

Vishwakarma Institute of Technology, Pune

(An Autonomous Institute affiliated to Savitribai Phule Pune University)



A

Project Report

On

(Term-8th Sem)

“APPLICATION OF RECOMMENDER SYSTEM TO H1B VISA”.

Presented By:

- | | | |
|----|-----------------------|---------|
| 1. | ANAY DOMBE | :151666 |
| 2. | ASHITOSH ASHTURE | :151699 |
| 3. | KUSHANKUR CHAKROBORTY | :141378 |
| 4. | SACHIN KHUTAN | :151045 |

Academic Year: 2018-2019

Under Guidance of

Prof. Mr. Debabrata Swain

DEPARTMENT OF IT ENGINEERING

Vishwakarma Institute of Technology, Pune-411 037

BANSILAL RAMNATH AGARWAL CHARITABLE TRUST'S
VISHWAKARMA INSTITUTE OF TECHNOLOGY

(An Autonomous Institute affiliated to Savitribai Phule Pune University)

PUNE – 411037



C E R T I F I C A T E

This is to certify that the Dissertation title **Application of Recommender System to H1B Visa** submitted by **Anay Dombe (151666), Ashitosh Ashture (151699), Kushankur Chakroborty (141378), and Sachin Khutan (151045)** is in fulfillment for the award of Degree of Bachelor of Technology in Information Technology Engineering of Vishwakarma Institute of Technology, Savitribai Phule Pune University. This Dissertation is a record of bonafide work carried out under my guidance during the academic year 2018-19.

Guide

Prof. Debabrata Swain

Dept. of IT Engineering
Vishwakarma Institute of Technology

HOD

Prof. Dr. Premanand Ghadekar

Dept. of IT Engineering.
Vishwakarma Institute of Technology

External Examiner

Date: 18/05/2019

BANSILAL RAMNATH AGARWAL CHARITABLE TRUST'S
VISHWAKARMA INSTITUTE OF TECHNOLOGY

(An Autonomous Institute affiliated to Savitribai Phule Pune University)

PUNE – 411037

Project Synopsis

Group No :G-33

Group Members:

Roll No	Name	Class	Contact No	Email-Id
16	Anay Dombé	H	8975471732	anay.dombel5@vit.edu
8	Ashitosh Ashture	H	9730123525	ashitosh.ashture15@vit.edu
41	Kushankur Chakroborty	H	8879503311	kushankur.chakraborty14@vit.edu
49	Sachin Khutan	H	8796327175	sachin.khutan15@vit.edu

Academic Year : 2018-19

Project Title : Application of Recommender System to H1B Visa

Project Area : Machine Learning, Recommendation Systems

Internal Guide : Prof. Debabrata Swain

External Examiner

Internal Examiner

ABSTRACT

The H1B is a visa that allows US employers to employ foreign workers in specialty occupations. The number of H1B visa applicants is growing drastically. Due to a heavy increment in the number of applications, the lottery system has been introduced, since only a certain number of visas can be issued as per the category every year. But, before an application enters the lottery pool, it has to be approved by the Labor Committee Application (LCA). The approval or denial of this visa depends on a number of factors such as salary, work location, full time employment, etc. The purpose of this research is to predict the outcome of an applicant's H1B visa application by using deep learning and machine learning techniques and generate recommendations if the prediction is denied so that the applicant can work on the recommended areas to strengthen his application and increase his chances of approval.

Keywords: Machine Learning, Artificial Neural Networks, Deep Learning, Associations Rule Mining

INDEX

Chapter No.	Title	Page No.
	LIST OF FIGURES	6
1	INTRODUCTION	7
	1.1 Project Plan	8
	1.2 Background	9
	1.3 Types of Visas	10
2	LITERATURE REVIEW	12
3	H1B VISA RECOMMENDER SYSTEM	13
	3.1 Objective	13
	3.2 Problem Definition	13
	3.3 Requirement Analysis	14
	3.4 Implementation	16
	3.4.1 Flow Chart	17
	3.4.2 Data Analysis	17
	3.4.3 Data Pre-processing	26
	3.4.4 Algorithms	29
4	EXPERIMENTS	34
	4.1 Training and Testing	34
	4.2 Results	34
	4.3 Recommendations	35
5	FUTURE SCOPE	37
6	REFERENCES	38

LIST OF FIGURES

Fig. No.	Title	Page No.
1	Overview of recommendation system	17
2	Analysis of the H-1B by the status of their visa applications	18
3	K-Means Clustering implementation to see the location and density of the applications.	19
4	K-Means Clustering to analyze top 10 cities	20
5	Median Salary of top 10 cities	21
6	Top 6 company with most application number	22
7	Analysis of job title	23
8	Analysis of salary	24
9	Comparison of full time versus part time	25
10	Total Application numbers	25
11	One hot encoding	28
12	Neural Network Architecture	29
13	Logistic Regression	32
14	Accuracy obtained by Neural Networks	35
15	Accuracy obtained by Random Forest Classifier	36
16	Accuracy obtained by Gaussian NB	37
17	Recommendations using cosine similarity	38
18	Recommendations using ARM	39

CHAPTER 1

INTRODUCTION

The H1B visa is a work visa granted by the United States department of Immigration under the immigration act of the United States constitution for highly skilled foreign workers who want to enter the US on a valid work visa issued through a selective process. The visa is validated under strict stipulations. The applications are normally made by MNC's on behalf of their employees to the US embassies in their respective countries. After considerable screening processes the applicants receive their H1B visas. The requirements set forth by the government are: 1) A Degree of a bachelor's or master's course (or the foreign equivalent degree from your Country), OR 2) Work experience in a relevant field for at least 12 years, OR 3) Higher education + work experience in relevant field. The general qualification rule is 3 points per year of University - for every 1 year of work experience = 1 point. 12 points in 'total' are required to qualify for the H1B visa program. With only around 75000 IT engineers securing the H1B visa in a year and applications numbering over a million, the selection process is extremely competitive. And with the current changes in the visa policy by the Administration the chances of getting a visa for an average candidate have become slim. Current immigration law allows the US department of Immigration policies to approve a total of 85,000 new H-1B applications per fiscal year. Among these, 65000 are issued for highly skilled foreign workers working with a professional MNC organization (Mostly IT) and remaining 20,000 are reserved for master's degree foreign students who have studied in a US accredited university and who have been offered employment by a US company. Increasingly the H1B visa cap has been overly applied for. Therefore, the lottery system is also used to grant the visas. The idea behind our project is to predict the chances of a visa applicant after analyzing parameters such as his/her salary, job profile, company profile, education, ethnicity, gender, country of origin, degree, duplicacy, background check etc. The use of Data Analytics is fundamental for this project. We have used Anaconda and Spyder for implementation. Through Kaggle, we found the data of over 3 million previous H1B visa applications from around the world. By incorporating Machine Learning Algorithm, we are predicting a probability of visa approval chances. Increasingly the H1B visa cap has been overly applied for [1]. Therefore, the lottery system is also used to grant the visas. However, before an application enters the lottery process, the LCA either approves or denies the visa. If the visa is approved, then it enters the lottery pool. In this paper, we have proposed a model that predicts whether the H1B visa will be approved or denied by taking inputs from the user and provides suggestions to the user to maximize his/her approval chances.

1.1 PROJECT PLAN

Work Activity	Jan	Feb	Mar	Apr	May
Problem identification.					
Literature review.					
Data collection.					
Identification of area of Improvement					
Scope for improvement.					
Preparation of action plan.					
Collection of data after implementation of action plan					
Implementation					
Report writing and submission					

1.2 BACKGROUND

Since the beginning it was decided that the domain of our project would be in data analytics and machine learning. With these fields currently in high demand we were eager to work and study on them. Every IT engineer's goal is to secure an H1B visa and travel to US and work on good projects. Upon a casual discussion on the same with respect to current visa issues, an idea struck what if we develop software which could tell the user his/her chances of getting an H1B visa after analyzing parameters such as salary, job title, degree, company profile etc. Further reading on the same, it was found that we would have to extensively use Machine Learning algorithm like Logistic Regression. While studying about them we concluded that all the coding will have to be done in Python. Python is a programming language with high level interpreted features intended for general development use. It is built with the philosophy that puts an emphasis on readability via white spaces. Clustering is a method of quantization, originally from signal processing, that is popular for cluster analysis in data mining. K-means cluster in gains to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. Machine learning (ML) is a category of algorithm that allows software applications to become more accurate in predicting outcomes without being explicitly programmed. The basic premise of machine learning is to build algorithms that can receive input data and use statistical analysis to predict an output while updating outputs as new data becomes available.

