Machine Learning Engineer Nanodegree

Capstone Project

I. Definition

Predicting the outcome for H-1B Visa eligibility in the U.S.

Project Overview

Every year hundreds of thousands of international workers apply for H-1B non-immigrant visas in the United States. In order to be able to qualify for worker H1-B visas, a person needs to have a job offer from a U.S. based company. This is also the kind of visa usually requested by international students pursuing higher education in the country. This study aims to train a classifier based on features of the dataset to be able to predict whether a given request would be granted eligibility to the H-1B program. Given the number of people requesting visas every year - and the likelihood to increase over the next years despite political pressure - it would be interesting to analyze some of the existing data and provide a model that could help to understand successful over non-successful applications. This is handled as a multi-class classification problem as we have to identify one among different solutions, however the number of outputs used will be discussed given the fact that some of these results are more influenced by external factors and do not have a significant impact on the results. To approach this problem three different classifiers are trained and compared to identify the five most important features to tackle this problem. While prevailing wage is the highest weighted features as expected, part-time positions weight stronger than expected and the worksite does not necessarily affects the outcome of the application. Finally, a Logistic Regression classifier proved to be the best option among those analyzed to process this data considering time needed to train and predict as well as output produced.

Problem Statement

Immigration has always been one of the most important assets of developed nations. According to the United Nations [1] there is an estimation of 244 million international migrants worldwide, of which almost twenty percent (47 million) are in the United States. That means the larger immigrant population in the world. In order to deal with this worldwide immigration, governments usually offer many different visa categories an applicant could choose from in order to either visit or stay in the new country. One type of these applications come from skilled professionals looking for new opportunities. In the U.S. every year hundreds of thousands of skilled professionals apply for a non-immigrant visa H1-B to fill a temporary position. These positions are demanded by different industries and sectors all across the country. However, as previously reported (Sundararaman et. la 2017) some employers abuse this system to hire international workforce cheaper than its domestic counterparts. Some mechanisms to tackle this problem include merit-based and salary-based systems. Given that salary seems to be an important factor when applying for a visa, this paper attempts to analyze how important is the salary compared with other features and what are the next important factors to keep in mind when applying for a H1-B visa category. In order to achieve that, a comparison among three different classifiers will be presented: RandomForest, DecisionTree, and Logistic Regression. Other classifiers were also studied and later on dismissed for performance reasons. In summary, the proposed structure is following:

- Analyze the dataset with H1-B visa applications to visualize important facts and potential features.
- Perform statistical operations to understand minimum, maximum, and average prevailing wage as well as to identify possible outliers.
- Identify the five most important features and its weight on predicting new petitions.
- Implement three machine learning classifiers and compare its performance to predict H1-B visa eligibility.

Selecting the best algorithms to predict eligibility for new visa petitions might be useful in the future to work with larger and more comprehensive data. On the other hand, analyzing the characteristic of non-immigrant visa petitions and identifying the most important features can be helpful for both companies and applicants to avoid wasting time in less important variables and focus on improving eligibility for each new application.

H-1B visa process

When a foreigner candidate is willing to accept a job offer in a U.S. company, the employer needs to file a Labor Condition Application (LCA) with the US Department of Labor (DOL) before they can file a H-1B petition with the US Citizen and Immigration Services (USCIS). This company, also called sponsor, needs to file the ETA Form 9035 / 9035 E electronically with the DOL. A LCA contain the following fields:

- Job title
- SOC code: the occupational code associated with the job being requested for temporary labor condition, as classified by the Standard Occupational Classification (SOC) System
- Duration of the job position
- Information about whether the position is full time or not
- Number of job positions the LCA is applied for
- Rate of pay offered for the position
- Worksite, location of the job position
- Prevailing wage for the same position in the area. The wage is listed at annual scale in USD. The prevailing wage for a job position is defined as the average wage paid to similarly employed workers in the requested occupation in the area of intended employment. The prevailing wage is based on the employer's minimum requirements for the position

- Employer's and attorney contact information

The LCA is important overall because it aims to protect foreign workers and their fundamental rights in terms of work conditions, compensations and policies. Furthermore, it provides the applicant the right to know if the employer is planning H-1B petition filing. As part of public disclosure, DOL publishes LCA information about employer's petitions. Therefore, this data is LCA approval and denial by the Department of Labor instead of H-1B filing by USCIS. However, it is a requirement to have LCA certification in order to file H-1B visa petitions.

