Predictive analytics for classification of immigration visa applications: A discriminative machine learning approach

by

Sharmila Vegesana

B.Tech., Jawaharlal Nehru Technological University, India, 2012
M.Tech., GITAM University, India, 2014

A REPORT

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Computer Science
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2018

Approved by:

Major Professor
Dr. William H. Hsu

Copyright

Abstract

This work focuses on the data science challenge problem of predicting the decision for past immigration visa applications using supervised machine learning for classification. I describe an end-to-end approach that first prepares historical data for supervised inductive learning, trains various discriminative models, and evaluates these models using simple statistical validation methods.

The H-1B visa allows employers in the United States to temporarily employ foreign nationals in various specialty occupations that require a bachelor's degree or higher in the specific specialty, or its equivalents. These specialty occupations may often include, but are not limited to: medicine, health, journalism, and areas of science, technology, engineering and mathematics (STEM). Every year the United States Citizenship and Immigration Service (USCIS) grants a current maximum of 85,000 visas, even though the number of applicants surpasses this amount by a huge difference and this selection process is claimed to be a lottery system. The data set used for this experimental research project contains all the petitions made for this visa cap from the year 2011 to 2016.

This project aims at using discriminative machine learning techniques to classify these petitions and predict the "case status" of each petition based on various factors. Exploratory data analysis is also done to determine the top employers, the locations which most appeal for foreign nationals under this visa cap and the job roles which have the highest number of foreign workers. I apply supervised inductive learning algorithms such as Gaussian Naïve Bayes, Logistic Regression, and Random Forests to identify the most probable factors for H-1B visa certifications and compare the results of each to determine the best predictive model for this testbed.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to express my sincere gratitude to my committee members William H. Hsu, Torben Amtoft, Mitchell Nielsen for taking time to serve on my committee and their encouragement in the process of this project.

I am especially indebted to my major advisor and mentor Dr. William H. Hsu for believing in me and for his constant support from my very first day here at Kansas State University. His passion to expose students to various facets of machine learning and artificial intelligence and the effort put in organizing the weekly meetings has led me to gain extensive knowledge in these fields. Thank you for everything!

Also, a special mention to Dr. Jorge Valenzuela, for being a splendid mentor during my Teaching Assistant phases.

Maa, Daddy, you guys are my guideposts for everything in life. Pallavi, my little rainbow, thank you for being my eternal source of happiness. You guys are my strength. Teja, I can't thank you enough for all the sacrifices you made for me. Your encouragement was the very first step towards this goal. Despite being far away, you all have walked every step of this journey with me and have been my constant rays of sunshine. I would be nowhere without any of you; your love and support have fueled my ambition to achieve my dreams. My cousins, Bharath and Praneeth need a special mention for their timely help and advice during the sticky phases of my Master's degree. I'm also grateful to my grandparents and in-laws for their love and belief in me.

And last, but in no way the least, I would like to thank my friends and companions through this journey, Poojitha, Sneha, Sindhu, Sravani, Nikhil, Pruthvi, Pavan, and Nithin for their support and friendship. Every moment spent in the past two years with you all is memorable.

Chapter 1 - Introduction

This chapter discusses the problem statement, the goals and a synopsis of this experimental research project.

1.1 Problem Definition

This project deals with the task of predicting the outcome of immigration visa applications by using machine learning methods. The prediction task is treated as one of supervised inductive learning of classification functions. Exploratory data analysis was also performed to analyze the main characteristics of the data set. A prediction model was developed which trains on historical data of the same application domain and uses machine learning classifiers to predict the outcome of visa petitions.

The H-1B visa allows companies and organizations in the United States to employ foreign workers in specialty occupations that require major technical expertise in fields such as accounting, architecture, engineering, finance, information technology, mathematics, medicine, science, etc. A bachelor's degree or equivalent work experience are some of the basic requirements for a H-1B visa. A job offer should also be forwarded to the worker from the employer filing the petition.

Each fiscal year the United States government grants a total of 85,000 new H-1B visas; this includes 65,000 visas for overseas workers with at least a bachelor's degree and 20,000 visas available for those employees with an advanced degree in a specialty field from an accredited United States academic institution. The allotment of these visas is said to be a lottery system not depending on any specific criteria. Thus, a foreign national is not guaranteed selection for an H-1B visa merely because he or she fulfills all the qualifying criteria. This work aims to use machine

learning on known factors to make predictions under this uncertainty, with accuracy better than that achieved using the prior probability.

1.2 Goals and Technical Objectives

The goal of this project was to develop a supervised inductive learning model which adopts methods of classification to predict the results of the filed applications. Various statistical evaluation methods were used to determine the accuracy of these predictive models.

The visa petitions were classified in to “Certified” or “Denied” classes formed from the “Case Status” attribute of the data set. The data present dates from the year 2011 to 2016. The data from the earlier years was used to training the machine learning models and the data from years 2015-2016 was used to testing and validating the algorithms and their performance. Discriminative models such as Logistic Regression and Random Forests were the main ones used for binary classification in this project. Data from the United States Bureau of Labor Statistics was used for computing the state-wise salary median, these values can be used to play an important role in classification. The overall objective was to use three main attributes of the data set and see how they influence the accuracy, precision and recall of the learning models and see which model works best for predicting the outcome of a visa petition.

1.3 Project Synopsis

Each petition in this data set has various factors such as ‘Job Location’, ‘Employee Role’, ‘Job Category’, ‘Salary’, etc. The supervised inductive learning algorithms were trained using examples from historical data over a chosen range of years, and predict the outcome of the attribute – “Case Status”. Data from later years was used as a test data set for these learning models. The

machine learning and data analysis software package *Scikit-learn*, written in the Python programming language, was used to implement the training and classification functions for these models. The results of the algorithms were compared using the confusion matrix and the ROC curves of these algorithms and scope of improvement was checked.