1.3 Types of Visas

The most common types of US Visa from India are as follows:

Tourist or business Visa

There are two types of Visas offered under this tourist or business Visa. They are:

B-1 for business associates, those attending scientific, educational, business conventions, settling an estate or to negotiate contracts.

B-2 for pleasure or for medical treatment. This includes tourism, visiting friends and family, medical treatment, social or service activities.

Most often the two types are combined and issued as one Visa. You have to prove to the consular officer that the purpose for you to travel to the U.S. is for a limited period and that it is temporary and you have to show the evidence of funds to cover your expenses while you are in the U.S. You must also show that you have residence outside the U.S. that you have to return to. While you are on a visitor Visa you are not permitted to accept employment in the U.S.

Work Visa

In order to work in the U.S. on a temporary basis, you need a specific Visa based on the type of work that you will be doing. Applicants of H, L, O, P and Q will have to get their petition approved on their behalf by USCIS. Form I-129 must be approved before applying for work Visa at the Consulate. After which the employer will receive Form I-797 that serves as your petition's approval notification. While giving your interview at the Consulate, you will have to bring I-129 and copy of Form I-797.

Visas offered for work are as follows:

H1-B for Specialty occupation

To qualify for the H1-B Visa you must hold a Bachelor's degree in specific specialty and USCIS will determine if your employment constitutes a specialty occupation and if you are qualified to perform the service. Employer is required to file labour condition application with the Department of Labor regarding the terms and condition of the contract of employment with you.

H-2A for Seasonal agricultural workers

This Visa is allows U.S. employers to bring foreign nationals to fill the temporary agricultural job for which U.S. workers are not available. Employer must file Form I-129 petition on your behalf. Indians are not eligible for this Visa.

- **H2-B for skilled and unskilled workers**

This Visa is granted to those filling up a temporary or a seasonal job for which there is a shortage of U.S. workers. Indians are not eligible for this Visa.

- **H-3 for trainees**

This is required if you are coming to the U.S. to receive training in any field from an employer for a period of up to 2 years. You can be paid for the training but it cannot be used to provide productive employment.

CHAPTER 2

LITRETURE REVIEW

A large number of changes are taking places while recruiting new employees due to the restrictions imposed by H1B visa. For instance, Paper [4] has presented the changes in recruiting tracts because of the changes in H1B policies. H1B visa has played a key role in hiring international talent. Paper [5] has showed how the United States has managed to improve its post-secondary education by hiring top faculty from overseas.

The approval of H1B visa is decided on several parameters. In the current scenario, the salary parameter is extremely important due to some of the changes, which have been proposed by the administration and as a result, it is crucial to understand the salary trends of the H1B workers [3]. Interesting salary versus case status trends have been demonstrated by [6] which show how vital the salary parameter is for the applicant. It is evident how salary changes from one state to another from the salary versus job location trends [6]. Similarly, other interesting trends have been graphically presented [6]. These trends can be useful not just from the application point of view but also prior to finding a job. For example, a trend shows that people working in California are earning quite higher than any other states [6]. This trend can be helpful for people who are looking for the highest paid job.

Paper [1] presented an idea to predict the outcome of H1B visa using machine-learning algorithms. However, they have only used Logistic Regression and Random Forest Classifier for prediction. In this paper, we have compared various machine learning and deep learning algorithms and we have selected the best one based on certain parameters for prediction. Paper [2] also states some machine learning approaches that can be used for prediction. However just predicting the case status of any application will not guide the applicant in a proper direction. Thus, it is necessary to suggest the areas; the applicant needs to focus on so that the applicant has a higher chance of H1B approval.

CHAPTER 3

H1B VISA RECOMMENDER SYSTEM

3.1 OBJECTIVE

Objectives of our model are listed as follows:

- To pre-process the data and balance the dataset.
- To accurately determine the prediction of H1B visa.
- To generate recommendations if the prediction is denied so that the applicant can work on the recommended areas to strengthen his / her application.

3.2 PROBLEM DEFINITION

Given the attributes of an applicant as input, determine the likelihood of H1B visa approval and if our model shows prediction as DENIED, then recommend the areas (attributes) he/she needs to work on to increase the probability of approval.

Approach The model has been designed in three phases

Phase 1: **Data Pre-processing**

In this phase, data pre-processing has been done. One hot encoding of values has been done and we have removed the outliers. The dataset is also balanced.

Phase 2: **Prediction of the H1B visa application (Approved or denied)**

In this phase, we are predicting the case status using machine learning algorithms.

Phase 3: **Generating recommendations if the prediction is denied.**

In this phase, we are generating recommendations by finding the most similar user using cosine similarity and we are generating recommendations using one to one mapping.

3.3 REQUIREMENT ANALYSIS

3.3.1 Feasibility Study

The important Outcome of the requirement investigation is the determination that the System requested is feasible.

Preliminary investigations examine project feasibility; the likelihood the system will be useful to organization. Their details are as follows:

A Feasibility study is undertaken to determine the portability or possibility of either improving the existing system or developing a completely a new system.

It is conducted to test the technical feasibility of the system.

3.3.2 SOFTWARE AND HARDWARE REQUIREMENT

- **Software Requirement**

1. Python Environment
2. Scikit Learn
3. Tensorflow
4. Keras
5. Pandas
6. GPU Drivers

- **Hardware Requirement**

1. Laptop
2. Screen Size: Variable Screen size are handled by application.

- **Libraries**

- **NumPy** stands for Numerical Python. The most powerful feature of NumPy is n-dimensional array. This library also contains basic linear algebra functions, Fourier transforms, advanced random number capabilities and tools for integration with other low level languages like Fortran, C and C++
- **SciPy** stands for Scientific Python. SciPy is built on NumPy. It is one of the most useful library for variety of high level science and engineering modules like discrete Fourier transform, Linear Algebra, Optimization and Sparse matrices.
- **Matplotlib** for plotting vast variety of graphs, starting from histograms to line plots to heat plots.. You can use Pylab feature in ipython notebook (ipython notebook --pylab = inline) to use these plotting features inline. If you ignore the inline option, then pylab converts ipython environment to an environment, very similar to Matlab. You can also use Latex commands to add math to your plot.
- **Pandas** for structured data operations and manipulations. It is extensively used for data munging and preparation. Pandas were added relatively recently to Python and have been instrumental in boosting Python's usage in data scientist community.
- **Scikit** Learn for machine learning. Built on NumPy, SciPy and matplotlib, this library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.
- **Keras** Keras is written in Python and can run on top of TensorFlow (as well as CNTK and Theano). The TensorFlow interface can be a bit challenging as it is a low-level library and new users might find it difficult to understand certain implementations. Keras, on the other hand, is a high-level API, developed with a focus to enable fast experimentation. So if want quick results, Keras will automatically take care of the core tasks and generate the output. Both Convolutional Neural Networks and Recurrent Neural Networks are supported by Keras. It runs seamlessly on CPUs as well as GPUs. A common complaint from deep learning beginners is that they are unable to properly understand complex models. If you're one such user, Keras is for you! It is designed to minimize user actions and makes it really easy to understand models.

We can broadly classify models in Keras into two categories:

- Sequential: The layers of the model are defined in a sequential manner. This means that when we're training our deep learning model, these layers are implemented sequentially.
- Keras functional API: This is generally used for defining complex models, such as multi-output models or models with shared layers. Keras has multiple architectures, mentioned below, for solving a wide variety of problems. This includes one of my all-time favorites – image classification!

1. VGG16
2. VGG19
3. InceptionV3
4. Mobilenet

There are numerous components that go into making TensorFlow.