Metrics

For this kind of problem the evaluations metrics used usually are: precision, accuracy, recall, and F1 (F-Score), where precision provides a ratio of right outcomes or true positives (TP) over all predicted observations, which means high precision will have low false positive (FP) rate. Accuracy, is just a ratio of correct outcomes over all observations and its value is more relevant when datasets are more even (FP and FN are similar). Recall, or sensitivity, is a ratio of right predicted observations over all observations that are true (or false). In other words, of all the international workers who were 'certified', how many were correctly labeled. Finally, F1 is a relationship or weighted average of precision and recall values in order to take FP and FN into account. Having said that and given the nature of this unbalanced dataset, it will probably make sense to dismiss accuracy and use instead precision, F-score or Confusion Matrix.

$$Precision = \frac{tp}{tp + fp}$$

$$F\ - score = \frac{precision\ .recall}{precision\ +\ recall}$$

II. Analysis

Data Exploration

The Office of Foreign Labor Certification (OFLC) from the U.S. Department of Labor provides public data about the H1-B visa. The dataset includes petitions from 2011 – 2016 and has around 3 million records [3][4][5].

The dataset [3] includes the following columns from which the features would be extracted:

1. CASE_STATUS: Status associated with the last significant event or decision. Valid values include “Certified,” “Certified-Withdrawn,” Denied,” and “Withdrawn”.
2. EMPLOYER_NAME: Name of employer submitting labor condition application.

3. SOC_NAME: Occupational name associated with the SOC_CODE. SOC_CODE is the occupational code associated with the job being requested for temporary labor condition, as classified by the Standard Occupational Classification (SOC) System.
4. JOB_TITLE: Title of the job
5. FULL_TIME_POSITION: Y = Full Time Position; N = Part Time Position
6. PREVAILING_WAGE: Prevailing Wage for the job being requested for temporary labor condition. The wage is listed at annual scale in USD. The prevailing wage for a job position is defined as the average wage paid to similarly employed workers in the requested occupation in the area of intended employment. The prevailing wage is based on the employer's minimum requirements for the position.
7. YEAR: Year in which the H-1B visa petition was filed
8. WORKSITE: City and State information of the foreign worker's intended area of employment
9. lon: longitude of the Worksite
10. lat: latitude of the Worksite

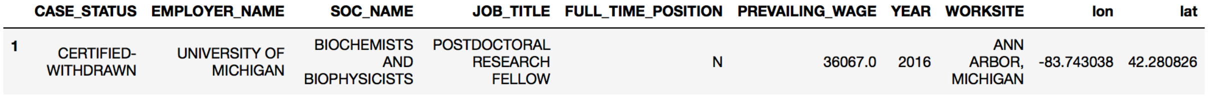
Among them, we have the following types of features:

- Categorical: {1, 3, 5}
- Text: {2, 4, 8}
- Numerical: {6, 7}

As expected, analyzing the dataset shows an unbalanced distribution behavior since most of the cases have been classified as 'certified' (fig. 2). Accuracy might not be representative as there is a strong possibility to be accurate on the 'certified' state, hence evaluation metrics were adjusted. Additionally, prevailing wages were limited until $1.000.000 to avoid dealing with outliers. Below are some statistics about the data along with a sample of what the raw data looks like:

```
Statistics for applications:

Minimum wage: $0.00
Maximum wage: $999,611.00
Mean wage: $70,407.48
Median wage $65,000.00
Standard deviation of wage: $26,714.00
```



	CASE_STATUS	EMPLOYER_NAME	SOC_NAME	JOB_TITLE	FULL_TIME_POSITION	PREVAILING_WAGE	YEAR	WORKSITE	lon	lat
1	CERTIFIED-WITHDRAWN	UNIVERSITY OF MICHIGAN	BIOCHEMISTS AND BIOPHYSICISTS	POSTDOCTORAL RESEARCH FELLOW	N	36067.0	2016	ANN ARBOR, MICHIGAN	-83.743038	42.280826