Chapter 2 - Background and Related Work

This chapter introduces the data set used for this research project in detail and describes the claims made regarding this data set in the recent past. Background information regarding the algorithms used in this work is also provided.

2. 1 Literature Survey

2.1.1 Classification for Decision Support: General Methodology

A common scenario in machine learning tasks involves building a classification model that is trained on a training data set, producing a model that is then used to predict the probability of data belonging to one of the classes of the test data set. The trained model solves a classification task, producing predictions of a target variable over previously unseen inputs, which represent new cases. Since historical data is used to train the model this falls under the supervised learning techniques. This classification methodology can predict the classes in two kinds of situations; one, when predicting the decisions made by a third party is involved. For example, some common applications of classification would be loan approval prediction, medical diagnosis, etc. And the second case is when classification is based on some historical ground truths. Example spam email filtering.

The main aim of this project is to predict the certification status of the visa applications. For this purpose, the concept of classification is used. That is, when all the attributes are provided, the classifying model is trained on historical data of this application domain and probability of rows belonging to one class or status over the other is determined. This aids in predicting decisions made by USCIS and to recognize if there is any specific pattern behind their decisions.

2.1.2 Application Domain: Immigration Visas

One of the crucial foundations for economic growth and a key input to the knowledge economy is highly qualified and proficient personnel. Several approaches were adopted by countries in the past to achieve this, they can either obtain the desired level of expertise through their education system or obtain workforce from overseas. To be on the competitive edge in world economy, countries need to possess high skilled employees in industries such as health care, engineering, science and information technology. The demand and supply of workers in fields such as the above-mentioned ones is unevenly distributed and poorly matched. Despite all these factors, United States has a competitive edge over the other nations for being a global center for academic training, and international students comprise of a considerate percentage of U.S. higher education enrollment.

Various U.S. employers aim to retain these international students and immigrate skilled workers from foreign nationalities for many reasons; they also face a lot of legal hurdles in doing so. According to the Immigration Act of 1990, the H-1B visa program allows companies based in U.S. to hire foreign nationals to work for a temporary amount of time in specialized occupations. Initially, the number of visas to be issued under this cap was set to 65,000, but it rose drastically to 195,000 between the years 2001 and 2003 and from the year 2004, it has been set to 65,000 per year plus an additional 20,000 for individuals with advanced degrees from U.S. accredited institutions. The H-1B visa once issued is valid for three years with one chance for renewal for another three years. Each initial petition is counted under the cap, but renewals do not account for this number. Over the past couple of years, this visa cap has been the talk of many political scenarios and cause for unrest among many individuals. Despite all this, the presence of other work-related permits such as L-1 and the heavy competition one must go through to obtain a visa

under H-1B, it remains the most popular one and the one most sought after by foreign students and non-immigrants alike. In this project, I try to device an algorithm which best predicts the outcome of an individual's visa petition given various socio-economic factors such as their salary, location, job type and the state median income.

2.2 Established Techniques and Validation Measures

Nilson (2005) describes the science of machine learning often refers to the changes in system behavior because of certain criteria; that perform tasks associated with artificial intelligence (AI). Such tasks involve recognition, diagnosis, planning, robot control, classification, prediction, etc. These changes might be either categorized as enhancements to already performing systems or ab-initio synthesis of new systems.

Let us say there is a function *f*, and the task of the machine learning model is to predict what it is; and *h* forms our hypothesis about the function to be learned. Both *f* and *h* are functions of a vector-valued input $X = (x_1, x_2, \ldots x_i, \ldots, x_n)$ which has *n* components (in Fig 2.1). *h* is implemented by a model which takes in *X* as input and produces *h(X)* as the output. We assume a priori that the hypothesized function, *h*, is selected from a class of functions *H*. Sometimes we know that *f* also belongs to this class or to a subset of this class. We select *h* based on a training set, Ξ, of *m* input vector examples.

Many important details depend on the nature of the assumptions made about these entities.

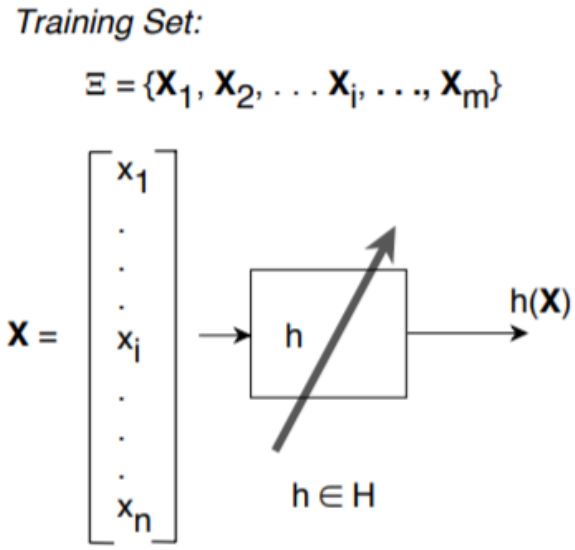

Fig 2.1: Input-Output Function *[Adapted from Nils J. Nilson, 2005]*

2.2.1 Discriminative Models

Discriminative models use conditional probabilities of a test label y over a feature set x to perform the machine learning task. Thus, the joint probability P(x,y) is not possible. The model can be trained over a given set of features and the loss function can be minimized by mapping inputs to outputs.

2.2.2 Classification Algorithms

When the goal of a problem is to learn how to map inputs from X to outputs from Y, where $Y \in \{1, \ldots C\}$, where C is the number of classes. When the number of output classes is two, C = 2, then it called Binary Classification. If the value of C > 2, then the classification is called Multiclass Classification. We use binary classification in this project. There are many classification methodologies present for binary classification such as Support Vector Machines, Naïve Bayes, Decision Tree, etc. Not all the algorithms are suited for all problems and some result in over-fitting

or under-fitting. In this project, Logistic Regression, Random Forests, Gaussian Naïve Bayes and k – Nearest Neighbors are implemented and their results are compared.