The two standout ones are:

TensorBoard: Helps in effective data visualization using data flow graphs

The flexible architecture of TensorFlow enables us to deploy our deep learning models on one or more CPUs (as well as GPUs). Below are a few popular use cases of TensorFlow:

1. Text-based applications: Language detection, text summarization
2. Image recognition: Image captioning, face recognition, object detection
3. Sound recognition
4. Time series analysis
5. Video Analysis

3.4 IMPLEMENTATION AND METHODOLOGY

3.4.1 Flow Chart:

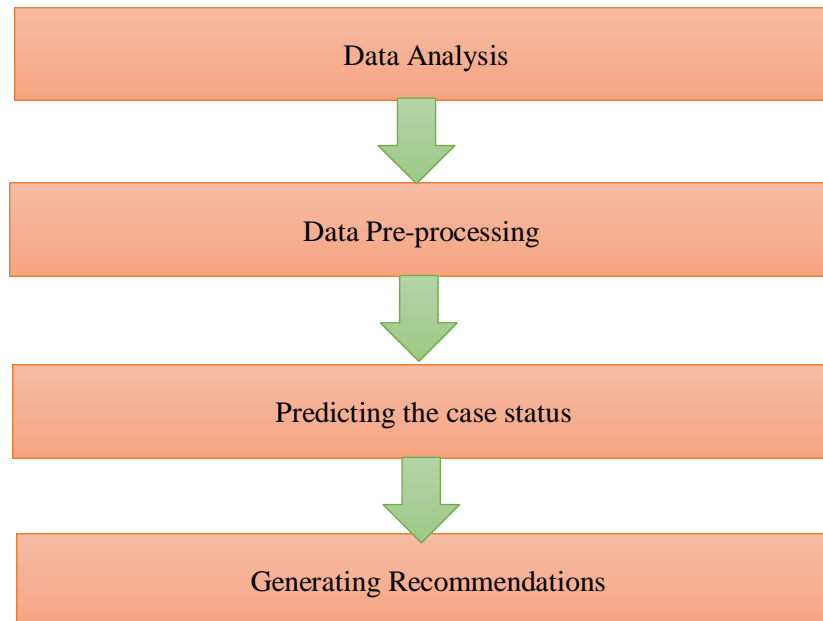


Fig 1 Overview of the Recommendation System

3.4.2 Data Analysis:

ANALYSIS OF THE H-1B BY THE STATUS OF THEIR VISA APPLICATIONS

The function `showCASE_STATUS(self, H1Infor)` of the source code shows the percentage of the status of all application, which is illustrated in the Figure 2. The panel shows that the majority (87%) of the visa application is certified, followed by 7% Certified-Withdrawn. It is noticed that there are 3% visa application is denied.

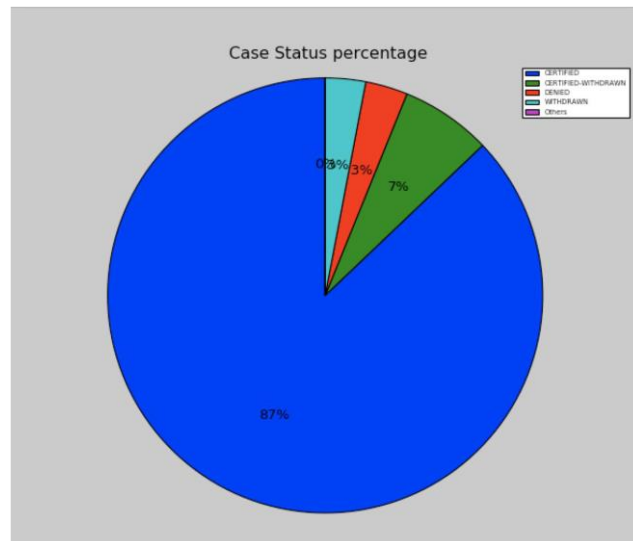


Figure 2. Analysis of the H-1B by the status of their visa applications.

The panel shows the percentage of the visa status, certified, certified-withdraw, denied, withdraw and others. The majority of the application is certified.

K-MEANS CLUSTERING TO SEE WHERE H-1BS ARE

The function `def K_meansAnlyze(self, H1Infor)` of the source code implements the K-Means algorithm. It classifies the application locations based on the longitude and latitude of the sites. The density of the applications were shown in Figure 3. It suggested that the majority of the applications were located in the California and Northeast states of US.

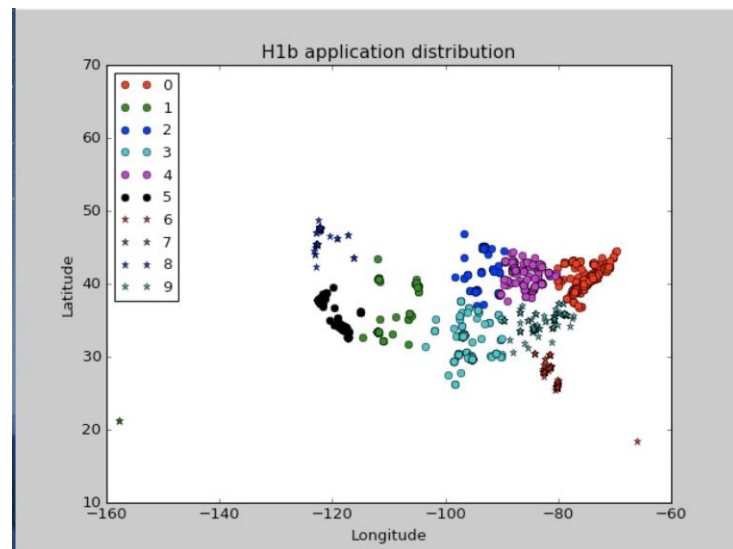


Figure 3. K-Means Clustering implementation to see the location and density of the applications.

K-MEANS CLUSTERING ALGORITHM TO ANALYZE THE TOP 10 CITIES THAT APPLY THE H-1B VISA.

The function `showWORKSITE(self, dense, H1LatLong)` of the source code utilized the K-Means clustering algorithm to get the 10 clustering, which suggested the top 10 cities that apply the H-1B visa for the employees (Figure 3).

As shown in Figure 3, the most H-1B were applied from New York (about 190,000 cases). It is noticed that there are three cities, San Francisco, San Jose, Sunnyvale in the state California were list in the top 10 cities of the application. Considering these two phenomena, the New York and California have the most applications, which is consistent with our previous finding that plotting the locations of the application in map (Figure 3).

In addition, the Houston, TX has the 2nd application, which about 90,000 cases. Another city, Dallas, also from Texas, was also listed in this location plotting. This suggested that the cities in Texas also has many applications in US.

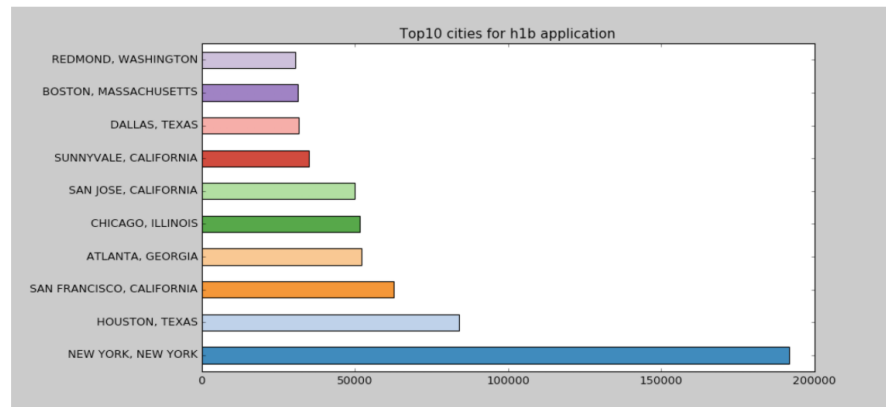


Figure 4. K-Means clustering algorithm to analyze the top 10 cities that have applied the H-1B visa.

The bar shows the number of the application based on the training data. The most cases of application was proposed in New York, followed by Houston in Texas. There are three cities in California and Two cities in Texas were listed here.

ANALYSIS OF AVERAGE SALARY FOR THESE TOP 10 CITIES

The function salaryAnalyze(self, H1Info, H1LatLong) implements the salary analysis of these 10 top cities, which is shown in Figure 5. The highest median salary of these 10 cities are Houston in Texas. The 2nd and most similar high salary is found in San Jose in CA. They show that about 80,000 USD is the median salary of all the applications. However, San Francisco in CA has the lowest salary in these top 10 cities, which about 55,000 USD.

