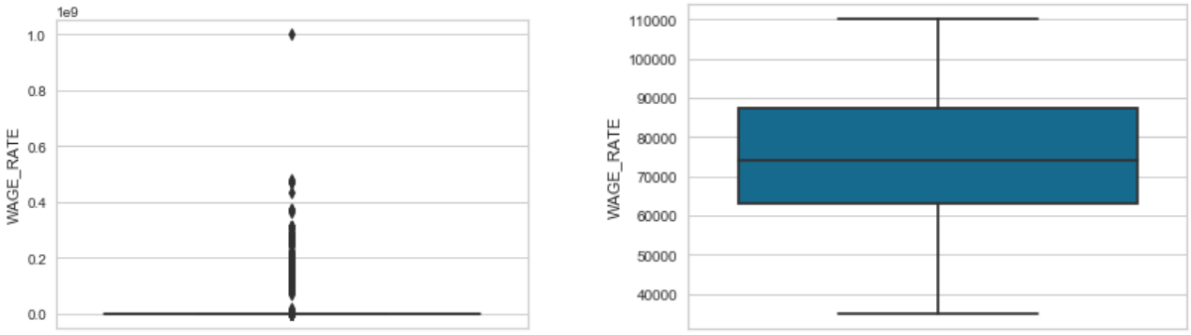


Figure 4.1: Case Status VS Number of Petitions

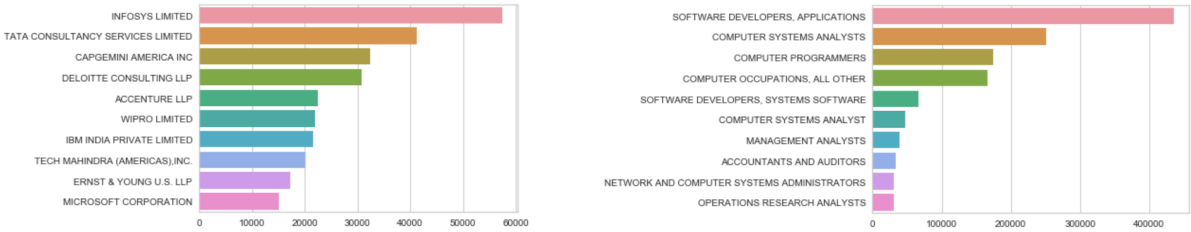
only considering H-1B_Disclosure_RAW_Data_FY15.csv, H-1B_Disclosure_RAW_Data_FY16.csv, H-1B_Disclosure_RAW_Data_FY17.csv and H-1B_Disclosure_RAW_Data_FY18.csv for classification analysis. On visual analysis the data many interesting were quite obvious to find. Some of the findings are listed below.



(a) Wage Rate with outliers

(b) Wage rate after removing Outliers

Figure 4.2: Wage rate Box Plot



(a) Top 10 H1-B visa sponsoring Employers

(b) Top 10 Job Profiles

Figure 4.3: Bar plot showing top employers and Job profile

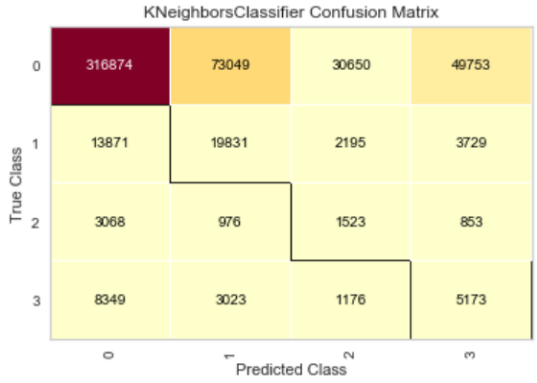
- Number of features in 2015 and 2016 data are 40, which is 52 for 2017 and 2018. Most of the relevant features are present in all years, although some feature renaming was required to bring data in sync.
- Once we have the data frame all the entries containing null values are dropped [6].
- On thoroughly analysing and taking reference from the prior work, feature sub-set selection is done. Features like case number, latitude and longitude of location, information about immigration lawerys and their address is dropped. Also any VISA_CLASS other than H1B is also dropped since they are not relevant to our problem.
- First plot is a bar plot showing overall shape of data in terms of Case Status. 4.1. The graph clearly shows the imbalance the the class data. We will discuss more about this latest in this section on how to stratify data before training the classification model
- The combination of WAGE_RATE and WAGE_UNIT_OF_PAY columns are used to calculate the wages of applicants on a yearly basis. Further, the two box plot shows the

data before and after removing outlier in the WAGE_RATE. 4.2. The lower bound and upper bound are selected after referring the average salary in US [5]

- Next interesting analysis is the state wise number of applicants. The plot clearly shows that California and Texas has the highest number of applications.
- Similar to the above analysis,bar plot for top 10 employers filling the H1-B visa petitions and top 10 Job profiles for which H1-B visas were raised shows that Infosys.