2.2.1.1 Logistic Regression

This algorithm assigns observations to discrete sets of classes and uses a sigmoid function to return a value to map two or more classes. There are various types of Logistic Regression such as:

a. Binary
b. Multi
c. Ordinal

Binary Logistic Regression is used in this project.

Sigmoid Function is used in Logistic Regression and this maps a value to another value and these values range from 0 to 1. The sigmoid function is given by:

$$S(z) = \frac{1}{1 + e^{-z}}$$

Fig 2.2 Sigmoid Function

Here *S(z)* is the output between 0 and 1, *z* is the function's input. And *e* is the natural log's base. A threshold value called the decision bound is selected in order to map the probability score which the classification function returns to a discrete class.

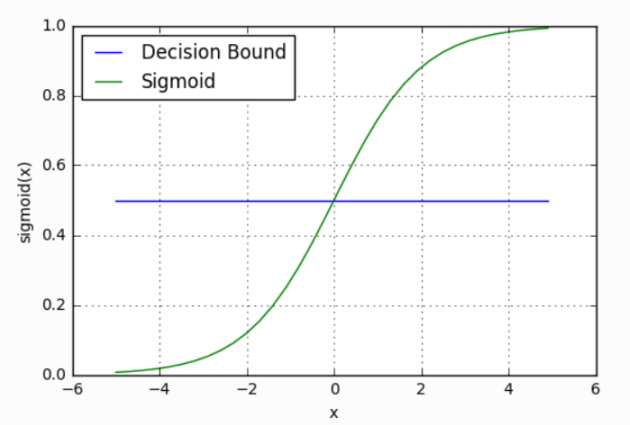

Fig 2.3 Logistic Regression

From these sigmoid functions and decision boundaries, we can compute the prediction outcome of the classification by the Logistic Regression model.

2.2.1.2 Random Forests

Random Forests are an ensemble model of machine learning with their roots in Decision Trees. These decision trees individually may over fit the data set and thus they come together to form a much stronger model. Numerous decision trees are first built and based on these by performing random sampling of the attributes, a group of decision trees are assembled to form a Random Forest.

The process of forming Random Forests involves a process of bootstrap aggregating or bagging the data.

The process of Random Forests is as follows:

a. Decision Tree Learning: Decision Trees are built and if they overfit then the process of building Random Forests begins.

b. The data is bagged or bootstrap aggregating is performed on it. For example let us say there exists a set of features X: {x_1, x_2,...x_n} and labels Y={$y_1 \ldots y_n$}, these are repeatedly bagged up to B times and a random sample of the replacement is chosen. The function can be given by:

$$\hat{f} = \frac{1}{B} \sum_{b=1}^{B} f_b(x')$$

Fig 2.4 Random Forests

These random forests do not overfit the data any more.

2.2.1.3 Naïve Bayes

Naïve Bayes methods are supervised learning methods based on the Bayes' Theorem and is perhaps one of the most widely used classification model to deal with real world problems such as spam mail filtering, classifying text articles into classes. To put the description in layman's terms, Naïve Bayes assumes the features are all independent of each other and calculates the probabilities of each attribute and it selects the outcome with the highest probability. Assume there are feature vectors $X_1 \ldots X_n$ and a class variable Y. Bayes' Theorem states the following:

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y) P(x_1, \ldots x_n \mid y)}{P(x_1, \ldots, x_n)}$$

Assuming the features are independent,

$$P(x_i | y, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) = P(x_i | y),$$

For each i value, this can be rewritten as,

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y) \prod_{i=1}^{n} P(x_i \mid y)}{P(x_1, \ldots, x_n)}$$

Given constant feature values for a particular instance - in this application domain, a visa application - the Naive Bayes Assumption yields following equations for classification by finding the most probable label given an observed instance:

$$P(y \mid x_1, \ldots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

$$\Downarrow$$

$$\hat{y} = \arg\max_y P(y) \prod_{i=1}^{n} P(x_i \mid y),$$

Fig 2.5 Naïve Bayes Formula *(Adapted from the official Scikit-learn website)*

Then Maximum-A-Posteriori (MAP) estimation is used. Naïve Bayes is one of the fastest classification algorithms and is quite good in classifying real world data sets.

2.2.1.4 k-Nearest Neighbors

The k-Nearest Neighbors (k-NN) classifier can be used to classify data of various natures and one of its most important traits is its versatility and robustness. k-NN is a supervised machine learning algorithm and is of discriminative nature. Let us assume x and y are feature and label respectively. The classification learning task is to find a function:

$$h : X \rightarrow Y$$

This function is consistent with training data (or is empirically good according to the evaluation criteria, e.g., a minimal loss function over labeled training data).

k-NN Classifier adopts the technique of forming a majority vote among the K most similar instances of a data set. Similarity between these data points is usually measured using the

Euclidean distance formula. There are however other distance formulas too such as Manhattan, Hamming etc.

$$d(x, x') = \sqrt{(x_1 - x'_1)^2 + (x_2 - x'_2)^2 + \ldots + (x_n - x'_n)^2}$$

Fig 2.6 Euclidean Distance

The value of K can be decided by the user. The higher the value of K, the more the system is resilient to outliers and the smoother the function is. The conditional probability of the classifier can be given by:

$$P(y = j|X = x) = \frac{1}{K} \sum_{i \in \mathcal{A}} I(y^{(i)} = j)$$

Fig 2.7 Conditional Probability of k-NN Classifier

An illustration of how the outcome of the classifier is shown below.