Figure 5 implied that among these 10 cities, the cities in Texas, for example, Dallas and Houston, have the highest median salary. Three cities in California, have the 2nd top highest median salary. Considering the living expense compared with Texas and California, the savings of the people in Texas might be more than those of the people in California. However, because the living expense is not available in our dataset, it is better if we could analyze other data to draw a conclusion.

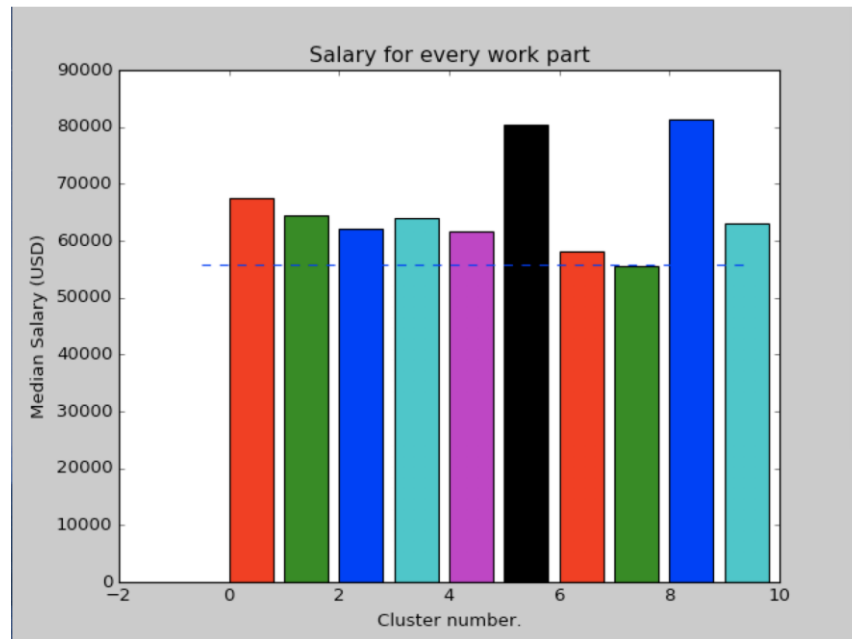


Figure 5. Median salary of these top 10 cities.

K-Means algorithm plotting of the Median salary of these top 10 cities. The greatest salary is from the 5th city, Houston.

THE TOP 6 COMPANY THAT HAVE THE MOST APPLICATION NUMBER AND THE APPLICATION TREND ANALYSIS

The function `showTOP6com_table(self, H1Info)` of the source code was implemented and it is used to analyze the top 6 companies that have the most applications. The H-1B application number of the top 6 companies was shown in Figure 6.

The trend of the application of these top 6 companies were also illustrated in Figure 6B. It is interesting to find that from 2011 to 2015, almost all companies have an increasing application, while in 2016, there is slightly decreased among these companies.

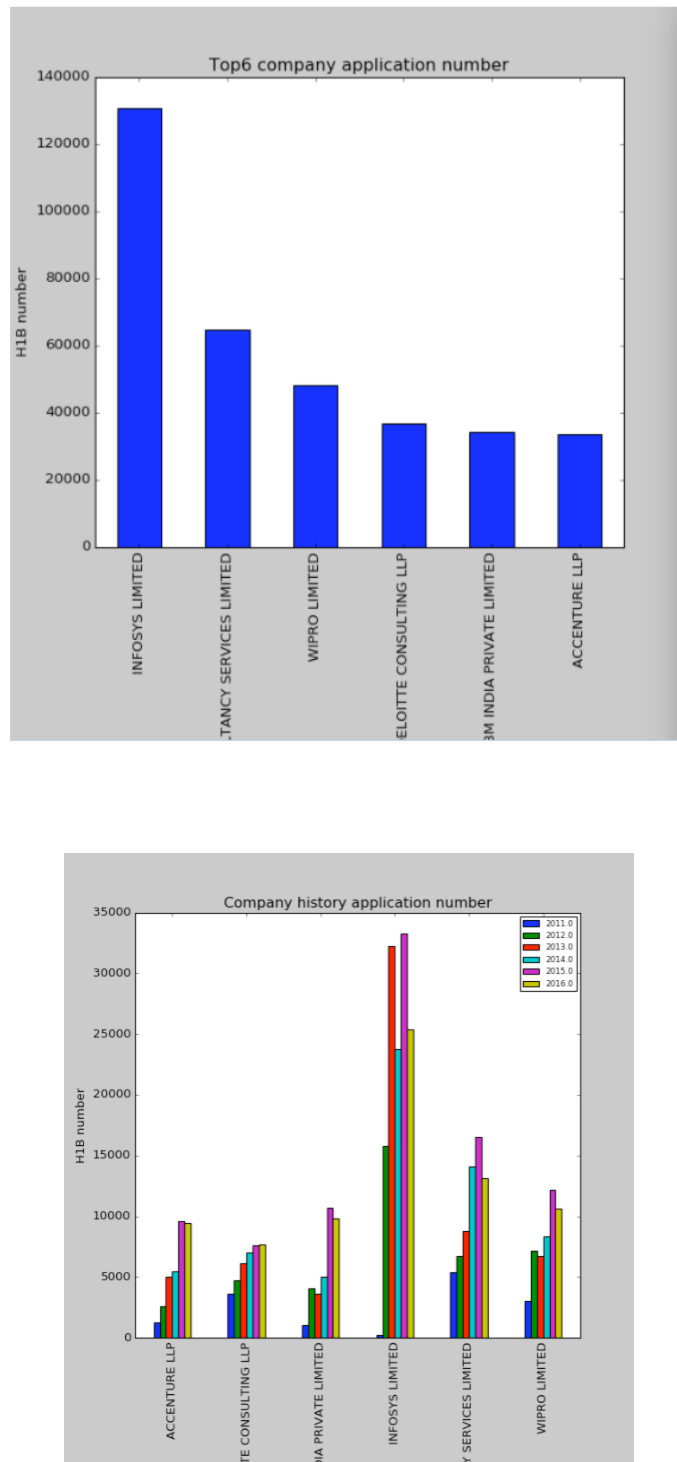


Figure 6. The top 6 company that have the most application number and the application trend analysis.

This figure shows the top 6 companies that have the greatest application numbers (A) and the trend of the application number in the recent years 2011 – 2016 (B). The trend shows that almost the application in 2015 has a obviously increased application and 2016 shows a slightly decrease of the application numbers.

ANALYSIS OF THE JOB TITLE FOR THE MOST APPLICATION

The function `showJOBTITLE_plot(self, H1Info)` of the source code analyze the top 20 jobs that get the most application proposed, which is shown in Figure 7.

The top job title was demonstrated to be the Programmer Analyst, with about 250,000 cases. The 2nd and 3rd top job titles were suggested to be Software Engineer and Computer Programmer. It is noticeable that there among the top 20 job titles, Computer Science-related job titles occupied most in this list. It might encourage more IT company and start-up company to focus on IT and technique-related area.

Other title jobs, such as Business analyst, consultant, physical therapist, accountant have also shown very high application numbers compared to other non-CS-related job titles. It could also provide some insight into the career choice for the students or the self-employed individuals.

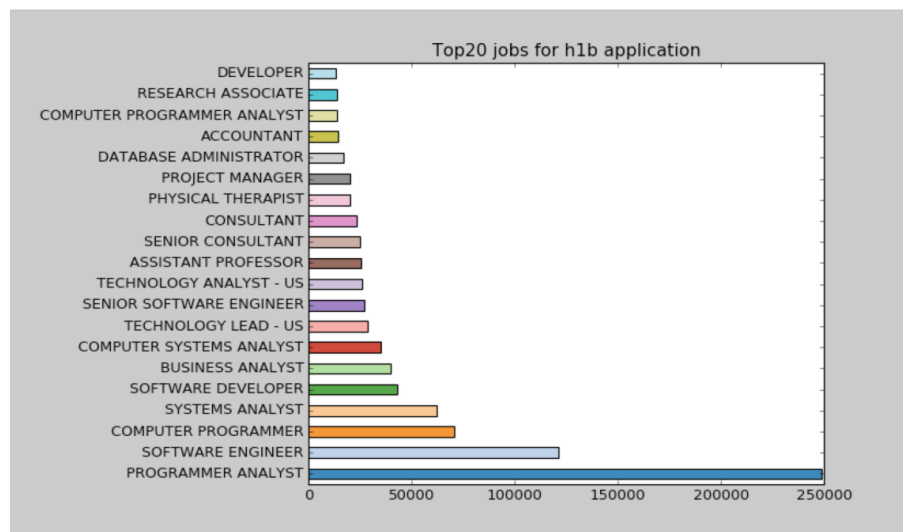


Figure 7. Analysis of the job title for the most application.