4.2 Model Evaluation

In this section, we will discuss performance of the model and further comparing it with the result of other models. The dataset is split into train and test part in 70-30 ratio. Lets discuss one by one

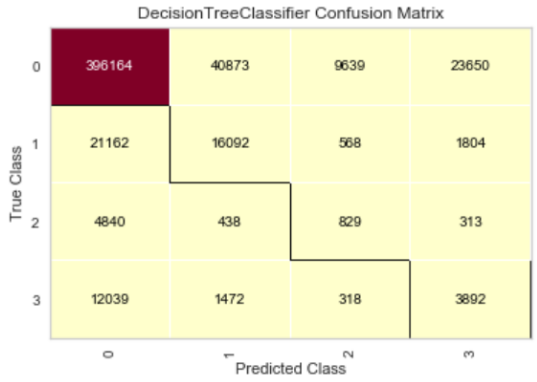


(a) KNeighbor Classifier Confusion Matrix

	precision	recall	f1-score
0	0.93	0.67	0.78
1	0.20	0.50	0.29
2	0.04	0.24	0.07
3	0.09	0.29	0.13
accuracy			0.64
macro avg	0.32	0.43	0.32
weighted avg	0.83	0.64	0.71

(b) KNeighbor Classifier Classification Report

Figure 4.4: KNeighbor Classification Model Evaluation

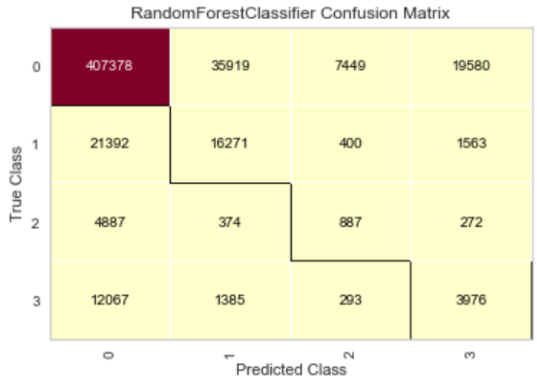


(a) DecisionTree Classifier Confusion Matrix

	precision	recall	f1-score
0	0.91	0.84	0.88
1	0.27	0.41	0.33
2	0.07	0.13	0.09
3	0.13	0.22	0.16
accuracy			0.78
macro avg	0.35	0.40	0.37
weighted avg	0.83	0.78	0.80

(b) DecisionTree Classifier Classification Report

Figure 4.5: DecisionTree Classification Model Evaluation



(a) RandomForest Classifier Confusion Matrix

	precision	recall	f1-score
0	0.91	0.87	0.89
1	0.30	0.41	0.35
2	0.10	0.14	0.11
3	0.16	0.22	0.18
accuracy			0.80
macro avg	0.37	0.41	0.38
weighted avg	0.83	0.80	0.82

(b) RandomForest Classifier Classification Report

Figure 4.6: RandomForest Classification Model Evaluation

1. K-Nearest Neighbor Classifier: This is the first classification model used in our prediction. Classification performance results are shown in figure 4.4
2. Decision Tree Classifier: Next we used Decision Tress Classification model. The results of decision tree are better when compared to KNN. We can observe the increase in recall which is 0.68 for KNN to 0.78 in Decision tree. Classification report at figure 4.5 confirms the better
3. Random Forest Classifier: As mentioned in the 3.1.2 that random forest is an improvement over 3.1.1 and the result also reflects the same. You may find the classification report in figure 4.6

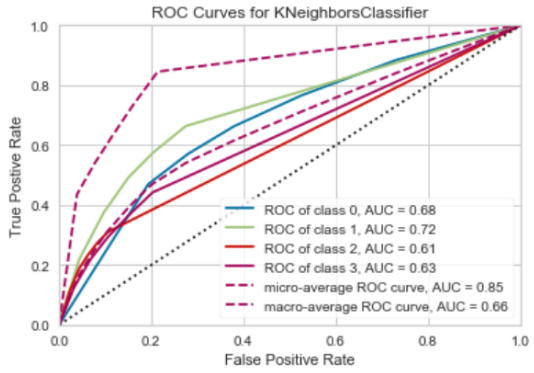
In the last part of this section ROC curve for all the models are drawn to characterize the trade-off between positive hits and false alarms [7] Lets concentrate on Micro-averege and macro-average ROC curve. AUC stands for "Area under the ROC Curve." which gives an aggregate measure of performance across all possible classification thresholds. From the above analysis we found that Random Forest model's performance is better than the other tow. AUC also confirms the same. Micro-average AUC for random forest is 0.93 compared to 0.89 for Decision Tree and 0.85 for KNN.