Figure 1 - Example of raw data

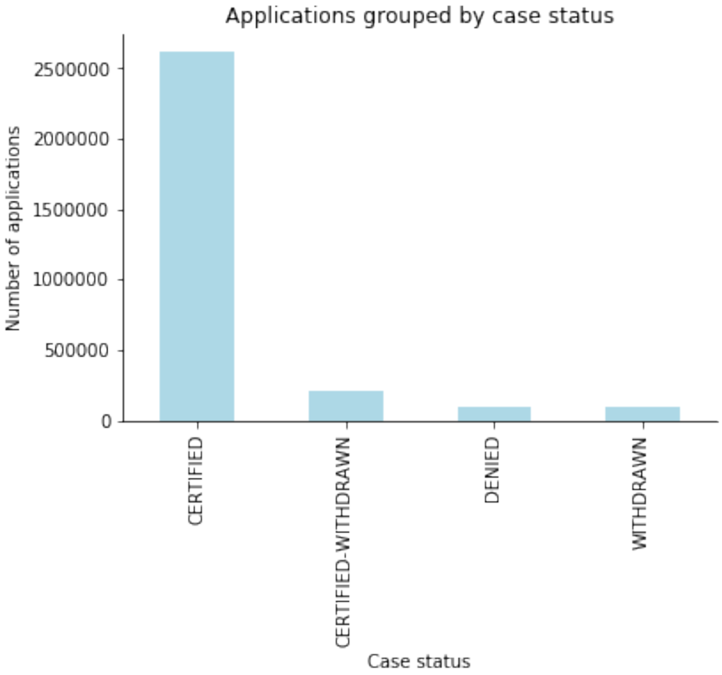


Figure 2 - The four case status of an application: certified, certified-withdrawn, denied, or withdrawn. The dataset is imbalanced with over 87% of cases labeled as certified.

Exploratory Visualization

Since the beginning of the data collection, in 2011 for this dataset, applications for H1-B visas are steadily increasing. Even after the new administration in the U.S. announced it's possible removal.

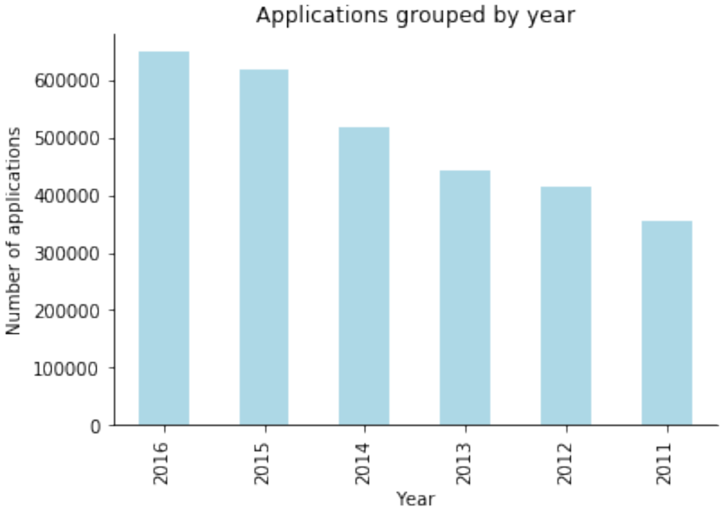


Figure 3 - Observations show how the number of applications for H1-B visas increased during the period from 2011 to 2016.

Considering the job titles that collect the highest number of applications nine out of the first ten are directly related with computer science (fig. 4), which clearly indicates a trend of what kind of skills are being 'imported' from abroad and probably on a domestic shortage.

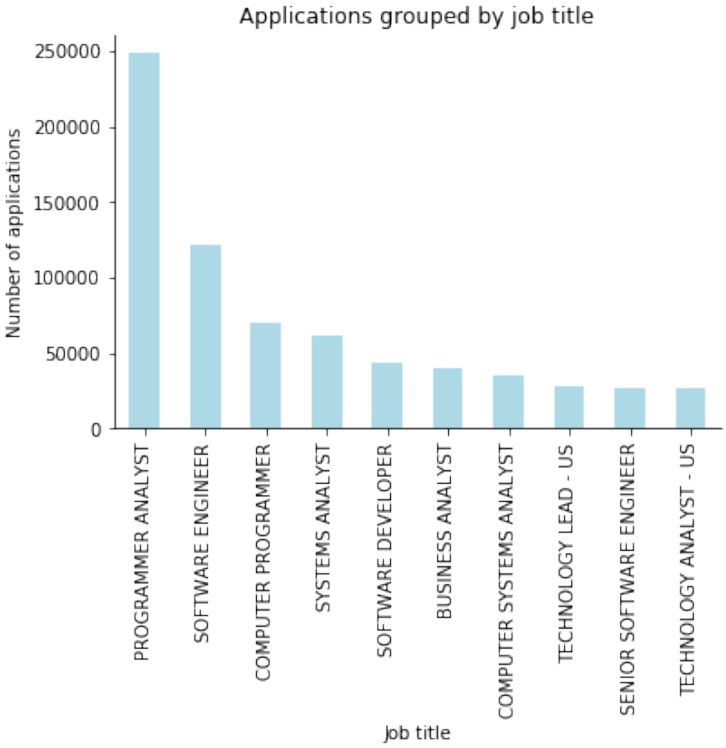


Figure 4 - All applications grouped by prevailing wages. The median wage would be $65,000 and the mean $70,408.1

On the other hand despite technology jobs being the most demanded, Silicon Valley does not account for the worksite receiving most of these applications (fig. 6). It is actually right after Houston and both of them are a long way behind New York, which doubles the number of applications of the second place. That being said, the East and West Coasts do account for most of the applications. In the last few years, Houston has been fueled by new jobs on the tech and oil industries and dominated the top of the list of America's fastest growing cities by Forbes [2].