The top 20 job titles show that the majority of the jobs that have the application number is computer science-related jobs. The most number of the application of the top 1 job title, Programmer Analyst, is about 250,000.

ANALYSIS OF THE TOP SALARY OF AS PER DIFFERENT JOB TITLES

The function `showAVGSalary_plot(self, H1Info)` of the source code provides the plotting of the top 20 average salary of different job titles, which is shown in Figure 8. Some senior job, such as director, vice president, manager, has no doubt high salary among the applications. Others like quality test engineer, developer also have very high salary. However, compared with the data from Figure 6, which is shown the large amount of application is for CS-related job, it shows that although there are large demand for the CS-related occupation, many other jobs, such as assistant professor, consultant, hospitalist, health economics, *etc.* have quiet high salary compared with the CS-related occupation.

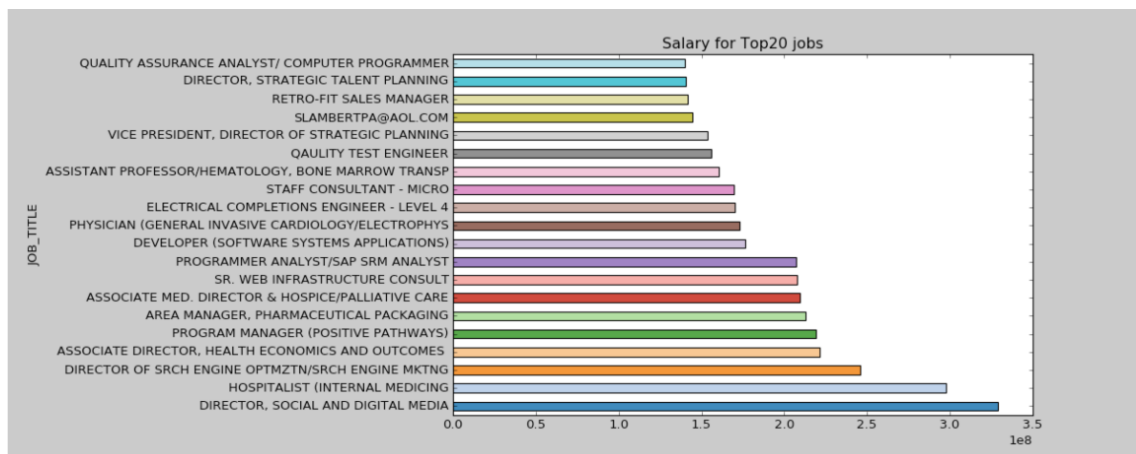


Figure 8. Analysis of the top salary based on different job titles.

Many senior level jobs earn highest salary among other occupation. Some CS-related job, such as developer, quality test engineer, also have high salary.

COMPARISON OF THE APPLICATION NUMBERS OF FULL-TIME V.S. PART-TIME JOBS

The function `showFullvsPart_plot(self, H1Info)` of the source code plots the H-1B application number comparison between the Full-time and Part-time jobs, which is shows in Figure 9. It is shown that the majority of the cases if full-time jobs.

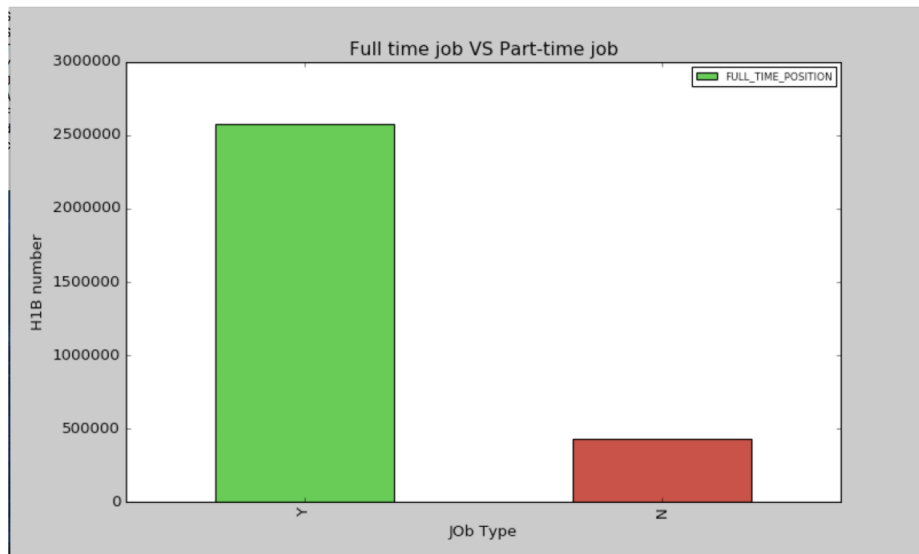


Figure 9. Comparison of the application numbers of Full-time V.S. Part-time jobs.

ANALYSIS OF THE TREND OF APPLICATION NUMBER FOR EACH YEAR

Figure10 shows the numbers of application for each year, by the function showYearTrend_plot. It provides an overall picture of the trend of application along with the year. It is shown that from 2011 to 2016, the application number is increasing.

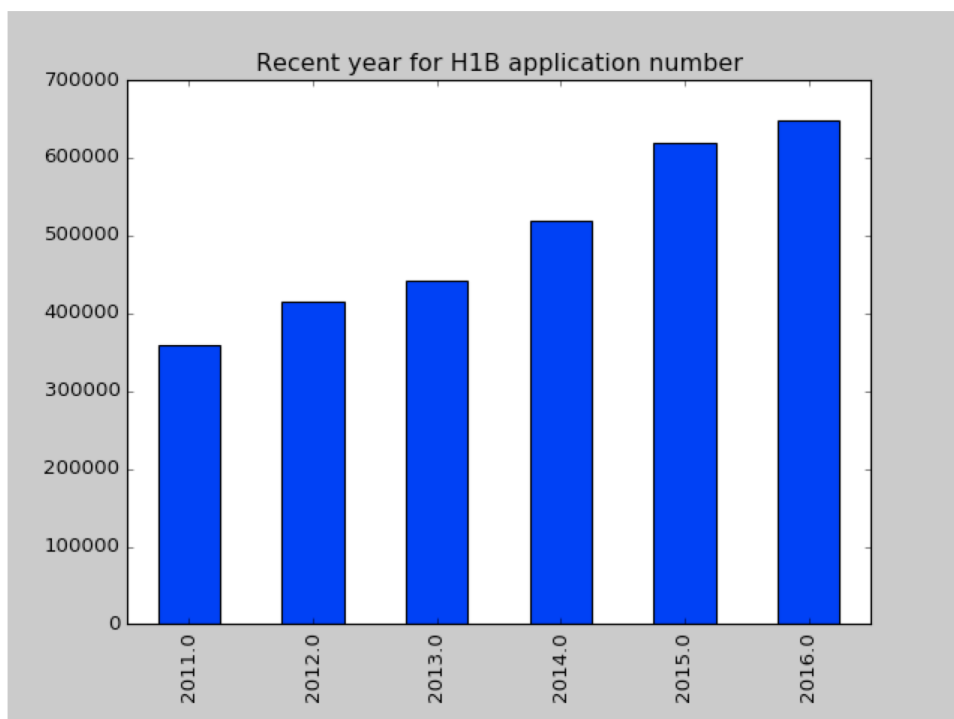


Figure 10. Total application numbers of the H-1B visa for each year.