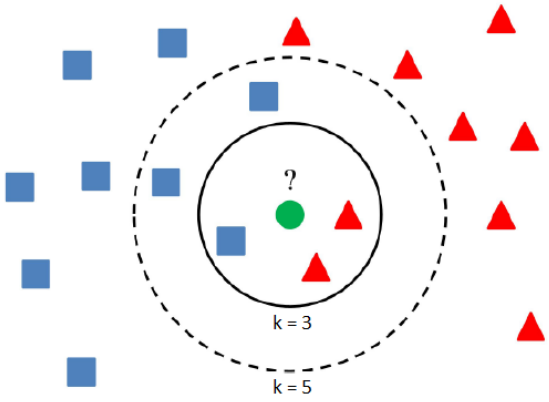

Fig 2.8 k-NN Classification

(Adapted from Tereza Abelova, An enhanced neighbor is better than any tree, (2017))

Chapter 3 - Implementation

This chapter encompasses an overview of the data, data preprocessing methods and implementation steps of this work.

3.1 Implementation Steps

The task that I used as a testbed for supervised inductive learning of classification was from a Kaggle data science challenge, uploaded by Sharan Naribole, 2017. Thus, the primary data set was downloaded from Kaggle and the supplementary data from the U.S. Bureau of Labor Statistics. In the below diagram, the process flow of the project can be found.

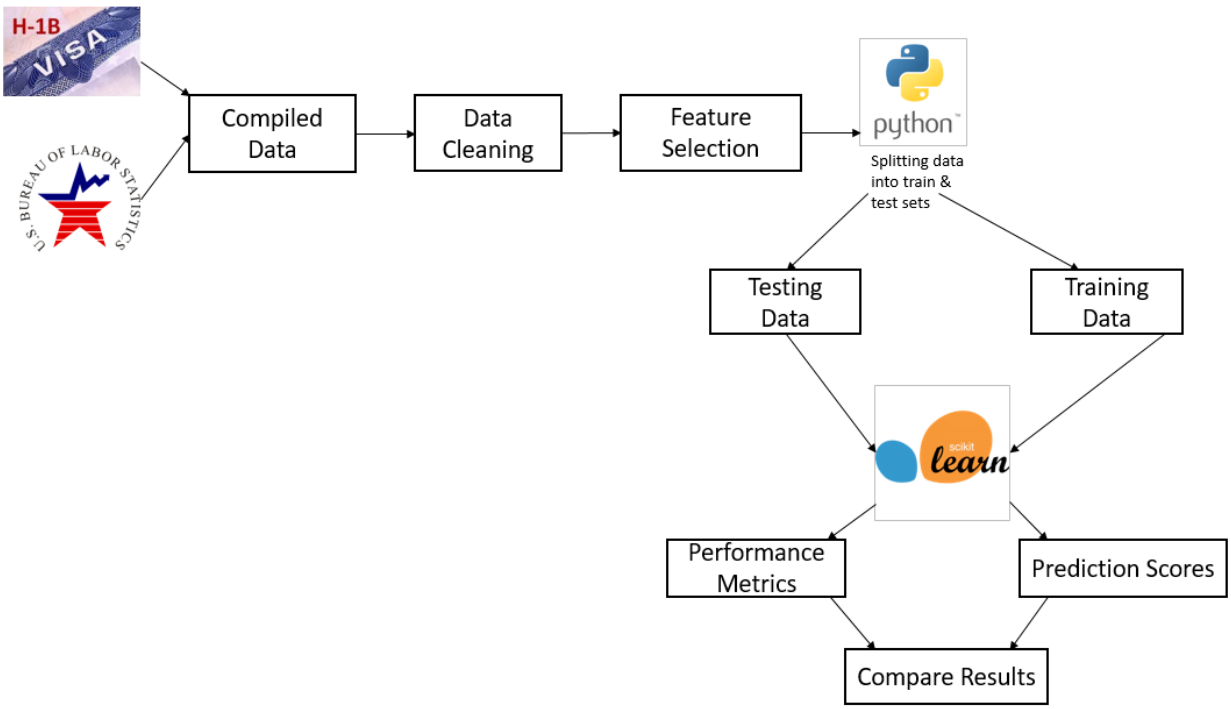

Fig 3.1 Process Flow of Project

The data was then combined. Data cleaning and preprocessing steps were followed on the data set. Depending on the task at hand, the features were divided into subsets. Python was used as the major programming language for this program and using in-built libraries, the data set was split into training and testing data in 70 and 30 percentages respectively. Scikit-learn was used to train the inductive learners and apply them to the data sets. The training data was fed to the learning models and the prediction scores of the classification task were obtained. These models were then tested on the testing data. The results are compared to find the best suited classifying model.

3.2 Overview of the Data

The data set was collected from Kaggle, and it was submitted by Sharan Naribole. All the data was originally extracted from the U.S. Office of Foreign Labor Certification's iCERT Visa Portal System. This is an electronic system for filling and processing of applications of H-1B non-immigrant workers requested by their employers. The information in this data set consisted of data collected from all the fifty states in USA. Each visa petition in the data set contains the following fields:

CASE_STATUS: This is the status associated with the latest noteworthy event or decision concerning the petition. Valid values of this field include "Certified", "Denied", "Certified-Withdrawn" and "Withdrawn". For the purpose of binary classification in this project, we consider Certified and Denied as the valid values.

EMPLOYER_NAME: This is the name of the employer who files for the labor condition application.

JOB_TITLE: Title of the job role.

SOC_CODE: The Standard Occupational Classification (SOC) System classifies job roles, and this is the occupational code associated with the role requested in the petition.

FULL_TIME_POSITION: This field states whether the position being petitioned for is a full time one or not, “Y” and “N” are the valid entries.

PREVAILING_WAGE: This is the amount of prevailing wage for the position that is petitioned for in this labor condition application.

YEAR: The fiscal year in which this labor condition application was filed.

WORKSITE: This is a compiled field and it contains the city and state in U.S. where the job is based.

Lat: This field denotes the latitude coordinates of the worksite.

Lon: This field denotes the longitude coordinates of the worksite.

The data set has approximately 3 million records dating from the years 2011 through 2016.