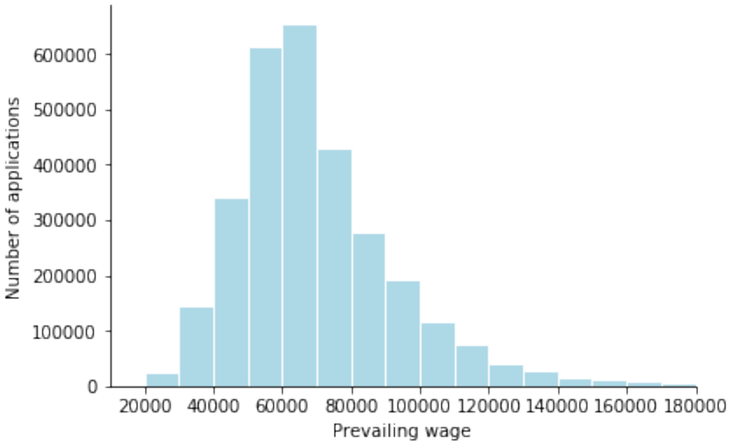


Figure 5 - All applications grouped by prevailing wages. The median wage would be $65,000 and the mean $70,408.19

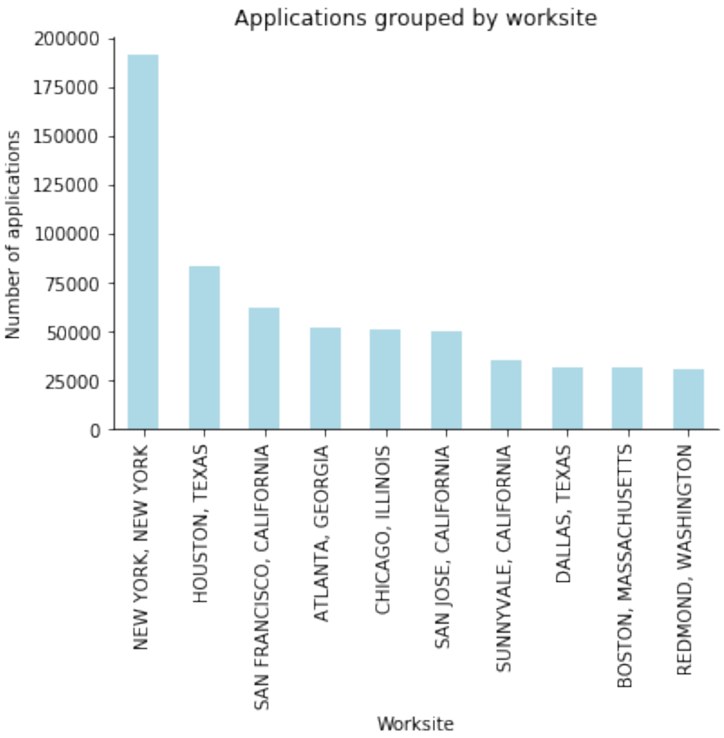


Figure 6 – Number of applications based on worksite. The West and East Coasts represented by New York and San Francisco as expected along with Houston, Texas at the top-three.

Algorithms and Techniques

The following so-called supervised shallow learning classifiers are being tested and compared against each other: Random Forest, Decision Tree and Logistic Regression. Gaussian Naïve-Bayes was used as benchmark on a reduced dataset and SVC was also tested but dismissed per performance reasons. Note that this study is not exhaustive and there are many other algorithms that could have been tested out but this analysis focus on those, which reported better performance. Moreover, since the dataset is already labeled for the different outcomes, only supervised algorithms were primarily chosen. Gaussian Naïve-Bayes is a basic but powerful algorithm, which does not require much tuning to achieve reasonable results, though it is usually not the most effective efficient option. On the other side, Decision Tree and Random Forest classifier have shown to provide very good results based on the metrics defined and the dataset provided. However, much of its performance comes at an expensive cost of training time. Logistic Regression showed good results in a much shorter training time. Below there is a deep dive into how each of these algorithms work:

- Logistic Regression (logit): It's named after its core function, the logistic function. In statistics, logistic regression is used to classify binary problems, however for those problems with more than two discrete outcomes, the classification method used is called 'multinomial logistic regression'. It is basically a model to predict the probability of a categorical dependent variable (in our problem the case_status) based on independent variables (features).
  Logit performs well and it is relatively easy to implement and train, which makes it a good starting point and even a benchmark model to later move on to more

complex algorithms. It is a generalized linear model so it would be not suitable to solve non-linear problems. Below is the definition of the logistic function σ, which takes any real input *t* and provide an output between 0 and 1:

$$\sigma(t) = \frac{1}{1 + e^{-t}}$$

- Decision Tree (DT): They can be used both for regression and classification problems and its functionality is not very complex. Depending on the dataset it can achieve good results with a little parameter tuning and without much training and predicting time.