3.4.3 Data Pre-processing:

The dataset was acquired through Kaggle [2]. The dataset has about 3 million entries. The name and description of each column has been given below:

1. EMPLOYER NAME: Name of the Employer
2. YEAR: H-1B visa petition year
3. SOC NAME: Name associated with the SOC CODE, which is an occupational code, associated as the job has been granted temporary labor condition, this is required for Standard Occupational Classification (SOC) System
4. JOB TITLE: Title of the job
5. PREVAILING WAGE: Temporary labor condition is requested for the job and to set Prevailing Wages. The wage is scaled and listed in USD. The definitions on Prevailing wages are as follows: the average wage paid to similarly employed workers in the requested occupation in intended employment. It is an indication of employer's minimum requirements at the company for the specified job title
6. FULL TIME POSITION: Y = Full Time Position; N = Part Time Position
7. WORKSITE: Information of the applicant's city and state intended for work in USA
8. CASE STATUS: It states if an applicant has been granted a visa or not. (Certified implies that a visa has been granted)
9. LAT/LON: It is the latitude and longitude of the worksite of the employer.

Chosen Attributes:

The dataset we have used had data for the year 2016 only. Therefore, we decided to remove the YEAR attribute. We are trying to predict the likelihood of approval based on the geographical area the individual works in. The exact location (longitude and latitude) may adversely affect the generality of the algorithm. Therefore, we decided to remove the same and keep WORKSITE only in the dataset. Therefore, the resulting dataset used for training and testing had the following attributes:

EMPLOYER_NAME

SOC_NAME

JOB_TITLE

FULL_TIME_POSITION

PREVAILING_WAGE

WORKSITE, CASE_STATUS

Removing Outliers:

The only numeric attribute in the dataset was PREVAILING WAGE. We used box-plot to detect the outliers and removed the same using $1.5 \times \text{IQR}$ rule.

Balancing Dataset:

The original dataset had four possible values for the attribute CASE_STATUS, which were CERTIFIED, DENIED, WITHDRAWN and CERTIFIED-WITHDRAWN.

We removed WITHDRAWN and CERTIFIED-WITHDRAWN rows from the dataset because it solely depends on the applicant. In addition, the original dataset was highly skewed. The number of certified rows was very less compared to that of denied. Therefore, we randomly selected rows from the dataset such that the resulting dataset had equal number of certified and denied rows to avoid biasness.

One-Hot Encoding:

Except PREVAILING WAGE, all other attributes were in textual format. To feed them to the neural networks, it was necessary to convert them into numeric format. FULL_TIME_POSITION column had only two possible values: Y and N. We replaced Y with 1 and N with 0. CASE_STATUS column also had only two possible values after removing withdrawn cases in the previous step. So we replaced CERTIFIED with 1 and DENIED with 0.

The remaining four columns: EMPLOYER_NAME, SOC_NAME, JOB_TITLE, WORKSITE were textual. To get the numeric representation of these columns we used one hot encoding technique from Keras library.

The pictorial representation of the one-hot encoding is shown in Fig.11.0 Left side of the fig. shows six columns. Four out of the six columns are then one-hot encoded separately. All the resultant one hot encoded columns are merged to produce the final dataset.

Normalization:

After one-hot encoding, all attributes, except prevailing wage, had values 0 or 1. Therefore, we normalized prevailing wage using minmax normalization to scale its value in the range [0, 1]

$$x - \min \{X\} \quad y = \quad \max \{X\} - \min \{X\}$$

x- An attribute value

X- An array of all possible values for attribute x

y- Normalized value of attribute x

The resulting dataset, after preprocessing, was used for training and testing.

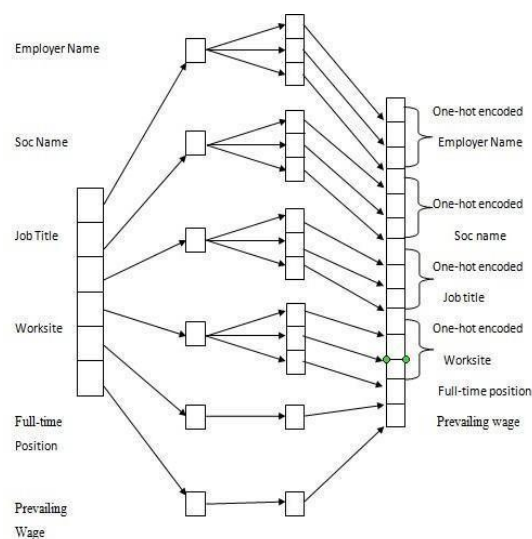


Fig. 11

3.4.4 Algorithms:

Artificial Neural Networks

Artificial Neural Networks, which is inspired by human brain, is a state of the art technique to find patterns in data and make predictions. We applied Multi-Layer Perceptrons (MLP) network to our problem to predict the case status of a visa application. The neural network architecture we used is shown in Fig. 12.0. The input to the neural network is one-hot encoded data generated after preprocessing. The input layer is of 16978. It has four hidden layers having neurons 512, 128, 64, 32 respectively. The activation function for the hidden layers is Rectified Liner Unit (ReLU). The output layer has a single neuron with sigmoid activation function. It gives a probability value of the case status. The output probability is then rounded off to get the final output. i.e. If probability is 0.5 or ≤ 0.5 then it becomes 0, which means DENIED and if probability is > 0.5 then it becomes 1, which means CERTIFIED. Since this is a binary classification problem, we used binary cross- entropy function. For training the network, we experimented with Stochastic Gradient Descent and Adam [7] optimizers and chose Adam since it was converging faster.

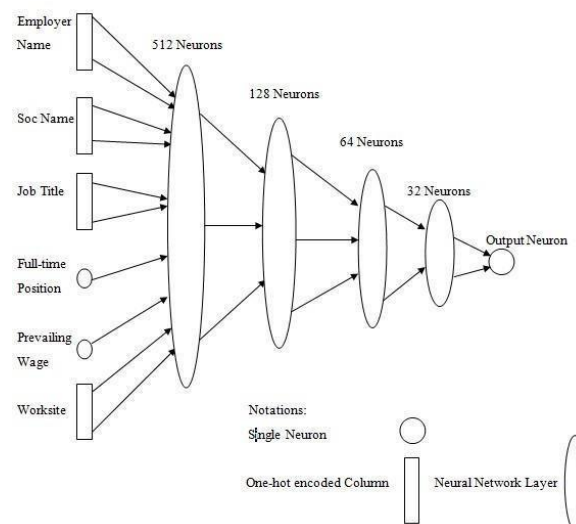


Fig 12

Associations Rule Mining

Associations Rule Mining is a machine learning method to detect patterns in large databases. The number of patterns detected depends on the size of the database. Thus, more the volume more will be the number of patterns since a large amount of data will be analyzed.

Consider a set of transactions $T = \{T_1, T_2, T_3 \dots T_n\}$ such that each transaction has a unique transaction id. Each transaction consists of items, which are a subset of I , where I is a set of items.

$$I = \{I_1, I_2, I_3 \dots I_n\}$$

A rule is given by $A \Rightarrow B$ such that $A, B \subseteq I$ and $A \cap B = \emptyset$

Support: Support is an indication of number of times the item set appears in the dataset. Thus for $A \Rightarrow B$, the support is given by

$$\text{Support}(A) = |\{t \in T; A \subseteq t\}| / |T|$$

The rule $A \Rightarrow B$ holds in the transaction set D with support s , where s is the percentage of transactions in D that contain $A \cup B$ (i.e., the union of sets A and B , or say, both A and B). This is taken to be the probability, $P(A \cup B)$. That is, $\text{support}(A \Rightarrow B) = P(A \cup B)$.

Confidence: Confidence indicates the number of times a rule is found to be true.

$$\text{Confidence}(P \Rightarrow Q) = \text{Support}(P \cup Q) / \text{Support}(P)$$

Where $\text{Support}(P \cup Q) = P(E_P \cap E_Q)$,

E_P = event that a transaction contains P

E_Q = event that a transaction contains Q

Example 1 $\{\text{pencil, eraser}\} \Rightarrow \{\text{notebook}\}$

The above association rule states that if a person buys pencil and eraser, then he is most likely to buy a notebook.

Example 2 $\{\text{Salary} > 65000, \text{California, Google}\} \Rightarrow \{\text{Certified}\}$, confidence = 0.8

This rule states that most applicants having a salary > 65000 and who work at Google, California have been issued the visa. The value of confidence indicates that the possibility of the occurrence of consequent is 80 % for the given antecedent.

Consider an applicant whose prediction for the case status is denied. Now, we can convert this input query of the applicant to an association rule as shown below.

$$\{\text{Salary} > 65000, \text{Chicago}, \text{Google}\} \Rightarrow \{\text{Denied}\}$$

From the associations generated by our model, we can recommend the applicant that applying from California will increase the chances of approval.