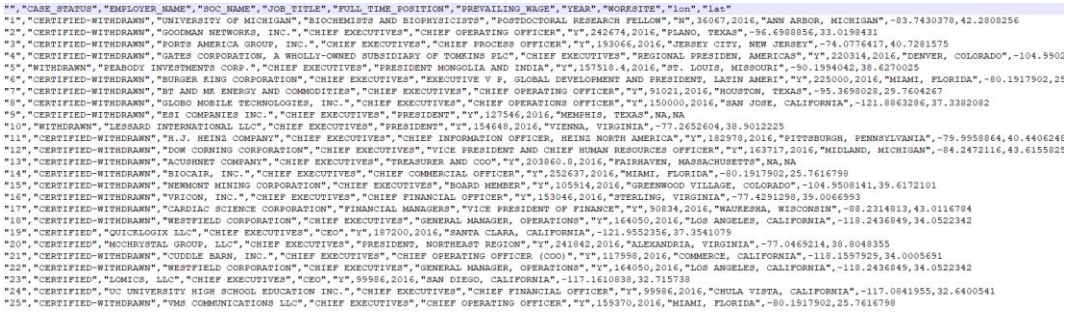

Fig 3.2: Snapshot of the data set

3.3 Data Preprocessing

It is highly probable that a data set in its raw form can include missing values, inconsistent entries and noise. These data entries affect the quality of the results. Thus, to improve the performance efficiency of the machine learning models, it is important to train and test the system on clean and consistent data. Redundant and unnecessary fields and values were also removed.

Some of the steps followed in this project for this purpose are:

Data Cleaning

Some petition rows in the data set contain some missing values, to handle such situations, the missing fields have either been filled with dummy values or the entire row has been discarded, depending on the best suited task. Some features such as ID are not necessary for any computations, so they have been completely ignored. Some computing tasks required features to be preprocessed by binary thresholding.

Feature Selection

The machine learning models were trained on the most important features of the data set, such as SOC_NAME, PREVAILING_WAGE and WORKSITE to see if selecting this subset of features improves the performance metrics of the learning models. The models were trained on various subsets of features before the above-mentioned subset was deemed to be the better one from the lot. Some functionalities of the project required only a subset of the features, for such cases, the rest of the features were ignored.

Chapter 4 - Experiments

An in-depth explanation of the experiments conducted on the data set is given in this section.

4.1 Data Preparation

For this project, supplemental data was downloaded from the U.S Bureau of Labor Statistics. It contains the state-wise median salary for each household. The percentile value of the "Prevailing Wage" column and the state-wise median salary was computed and this was used for further experiments. The attribute "Case Status" formed the target feature of this project. This attribute contains four valid entries – Certified, Denied, Certified-Withdrawn and Withdrawn. Further research of the application indicated that the values Certified-Withdrawn and Withdrawn were not beneficial for the analysis or the classification part of the project, hence the tuples with these values were removed from further processing. This converts the problem into a binary classification problem; the valid values of Case Status then became Certified and Denied. Apart from this, for data preparation data cleaning was performed to ensure the quality of further predictions.

Then the cleaned and compiled data set was split into training and testing data in a 70%-30% basis. The classifier was built upon the training features and its performance was measured by the test data.

- Total number of records: 3,002,458
- Training records: 2,010,721 (70%)
- Testing records: 991,737 (30%)

The main goal of this project was to predict the “CASE STATUS” of a visa petition. All the other attributes of the data set form the array of features for training. The accuracy of the classification model was tested upon the “CASE STATUS” value.

4.2 Design of Project

Various machine learning algorithms discussed in Chapter 2 are used as the basis for the experiments conducted in this project. They are described in detail below.

Python offers a package called Scikit-learn which has been used to implement all the inductive learning models. The evaluation criteria for the trained classifiers were: test set. These can help determine the accuracy of the classification task. These results were later manually compared to estimate which was the best classifier for this test bed.

4.2.1 Logistic Regression

This algorithm was implemented by importing the library Logistic Regression from Scikit-learn in this way: *from sklearn.linear_model import LogisticRegression*. The classifier was then fit on the training features and labels. The function predict_proba was used to estimate the probability. The function predict was used to make the actual predictions for class labels.

4.2.2 Random Forests

The algorithm Random Forests was implemented by importing the library in Scikit-learn as follows: *from sklearn.ensemble import RandomForestClassifier*. The depth, estimators and random state fields were specified before fitting the model and predicting the scores.

4.2.3 Gaussian Naïve Bayes

Similar methods were used to import the Gauissian Naïve Bayes library from Scikit-learn, using: *from sklearn.naive_bayes import GaussianNB*. The function fit was used to fit the learning model on the data and the function score was used to find out the F-score of this algorithm and to assess its performance.

4.2.4 K-Nearest Neighbors

For implementing this algorithm, *from sklearn.neighbors import KNeighborsClassifier* is used. Methods stated earlier were used to assess the performance of this model as well.

Chapter 5 - Results

This chapter includes the performance evaluation metrics and the results obtained after conducting the experiments mentioned in section 4.2.1, 4.2.2, 4.2.3 and 4.2.4.

5.1 Evaluation Metrics

The following methods were used to evaluate the performance of the machine learning models.

5.1.1 Precision, Recall, and F-Score

Precision can be defined as the number of true positive values in an evaluation divided by the total number of true positives and false positives.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Fig 5.1 Precision formula

Recall is the number of true positives divided by the sum of true positive and false negative values.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Fig 5.2 Recall Formula

F1-Score can be stated as the harmonic mean of Precision and Recall. Mathematically, it is given by the following formula.

$$F1 = 2 \times \frac{precision\ .\ recall}{precision + recall}$$

Fig 5.3 F1 Score Formula

5.1.2 Receiver Operating Characteristic Curve (ROC)

An ROC curve is one of the most widely used methods to analyze the performance of classifiers. It is a graph drawn between the true positive rate or recall and the false positive rate or fall out. Area under the curve (AUC) is often used to measure the quality of ROC Curves. And AUC is a totalized value of the area under the ROC curve. The maximum value of AUC is 1 and the higher the value is the better the performance of the classifier is.