  The representation of this model is a binary tree and its goal is to predict the value of a target variable through different input variables. It consists of a main node, middle nodes and ultimately final nodes or leafs. Every node would be a layer. In order to have more layers, the model will need more features. The more features added, the deeper the tree and its nodes (layers, levels) will be. However, it needs to be careful with going deeper. Too much nodes will classify the information close to 100% accuracy but also will cause the model to overfit. This is not desired as the model fails to generalize data that has not seen before and predict whether, as in this problem, the application would be eligible or not.

  A tree learns splitting the input set into subsets or partitions until either it has been divided into new classes that are pure (only members of a given class) or some criteria is met. However, not always a model is able to achieve new pure classes so some percentage of impurity might be tolerated. That is also known as *gini impurity* and it can be represented by summing the probability p_i of an item with label *i* times the probability of a mistake categorizing the same item.

$$\text{gini impurity:} \sum_{k \neq 0} p_k = 1 - p_i$$

  While the tree is been built, it needs to decide what feature will be used to for the next division. The best choice would be to pick a feature that gives as a result the best information gain and thus the less impure node. In order to measure this impurity, entropy will be used as it is defined as the randomness of elements.

$$\mathrm{H(T)} = -\sum_{i=1}^{J} p_i \log_2 p_i$$

  Information gain, then, will be the change in entropy from the current state to the next proposed state:

$$IG(T,a) = H(T) - H(T|a)$$

- Random Forest (RF): As well as Decision Trees, Random Forest can also be used for both classification and regression problems. Random Forest classifiers are in general more complex than Decision Trees and Logistic Regression models but their predictive accuracy could also be higher for specific problems. On the other hand, they might be much slower to train and predict given the fact that they are a combination of trees. The general idea is to build a random set of Decision Trees and provide a random subset of the training data to each of the trees, in a way that it helps to correct their tendency to overfitting. In summary, a large number of trees are generated and then the most popular class is voted.

Benchmark

To the best of author's knowledge there has not been many public algorithms dealing with the same dataset, other than those mentioned which focus on the exploration of the data analysis or whether there have been abuses on H-1B visa petitions. Therefore the model will be compared to the best naïve selection such as guessing the mean, the majority class, and against a simple model such as Gaussian Naïve-Bayes. Below it’s the classification report with actual scores of benchmark represented by the null hypothesis, which is to guess all applications have been certified. Case status 1: Certified, 2: Certified-Withdrawn, 3: Denied, 4: Withdrawn.

```
          Precision    Recall   F-score    Support

 1          0.87      1.00      0.87   2613575
 2          0.00      0.00      0.00    202652
 3          0.00      0.00      0.00     92257
 4          0.00      0.00      0.00     89412

avg / total  0.76      0.87      0.78   2997896
```

III. Methodology

Data Preprocessing

The dataset comes in CSV format and it is loaded directly into pandas dataframes. It has been already preprocessed where null and invalid values were removed and commas on PREVAILING_WAGE column were not present. Some fields require some processing though, for example PREVAILING_WAGE and YEAR are transformed and scaled as numeric values, FULL_TIME column is converted to binary, CASE_STATUS to numerical, and categorical fields are dealt by creating dummy variables. After that, outliers were identified and removed. Also, the data was checked for skewed distributions on the PREVAILING_WAGE column but no major case was found.

Implementation

As previously discussed, this is a multi-class problem which focus to predict whether a specific application would be eligible (certified) for a H-1B visa. It is important to note that the dataset offered many different case status: 'certified', 'denied', 'certified-withdrawn', 'withdrawn', 'rejected', 'invalidated' and 'pending quality and compliance review'. Since only fifteen cases are being labeled as pending quality, two rejected and finally just one has been invalidated we don't take these three status into account. They represent altogether only 18 out of 2.997.883 applications and hence should not provide much information to the analysis.

Although the dataset does not have too many columns, some of them can receive lots of different values, which make them interesting to analyze and extract additional features. However, this proved to be a difficulty during experiments as in order to extract more interesting relationships, more powerful systems are required and hence we limited the set of categorical data. The analysis was done working with 2.134 features from 2.997.866 application requests.

Several tests were conducted to include WORKSITE as categorical variable too but given that there are so many different worksite locations provided, 18.592, it was not possible to deal with them using the full dataset, thus they were studied in reduced versions.