Logistic Regression

It is used to estimate discrete values (Binary values like 0/1, yes/no, true/false) based on given set of independent variable(s). In simple words, it predicts the probability of occurrence of an event by fitting data to a logit function. Hence, it is also known as logit regression. Since, it predicts the probability, its output values lies between 0 and 1 (as expected).

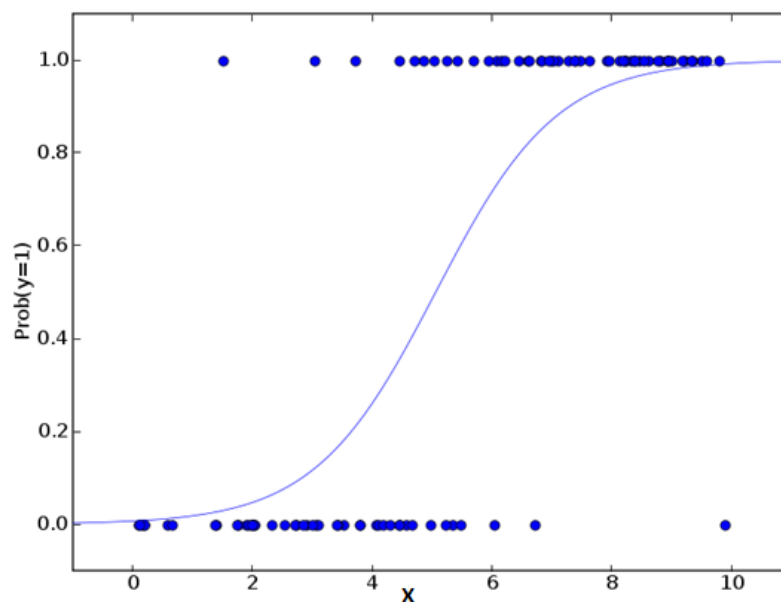


Fig 13 Logistic Regression

Random Forest Classifier

Random Forest is a trademark term for an ensemble of decision trees. In Random Forest, we've collection of decision trees (so known as "Forest"). To classify a new object based on attributes, each tree gives a classification and we say the tree "votes" for that class. The forest chooses the classification having the most votes (over all the trees in the forest). Random forest gives much more accurate predictions when compared to simple regression models in many scenarios. These cases generally have high number of predictive variables and huge sample size. This is because it captures the variance of several input variables at the same time and enables high number of observations to participate in the prediction.

Each tree is planted & grown as follows:

1. If the number of cases in the training set is N , then sample of N cases is taken at random but *with replacement*. This sample will be the training set for growing the tree.
2. If there are M input variables, a number $m \ll M$ is specified such that at each node, m variables are selected at random out of the M and the best split on these m is used to split the node. The value of m is held constant during the forest growing.
3. Each tree is grown to the largest extent possible. There is no pruning.

Gaussian NB

It is a classification technique based on Bayes' theorem with an assumption of independence between predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, a naive Bayes classifier would consider all of these properties to independently contribute to the probability that this fruit is an apple.

Naive Bayesian model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability $P(c|x)$ from $P(c)$, $P(x)$ and $P(x|c)$. Look at the equation below:

$$P(c | x) = \frac{P(x | c)P(c)}{P(x)}$$

$$P(c | X) = P(x_1 | c) \times P(x_2 | c) \times \cdots \times P(x_n | c) \times P(c)$$

Here,

- $P(c/x)$ is the posterior probability of *class (target)* given *predictor (attribute)*.
- $P(c)$ is the prior probability of *class*.
- $P(x/c)$ is the likelihood which is the probability of *predictor* given *class*.
- $P(x)$ is the prior probability of *predictor*.

CHAPTER 4

EXPERIMENTS

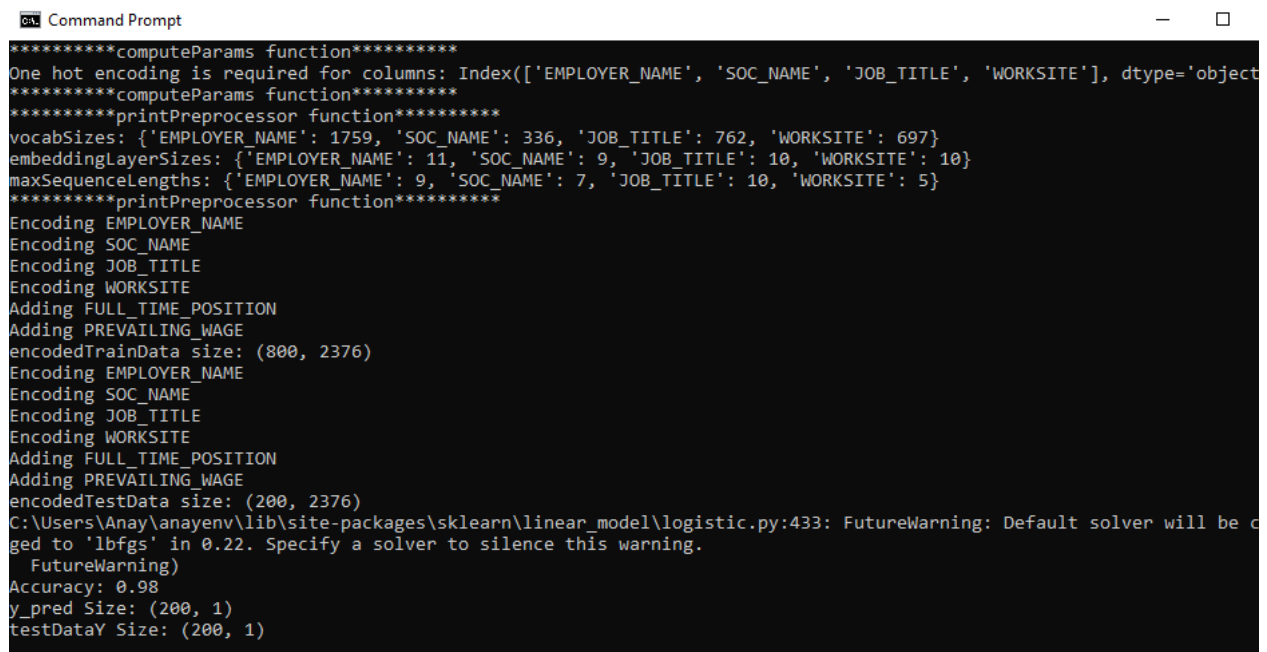
4.1 Training and Testing

The entire dataset was split into training and testing in the ratio 80:20. Further, 20% of the training dataset separated and used as validation data during training of the neural network.

The NN was trained on the training dataset for multiple epochs. After each epoch the neural network was fed validation dataset to test if the model is being trained properly or not. Once the accuracy on the validation dataset stopped improving considerably, the training was stopped.

The trained NN model was then used on the testing dataset for verification.

4.2 Results



```

*****computeParams function*****
One hot encoding is required for columns: Index(['EMPLOYER_NAME', 'SOC_NAME', 'JOB_TITLE', 'WORKSITE'], dtype='object')
*****computeParams function*****
*****printPreprocessor function*****
vocabSizes: {'EMPLOYER_NAME': 1759, 'SOC_NAME': 336, 'JOB_TITLE': 762, 'WORKSITE': 697}
embeddingLayerSizes: {'EMPLOYER_NAME': 11, 'SOC_NAME': 9, 'JOB_TITLE': 10, 'WORKSITE': 10}
maxSequenceLengths: {'EMPLOYER_NAME': 9, 'SOC_NAME': 7, 'JOB_TITLE': 10, 'WORKSITE': 5}
*****printPreprocessor function*****
Encoding EMPLOYER_NAME
Encoding SOC_NAME
Encoding JOB_TITLE
Encoding WORKSITE
Adding FULL_TIME_POSITION
Adding PREVAILING_WAGE
encodedTrainData size: (800, 2376)
Encoding EMPLOYER_NAME
Encoding SOC_NAME
Encoding JOB_TITLE
Encoding WORKSITE
Adding FULL_TIME_POSITION
Adding PREVAILING_WAGE
encodedTestData size: (200, 2376)
C:\Users\Anay\anayenv\lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
Accuracy: 0.98
y_pred Size: (200, 1)
testDataY Size: (200, 1)