5.2 Experimental Results

The results obtained for the algorithms discussed in Chapter 4 are explained here using their performance metrics.

Feature selection was performed using Scikit-learn's Recursive Feature Elimination (RFE) and Select K Best methods. Recursive Feature Elimination with Cross Validation (RFECV) was also performed. A negligible to no-change was noticed after their usage. Manual feature selection of the most probable features proved to be more advantageous for classification of this test bed.

K-fold cross validation was performed with the value of k=4. This seemed to have a good influence over the results.

5.2.1 Logistic Regression

The F1 score and Accuracy scores obtained when this model was run is given in the table below:

Performance Metric	Value
Prediction Accuracy	0.9672
F1 Score	0.8833

Table 5.1 Logistic Regression Performance Metrics

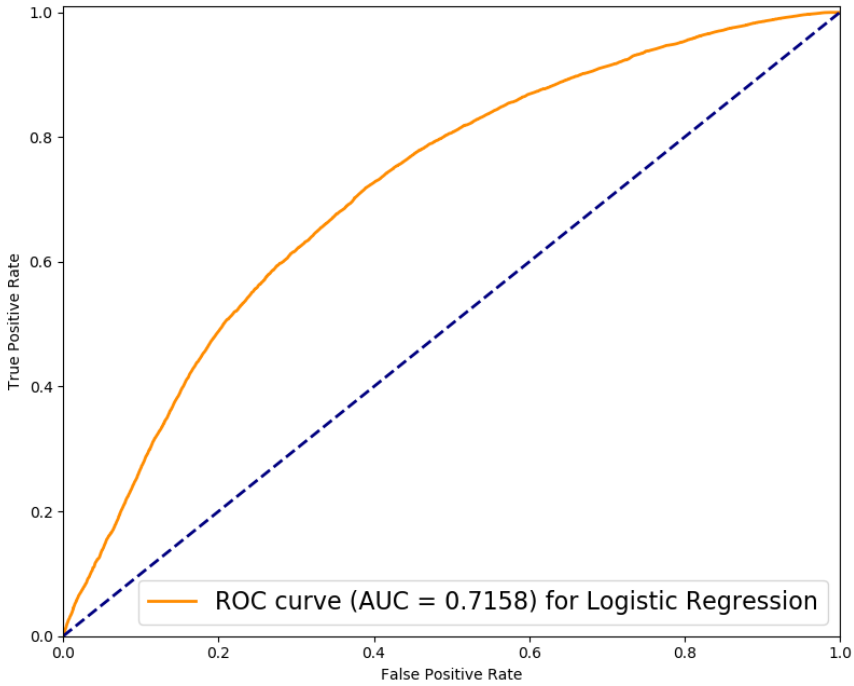

Fig 5.4 Receiver Operating Characteristic Curve for Logistic Regression

5.2.2 Random Forests

Bagged Random Forests, a type of Random Forests was used to perform the experiments in the project. Usage of this version of Random Forests helped in avoiding over-fitting of the data completely. The parameters for this algorithm were: max_depth = 5 and estimators = 40. After a series of experiments, these values were chosen as they gave the best prediction accuracy. The F1 score and Accuracy scores obtained when this model was run are given in the table below:

Performance Metric	Value
Prediction Accuracy	0.9677
F1 Score	0.9082

Table 5.2 Random Forests Performance Metrics

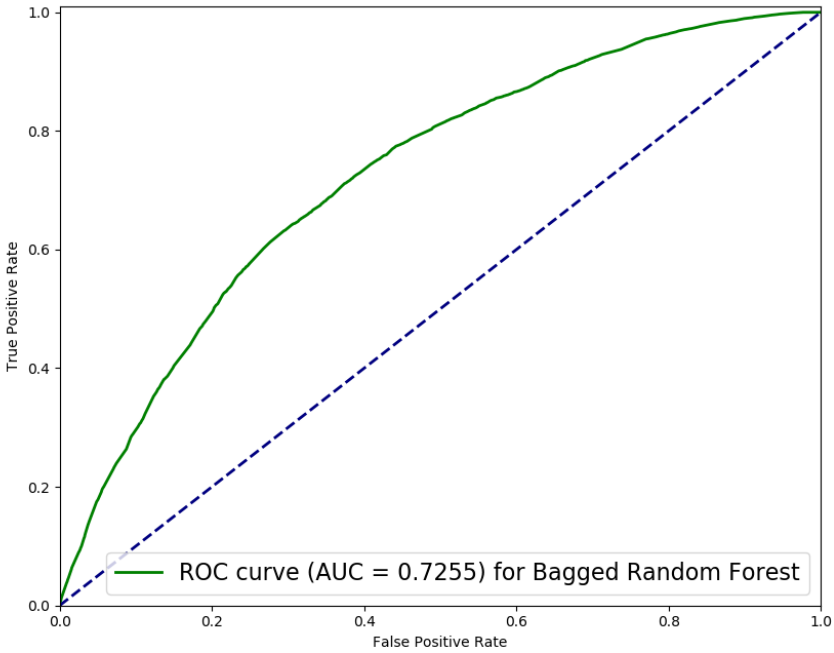

Fig 5.5 Receiver Operating Characteristic Curve for Random Forests

5.2.3 K – Nearest Neighbors

The F1 score and Accuracy scores obtained when this model was run are given in the table below:

Performance Metric	Value
Prediction Accuracy	0.9248
F1 Score	0.7637

Table 5.3 K-Nearest Neighbors Performance Metrics

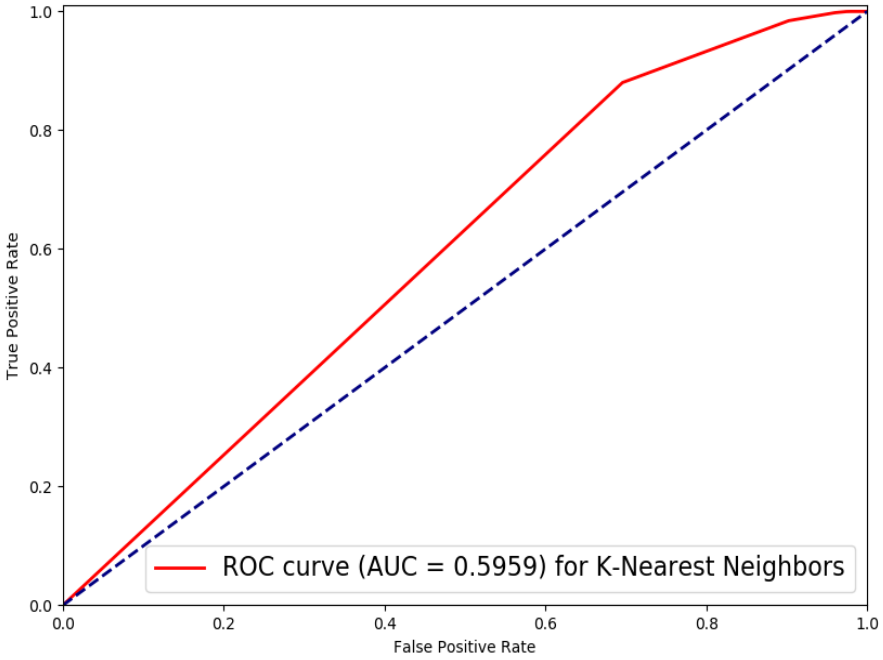

Fig 5.6 Receiver Operating Characteristic Curve for K-Nearest Neighbors

5.2.4 Naïve Bayes

I have used the Gaussian Naïve Bayes classifier for this project. The F1 score and Accuracy scores obtained when this model was run are given in the table below:

Performance Metric	Value
Prediction Accuracy	0.9324
F1 Score	0.8028

Table 5.4 Gaussian Naïve Bayes Performance Metrics

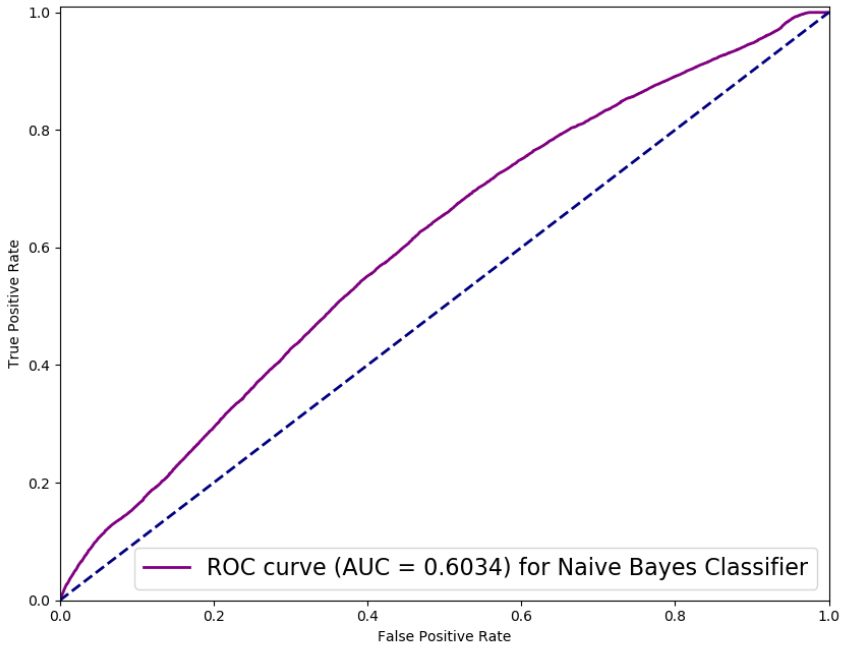

Fig 5.7 Receiver Operating Characteristic Curve for Gaussian Naïve Bayes

5.2.5 Comparison of all the classifiers

Classification and Prediction Times

The performance times for all the models were compared and it was observed that K-Nearest Neighbors took unusually long to train and predict compared to the others. Listed below are the classifiers and the time taken to train and predict over this data set.

Classifier	Performance Times (in seconds)
Logistic Regression	3.653
Random Forests	14.506
K-Nearest Neighbors	156.366
Gaussian Naïve Bayes	12.269

Table 5.5 Performance Times for all the Classifiers

F1 Scores

F1 Scores can be defined as a mean of precision and recall in simple terms. And all the classifiers and their F1 scores are listed below.

Classifier	F1 Score
Logistic Regression	0.8833
Random Forests	0.9082
K-Nearest Neighbors	0.7637
Gaussian Naïve Bayes	0.8028

Table 5.6 F1 Score for all the Classifiers

Receiver Operation Characteristic (ROC) Curve

A graph illustrating all the classifiers' performance was compiled.