The data was also shuffled and split into training (0.8) and test (0.2). Hence, when predicting results only 10% of the training set was used to provide an estimation result without having to process all training data. After all classifiers were defined and initialized under clf_X variables, they were sent as parameter into the train_predict pipeline. Each classifier was fit using X_train and y_train sets and afterwards used to predict the X_test set. Once predictions were created, they were compared with the so-called true set, in our case y_test, in order to calculate the precision and f-scores. A summary of results is discussed on the next section. The implementation code can be seen on GitHub [7].

Refinement

An important part of choosing an algorithm is to tune it properly, thus GridSearchCV was used to explore more comprehensively the best parameters of the following models: Logistic Regression, Decision Tree and Random Forest classifiers. In the former the following C values were analyzed using param_grid = {'C': [1, 10, 100, 1000]} but no significant improvements were observed. Results remained the following, 0.8723 in precision score and 0.8952 in F-score. While on Decision Tree, earlier implementations on a reduced dataset of 500.000 applications, showed that tuning only improved a few decimal points and it took extensive time to perform and therefore further tuning was dismissed on the full dataset with 3 million entries for this classifier. On Random Forest, parameters analyzed were 'max_leaf_nodes': [5, 10, 50, 100 ,200, 500]. Although precision did improve, F-score slightly decreased in respect to the non-optimized classifier meaning that tuning this parameter might not always improve both metrics. After some experiments with different parameter values and random_states the results remained similar.

Finally, SVC proved to be much more time-consuming without significantly improving results on a reduced version of the dataset, therefore further tuning was not implemented.

IV. Results

Model Evaluation and Validation

After further analysis and consideration of the different models, the logit classifier seems to be the best choice to solve the problem on the given dataset. In order to do that, different experiments were run comparing Decision Tree, Random Forest, and Logistic Regression. Below each algorithm's evaluation to face the H-1B eligibility problem:

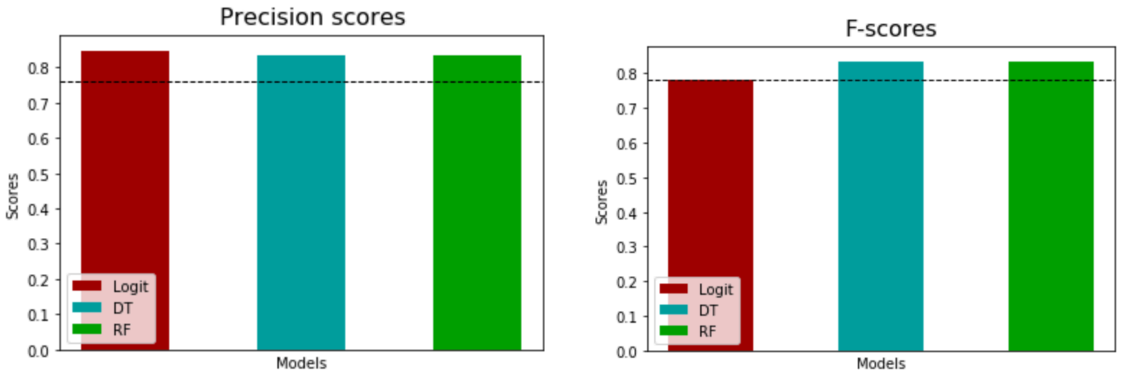


Figure 7 – On the left side, precision scores for the three models proved to be much better than benchmark (dashed line). On the right side, F-scores showed better results for Decision Tree and Random Classifiers, however Logit still delivered slightly above the benchmark.