```

Fig 14 Accuracy obtained by Neural Networks

```
Command Prompt
Train data size x:(800, 6) and y:(800, 1)
Test data size x:(200, 6) and y:(200, 1)
*****splitDataSet function*****
*****computeParams function*****
One hot encoding is required for columns: Index(['EMPLOYER_NAME', 'SOC_NAME', 'JOB_TITLE',
*****computeParams function*****
*****printPreprocessor function*****
vocabSizes: {'EMPLOYER_NAME': 1759, 'SOC_NAME': 336, 'JOB_TITLE': 762, 'WORKSITE': 697}
embeddingLayerSizes: {'EMPLOYER_NAME': 11, 'SOC_NAME': 9, 'JOB_TITLE': 10, 'WORKSITE': 10}
maxSequenceLengths: {'EMPLOYER_NAME': 9, 'SOC_NAME': 7, 'JOB_TITLE': 10, 'WORKSITE': 5}
*****printPreprocessor function*****
Encoding EMPLOYER_NAME
Encoding SOC_NAME
Encoding JOB_TITLE
Encoding WORKSITE
Adding FULL_TIME_POSITION
Adding PREVAILING_WAGE
encodedTrainData size: (800, 2376)
Encoding EMPLOYER_NAME
Encoding SOC_NAME
Encoding JOB_TITLE
Encoding WORKSITE
Adding FULL_TIME_POSITION
Adding PREVAILING_WAGE
encodedTestData size: (200, 2376)
Accuracy: 0.97
y_pred Size: (200, 1)
testDataY Size: (200, 1)
(anayenv) C:\Users\Anay\Desktop\H1B>
```

Fig 15 Accuracy obtained by Random Forest Classifier

❏ Command Prompt

```

Train data size x:(800, 6) and y:(800, 1)
Test data size x:(200, 6) and y:(200, 1)
*****splitDataSet function*****
*****computeParams function*****
One hot encoding is required for columns: Index(['EMPLOYER_NAME', 'SOC_NAME', 'J
*****computeParams function*****
*****printPreprocessor function*****
vocabSizes: {'EMPLOYER_NAME': 1759, 'SOC_NAME': 336, 'JOB_TITLE': 762, 'WORKSITE
embeddingLayerSizes: {'EMPLOYER_NAME': 11, 'SOC_NAME': 9, 'JOB_TITLE': 10, 'WORK
maxSequenceLengths: {'EMPLOYER_NAME': 9, 'SOC_NAME': 7, 'JOB_TITLE': 10, 'WORKSI
*****printPreprocessor function*****
Encoding EMPLOYER_NAME
Encoding SOC_NAME
Encoding JOB_TITLE
Encoding WORKSITE
Adding FULL_TIME_POSITION
Adding PREVAILING_WAGE
encodedTrainData size: (800, 2376)
Encoding EMPLOYER_NAME
Encoding SOC_NAME
Encoding JOB_TITLE
Encoding WORKSITE
Adding FULL_TIME_POSITION
Adding PREVAILING_WAGE
encodedTestData size: (200, 2376)
Accuracy: 0.875
y_pred Size: (200, 1)
testDataY Size: (200, 1)
(anayenv) C:\Users\Anay\Desktop\H1B>

```

Fig 16 Accuracy obtained by Gaussian NB

Table 1 indicates testing accuracy of different algorithms.

Algorithm	Accuracy
Logistic Regression	93%
Random Forest Classifier	97%
GaussianNB	87.50%
Neural Networks	98%

Table 1.0

Since, neural networks gave highest accuracy; we have used neural networks for final prediction of case status.

4.3 Recommendations

Using Cosine Similarity

We used cosine similarity on the testing input and the data from the training set to find the most similar applicant whose H1B visa was approved. Recommendations are then generated by the model by one-to-one attribute match of the test data with the most similar applicant's data. Instead of picking just the top most similar match, we can pick multiple such matches based on cosine values and use these matches to generate multiple recommendations.

Fig 14.0 shows such sample recommendations.

```
test input is.....
EMPLOYER_NAME      SOC_NAME      JOB_TITLE      FULL_TIME_POSITION      PREVAILING_WAGE
9 PEDDLER COFFEE LLC  GENERAL AND OPERATIONS MANAGERS  GENERAL MANAGER          1          0.255565  N

[EMPLOYER_NAME:99TH AVENUE HOLDINGS LLC,SOC_NAME:CHIEF EXECUTIVES,JOB_TITLE:CHIEF EXECUTIVE OF FITNESS MANAGEMEN
919,WORKSITE:NEW YORK, NEW YORK,CASE_STATUS:1.0,CosineValues:0.3290454444009676,}

[EMPLOYER_NAME:CALVARY DESIGN TEAM, INC.,SOC_NAME:CHIEF EXECUTIVES,JOB_TITLE:PRESIDENT,FULL_TIME_POSITION:1.0,P
ORK,CASE_STATUS:1.0,CosineValues:0.273878165319133,}

[EMPLOYER_NAME:TMS PLUMBING AND HEATING CORP,SOC_NAME:CHIEF EXECUTIVES,JOB_TITLE:OPERATIONS DIRECTOR,FULL_TIME_
NEW YORK, NEW YORK,CASE_STATUS:1.0,CosineValues:0.27194801109949124,}

A person having a profile similar to your profile and having following features has been issued the visa.....
EMPLOYER_NAME: 99TH AVENUE HOLDINGS LLC
SOC_NAME: CHIEF EXECUTIVES
JOB_TITLE: CHIEF EXECUTIVE OF FITNESS MANAGEMENT
PREVAILING_WAGE: 187200.0
```

Fig. 17 Input & its three nearest certified applicants with approved case status

Using Associations Rule Mining

Similar approach was used to generate recommendations using Associations rule mining.

```

IPython console
Console 1/A
>frozenset(['CERTIFIED'])
Confidence is .....0.9375
Support is ....0.166666666667
Lift is .....1.01656626506

frozenset(['Y', ' FLORIDA', 'CHIEF EXECUTIVES'])-->frozenset(['CERTIFIED'])
Confidence is .....0.9375
Support is ....0.166666666667
Lift is .....1.01656626506

frozenset(['Y', ' FLORIDA', 'Salary greater than 65000'])-->frozenset(['CERTIFIED'])
Confidence is .....0.9375
Support is ....0.166666666667
Lift is .....1.01656626506

frozenset(['Y', 'CHIEF EXECUTIVES', ' CALIFORNIA', 'Salary greater than 65000'])--
>frozenset(['CERTIFIED'])
Confidence is .....0.944444444444
Support is ....0.188888888889
Lift is .....1.02409638554

frozenset(['Y', ' FLORIDA', 'CHIEF EXECUTIVES', 'Salary greater than 65000'])--
>frozenset(['CERTIFIED'])
Confidence is .....0.9375
Support is ....0.166666666667
Lift is .....1.01656626506

Input : State: California  FULL Time: Y  Salary: 66,667
A similar user in FLORIDA has been issued the visa

```

Fig 18 Shows association rules generated and recommendation suggested by the model based on the rules

CHAPTER 5

FUTURE SCOPE

We have not yet evaluated our recommendation system. The evaluation of this recommendation will be done in future. We plan to make the changes to the user input as per recommendations and then feed the update input to the neural network to evaluate the recommendations.

CHAPTER 6

REFERENCES

- [1] Prediction of H1B Visa using Machine Learning Algorithms (in press).
- [2] Predicting filed H1-B Visa Petitions' Status. International Research Journal of Engineering and Technology(IRJET)
- [3] <http://athena.ecs.csus.edu/~poosarla/img/proposal.pdf>
- [4] Recruiting high skill labour in north america: Policies, outcomes and futures. International Migration, 52(3):40–54, 2014
- [5] Student flows and migration: An empirical analysis. 2005
- [6] H-1B Visa Data Analysis and Prediction by using K-means Clustering and Decision Tree Algorithms. [Online] Available:
<https://github.com/Jinglin-LI/H1B-VisaPrediction-by-MachineLearningAlgorithm/blob/master/H1B\%20Prediction\%20Research\percentage20Report.pdf>.
- [7] ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION. ICLR 2015