One interesting fact is when we compare the ROC curve of KNN (fig. 4.7a) and Decision Tee (fig . 4.7b) for CERTIFIED class. AUC of CERTIFIED class for KNN is 0.68, which is better than that of 0.64 of Decision Tree. Although, overall Decision tree is better than KNN. Similar trend is observed wih AUC of DENIED class for KNN and Decision Tree.

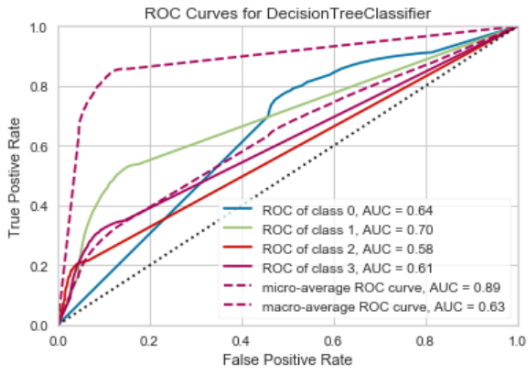
This means we can use different classification models for predicting specific class.

5 Conclusion and Future Work

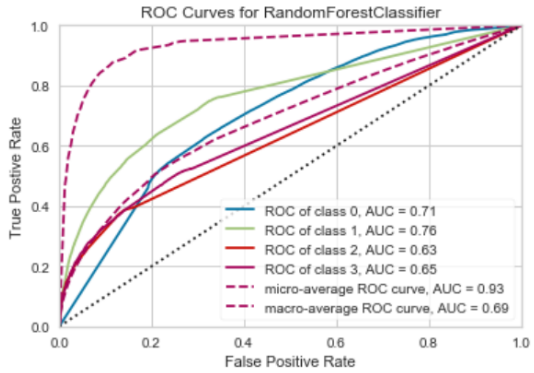
This project was an experiment and aimed at predicting the case status of H1-B applications. There are lot of interesting relations still in the data set which could be exploited



(a) KNN



(b) Decision Tree



(c) Random Forest

Figure 4.7: ROC Curve

to get better classification model. Its mentioned in section 1 that USCIS processes 85,000 applications in a year and out of those 85,000 applications 20,000 are reserved for applicants who have pursued higher education in United states. However, the data-set has no information about that. If there would have been more time, there are several direction this project can take. First of all, it would have been interesting to use Neural network models such as MLPClassifier. In addition, some of the features could be explored more to extract more relation between features, which is not much in data-set.

REFERENCES

[1] Decision Tree Classification. https://scikit-learn.org/stable/modules/tree.html#.

[2] H-1B Specialty Occupations and Fashion Model. https://www.uscis.gov/working-united-states/temporary-workers/h-1b-specialty-occupations-dod-cooperative-research-and-development-project-work

[3] H1B Prediction Case Status prediction for H1B Visa Applications(2015-2018). https:

//www.kaggle.com/abishekanbarasan1995/h1b-case-status-prediction.

[4] H1B Visa Prediction by Machine Learning Algorithm. https://github.com/Jinglin-LI/H1B-Visa-Prediction-by-Machine-Learning-Algorithm.

[5] Here's how much the average American earns at every age. https://www.cnbc.com/2017/08/24/how-much-americans-earn-at-every-age.html.

[6] How to Handle Missing Data. https://www.kaggle.com/abishekanbarasan1995/h1b-case-status-prediction.

[7] Multiclass averaging. https://cran.r-project.org/web/packages/yardstick/vignettes/multiclass.html.

[8] Predicting H-1B visa status Python. https://www.datacamp.com/community/tutorials/predicting-H-1B-visa-status-python.

[9] The H-1B dream has turned into a bureaucratic nightmare. https://qz.com/india/1760097/in-trumps-us-the-h-1b-dream-has-become-a-bureaucratic-nightmare/.

[10] Understanding Interpreting Dataset. https://blogs.sas.com/content/subconsciousmusings/2018/03/09/understanding-interpreting-data-set/.

[11] H - 1B Denial Rate: Analysis of H-1B Data for first tow Quaters of FY 2019. https://nfap.com/wp-content/uploads/2019/08/H-1B-Denial-Rates-Analysis-of-H-1B-Data-for-First-Two-Quarters-of-FY-2019.NFAP-Policy-Brief.August-2019-1.pdf, 2019.

[12] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[13] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[14] H. He, Y. Bai, E. A. Garcia, and S. Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1322–1328. IEEE, 2008.