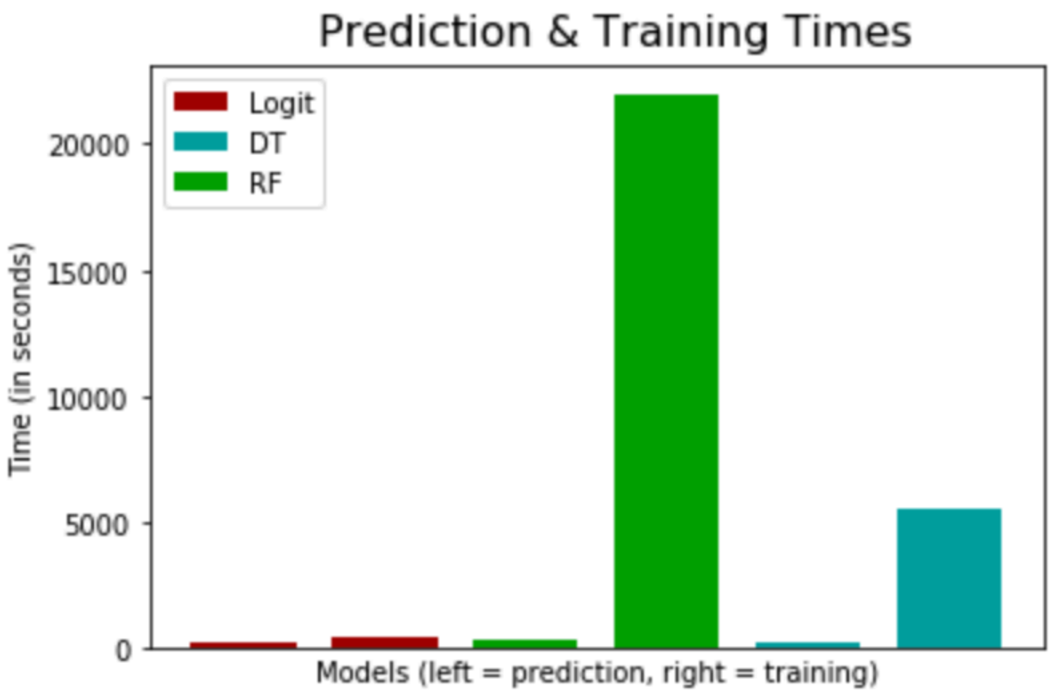


Figure 8 - Performance scores of Logistic Regression, Decision Tree and Random Forest classifiers during training and prediction stages. Time reported is in seconds and goes up to 20.000 for Random Forest training, while during prediction the highest value is below 60.

Logistic Regression: This classifier proves to bring the best results for this problem and its given dataset. It was able to process all 2134 features and train the model drastically faster than the Decision Tree classifier. Prediction time was also better, though precision and F-scores are slightly below. Metrics observed were, precision score: 84.69% and F-score: 78.26%.

Decision Tree: It took much more to train than Logistic Regression but it was almost four times faster than Random Forest, which means it beats the latter without decreasing much its performance. It was the second best classifier according to our experiments and its precision and F-score are 83.23% and 83.43% respectively.

Random Forest: Results are slightly better than Decision Tree at an expense of a much higher training time, which is more than double of the time required by DT. Moreover, predicting times are more demanding although the difference for this dataset is not very significant. Precision score registered is 83.16% and F-score 83.39%.

SVM and XGBoost were also studied but SVM showed to be inefficient to deal with this kind of problem and was therefore no longer considered. On the other hand, XGBoost seems to be a strong option compared to other 'boosters' such as Adaboost, which was also tested and did not perform well for this dataset compared with tree classifiers' performance. However, it requires some different preprocessing on the dataset where the data comes in a different format and style, which would make it not a very suitable solution compared to the other classifiers analyzed.

Ultimately, the models were validated by assessing its cross-validation score with the following results:

```
Logit cross validation score: [0.7818  0.7823  0.7826  0.7825  0.7821]
R^2 score: 0.872737582975
Mean score: 0.782323843597

DT cross validation score: [0.8198  0.8196  0.8210  0.8193759  0.8209]
R^2 score: 0.872737582975
Mean score: 0.820159585207

RF cross validation score: [0.8190  0.8179  0.8194  0.8190  0.8197]
R^2 score: 0.872737582975
Mean score: 0.81907770776
```

Justification

Given the data analyzed in the previous section, we have reasons to believe the model is robust and the solution is significant enough to have solved the problem.

```
                 Results Report

          Precision    Recall  F-score

      Logit     0.84      0.87      0.78
      DT        0.83      0.87      0.83
      RF        0.83      0.87      0.83

      Benchmark 0.76      0.87      0.78
```

The results observed from the classifiers confirm that the three proposed final models outperform the results reported by the benchmark. Although, in the case of logit the F-score and recall values reported are rather similar with a slight improve of 0.0026 and 0.0010 respectively, its precision score is the best registered among the models.

V. Conclusion

Free-Form Visualization

Although prevailing wage is the most important factor to get eligibility and eventually an approval for a H-1B visa, understanding the nature of the data can help to bring interesting relationships. For example, even though we are aware that IT jobs are on the rise, it's interesting to see which one of them have a highest probability of acceptance. Positions in the top-five of most predictive features are not only related with IT but directly with software engineers or software developers, which might be a good advantage for applications in these areas. Additionally, whether a position is full-time proved to be one of the most important features in additional experiments. On the other hand, some information provided might not very useful for a candidate such as year of application. Although it seems to be one of the most important features, it cannot directly improve a candidate's chance. In this case, further research with more specific data on time of the year can help to identify better seasons for applications.