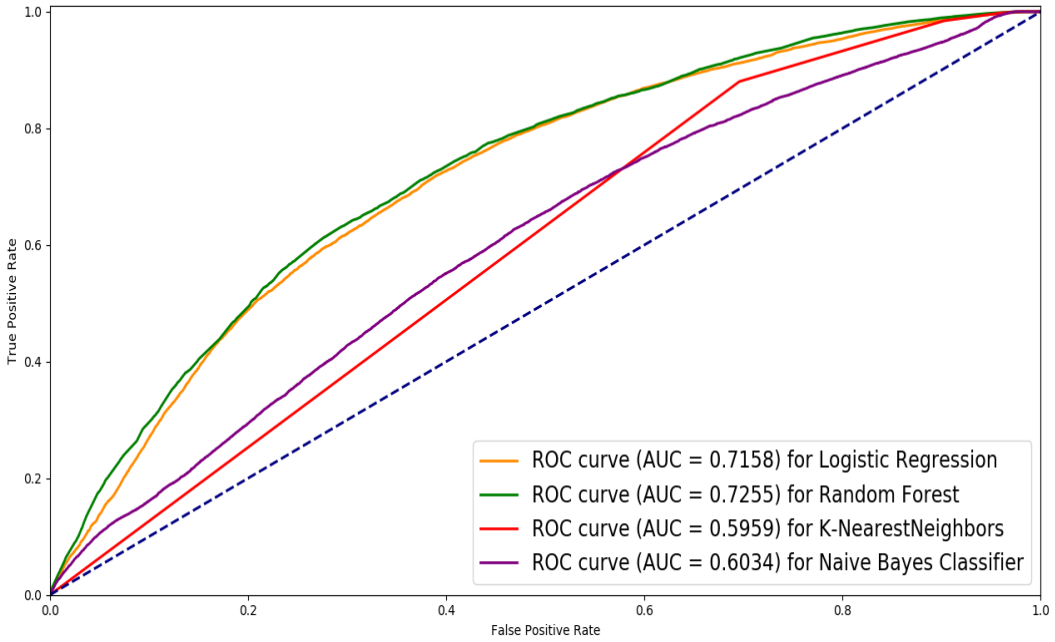

Fig 5.8 Receiver Operating Characteristic Curve for all Classifiers

From the graph it can be deduced that Random Forests and Logistic Regression are the better performing algorithms for this test bed, with Random Forests exceeding by a minute percentage. K-Nearest Neighbors and Naïve Bayes classifiers show a lower rate of performance.

The Area Under Curve values obtained are as follows:

Classifier	Area Under Curve
Logistic Regression	0.7158
Random Forests	0.7255
K-Nearest Neighbors	0.5959
Gaussian Naïve Bayes	0.6034

Table 5.7 Area Under Curve (AUC) for all the Classifiers

Chapter 6 – Summary and Future Scope

6.1 Summary and Interpretation of Results

Among all the classifiers that were implemented, it was observed that Bagged Random Forests out-performed K-Nearest Neighbors and Gaussian Naïve Bayes by a substantial margin and Logistic Regression by an extremely small amount. Though the performance time for Bagged Random Forests comes second to Logistic Regression, its AUC value and F1 Score is better than all the other algorithms experimented with, for this data set. The high prediction accuracy of 96.77%, F1 score of 0.90 and AUC value of 0.725 combined with the scalability nature of Random Forests make them a good classification learning algorithm for medium to large data sets such as this test bed. Logistic Regression is also a good alternative option as it has similar performance metrics – prediction accuracy: 96.72%, F1 score: 0.88, AUC value: 0.71 and a better performance time compared to Bagged Random Forests.

6.2 Future Scope of the Project

Supplemental data concerning the Standard Occupational Classification (SOC) can be gathered and used in coordination with this data set to obtain a more comprehensive analysis of how the H-1B Visa selection process works. By using the wage evaluations and ranges under SOC, the wage attribute in this data set can be correctly put in to a range of salaries which can then be used to classify the visa petitions based on occupation roles rather than location wise. In addition, other classification algorithms other than the discriminative models can be experimented with this testbed and their performances can also be analyzed.

Bibliography

A. Y. Ng, M. I. Jordan. (2001). On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes. *NIPS'01 Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, (pp. 841-848).

American Immigration Council. (2018). *The H-1B Visa Program: A Primer on the Program and its Impact on Job Wages, and the Economy.* Retrieved from American Immigration Council: https://www.americanimmigrationcouncil.org/sites/default/files/research/the_h-1b_visa_program_a_primer_on_the_program_and_its_impact_on_jobs_wages_and_the_economy.pdf

Bureau of Labor Statistics. (2018). *Standard Occupational Classification*. Retrieved April 01, 2018, from Bureau of Labor Statistics, U.S. Department of Labor: https://www.bls.gov/soc/

F. Harrell. (2017, January 15). *Classification vs. Prediction*. Retrieved April 01, 2018, from Statistical Thinking: http://www.fharrell.com/post/classification/

F. Pedregosa, et. al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research 12*, 2825-2830.

K. P. Murphy. (2002). *Machine Learning: A Probabilistic Perspective.* The MIT Press.

L. E. Peterson. (2009, Feburary 21). *K-nearest neighbor*. Retrieved April 01, 2018, from Scholarpedia: http://www.scholarpedia.org/article/K-nearest_neighbor

N. G. Ruiz, J. H. Wilson, S. Choudury. (2012, July). *The Search for Skills: Demand for H-1B Immigrant Workers in U.S. Metropolitan Areas, Metropolitan Policy Program.* Retrieved April 01, 2018, from Brookings: https://www.brookings.edu/wp-content/uploads/2016/06/18-h1b-visas-labor-immigration.pdf

N. J. Nilson. (2005). *Intoduction to Machine Learning.* The MIT Press.

R. Joshi. (2016, September 09). *Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures*. Retrieved April 01, 2018, from Exsilio Solutions: http://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/

R. P. Bunker, F. Thabtah. (2017, September 19). A machine learning framework for sport result prediction. *Applied Computing and Informatics*. Retrieved from https://doi.org/10.1016/j.aci.2017.09.005

S. Naribole. (2017). *H-1B Visa Petitions 2011-2016*. Retrieved January 29, 2018, from Kaggle: https://www.kaggle.com/nsharan/h-1b-visa

T. Fawcett. (2006). An introduction to ROC analysis. *Pattern Recognition Letters, Volume 27, Issue 8*, 861-874.

T. M. Mitchell. (1997). *Machine Learning.* McGraw-Hill, Inc.

Wikipedia. (2018). *Discriminative model*. Retrieved April 01, 2018, from Wikipedia: https://en.wikipedia.org/wiki/Discriminative_model