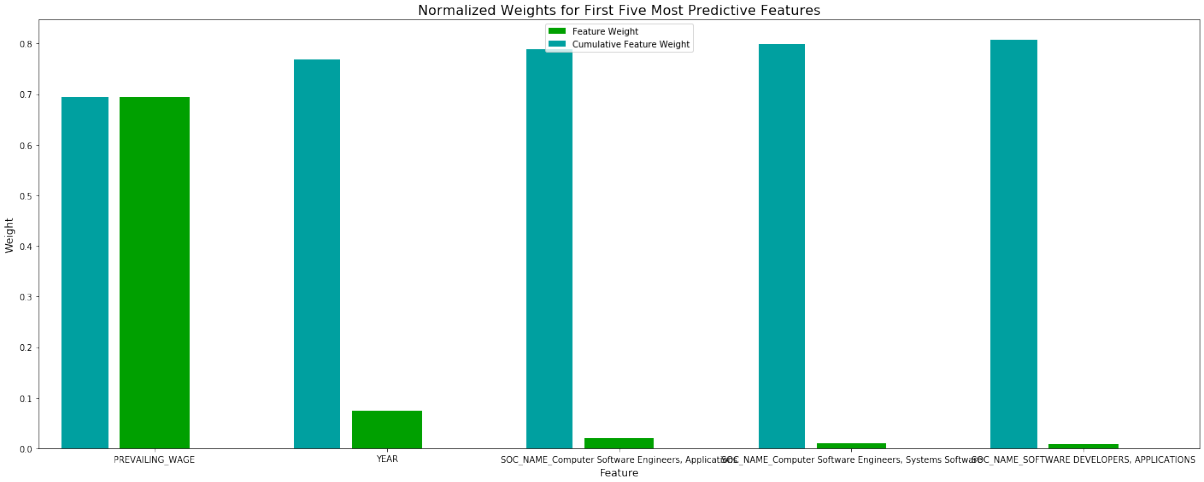


Figure 9 - Top-five of most important features. As previously showed, prevailing wage is the strongest feature in order to predict H-1B eligibility with almost 0.7 weight. Right after that, year was the second most predictive feature followed by job roles related to software Engineers and Software Developers.

Reflection

The project started by loading and exploring the data to understand its structure and identify potential missing or inconsistent values (e.g. to be removed / filled with zeroes and so on). Here was also possible to identify outliers and manage them. Afterwards, features were identified and created. Afterwards all results needed to be validated to prevent overfitting to the data. Therefore, the dataset was separated initially into training, cross validation, and test set, where cross validation was used to tune the algorithm and the test set to evaluate the metrics. Once validation has been finished, the metrics defined were evaluated in order to assess the algorithm's performance against the benchmark.

An important aspect learned through this project was the ability to gather new knowledge beyond of the expected results. Working on how to manipulate and visualize the data has been a significant step towards reaching a better understanding of the data in order to apply machine learning algorithms.

A particular challenge was to manipulate and process such a large dataset on a personal computer. Online platforms, like Kaggle, were also not able to process big datasets returning MemoryError. Furthermore, there was an attempt to migrate the dataset along with the jupyter notebook to Amazon Web Services in order to leverage more power from cloud clusters. However, after a few failed attempts with different EC2 instances, which finally led to further improvements on the algorithms, the classifiers ended up trained locally. After a few days though, a confirmation from Amazon came stating that the limit was increased to 1 p2.xlarge, which is technically better suited for ML algorithms but no further experiments were implemented in the cloud.

Improvement

As further improvement, it would be interesting to process more features such as worksite to see how it impacts the final features weight using feature_importances_ as in this study the analysis of the worksite was performed over a reduced version of the dataset and did not show to be a top feature. Work with more complete datasets where information about country of origin or others types of visa requested can have a richer impact. With more resources available it could be worth to further tune algorithms to see whether a significant difference in the metrics would take place. Furthermore, in case the dataset is adjusted accordingly, XGBoost can be a good option to consider and compare its performance with present results.

References

[1] An analysis of nonimmigrant work visas in the USA using Machine Learning:
http://export.arxiv.org/pdf/1711.09737

[2] America's Fastest-Growing Cities in 2015:
https://www.forbes.com/sites/erincarlyle/2015/01/27/americas-fastest-growing-cities-2015/

[3] H-1B visa applications dataset: https://www.kaggle.com/nsharan/h-1b-visa

[4] H-1B application process: step by step guide:
http://www.immi-usa.com/h1b-application-process-step-by-step-guide/

[5] Disclosure of data: https://www.foreignlaborcert.doleta.gov/performancedata.cfm

[6] Classifying US Visa applications:
http://charlesfranzen.com/posts/machine-learning-classifying-us-visa-applications/

[7] GitHub: https://github.com/zRapha/h-1b