H1-B Visa Status Prediction using Deep Learning Models

Sai Sri Meghana Dharmapuri†
Computer Science
California State University-
Sacramento, CA, US
sdharmapuri@csus.edu

Prerak Shah, Umeshchandra
Computer Science
California State University-
Sacramento, CA, US
prerakumeshchandras@csus.edu

ABSTRACT

In recent times, H1B visas have become the most demanded visa. It is a visa that is filed for highly skilled foreign nationals every year. The allocation of these visas is said to be a lottery method which does not rely on any criteria. There is a sense of uncertainty, employers find talents that fit their requirement, but they are unable to provide a sense of security. The H1B visa application process depends on factors like salary, work location, full-time employment etc. In this project, we are given a data based on the visa applications filed from 2011 to 2016. We processed the data and used Sklearn train test split method to randomly split into training set and test set, so that we can run model on the training set and measure the performance on the test set. The baseline models were Naïve Bayes, Random Forest, and XGBoost. Our aim is to outperform their results and train the data on new deep learning models like Neural Networks and Convolutional Neural Networks.

CCS CONCEPTS

• Logistic Regression • Convolutional Neural Networks • Neural Network • SVM • Random Forest • Deep learning

KEYWORDS

Data analysis, Data preprocessing, AI models, machine learning.

1 Introduction

A visa is typically a stamp issued to an alien (foreigner) to visit a country for purposes ranging from travel to business. There are many classes of visa depending on the country to which one intends to travel. The United States of America also has many classes like H1, L1, J1 etc. The specific category of visa that caught the limelight for all wrong reasons recently in the USA is the H-1B. It is a temporary non-immigrant work visa created in 1989 by the USA to attract highly skilled foreign workers and to tackle the severe labor shortages in the country. Here we stress the term 'highly skilled foreign workers' as the visa is offered to people who have exceptional skills in their field and often with an advanced educational background. [2]

The H-1B is a temporary (nonimmigrant) visa category that allows employers to petition for highly educated foreign professionals to work in "specialty occupations" that require at least a bachelor's degree[13] or the equivalent. Jobs in fields such as mathematics, engineering, technology, and medical sciences often qualify [1]. The number of applications increased rapidly due to which a lottery system was later introduced because it is practically not possible to issue visas to everyone. This work focuses on the data science challenge problem of predicting the decision for H1B visa status using Logistic Regression, Naïve Bayes, Random Forest, XG-Boost, Neural Networks, CNN. This project aims to use discriminative machine learning techniques to identify these petitions and predict each petition's "case status," based on different factors.

This paper is organized as follows. Section 2 describes our model and our approach to the H1B visa prediction problem. Section 3 describes the various algorithms used in this project and Section 4 discusses the dataset, data preprocessing and the results obtained. Section 5 outlines the related work in the industry on H1B visa prediction, section 6 has the conclusion and section 7 describes the work division and section 8 has learning experience.

2 Problem Formulation

The goal of the project is to classify the H1B status based on the data provided by Office of Foreign Labor Certification (OFLC). We used Employer name, Prevailing wage, SOC (Standard Occupational Classification) name, Job title, Full time position, year and worksite as the inputs and predicted the case status as the output. We predicted the Case status as 1 and 0. 1 represents the certified while 0 represents the denied status. We added multiple features like Occupation, and State extracted from the columns to improve our efficiency of our model and to derive meaningful insights. Additionally, we selected the important features from running on the classification models and ran on the deep learning models.

3 System/ Algorithm design

The focus is on Logistic regression (LR), LR with L1(Lasso) penalty, Naïve Bayes, Random Forest, XGBoost, Neural Networks, Convolution Neural Networks (CNN).

3.1 Logistic Regression

The logistic regression is a predictive analysis. Logistic regression is used to characterize data and illustrate the relationship between one dependent binary variable and one or more independent variables of nominal, ordinal, interval, or ratio point. Logistic regression uses the sigmoid function as its hypothesis as defined in

(1) to make sure hθ(x) ∈ (0,1). It makes the assumptions in (2) and finds the θ that maximizes the log likelihood of the data, l(θ), which is defined in (3).[3]

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}} \tag{1}$$

$$\begin{aligned} P(y=1|x;\theta) &= h_\theta(x) \\ P(y=0|x;\theta) &= 1 - h_\theta(x) \end{aligned} \tag{2}$$

$$l(\theta) = \sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log h_\theta(1-x^{(i)}) \tag{3}$$

3.2 Logistic Regression with L1 penalty

L1 regularization, uses a penalty term which encourages the sum of the absolute values of the parameters to be small. It has frequently been observed that L1 regularization in many models causes many parameters to equal zero, so that the parameter vector is sparse. This makes it a natural candidate in feature selection settings, where we believe that many features should be ignored [8].

3.3 Naïve Bayes Theorem

Naïve Bayes is a classification technique based on Bayes Theorem with an assumption of independence among predictors. A Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. The Naive Bayes algorithm reduces the complexity of the Bayesian classifier by assuming of conditional independence over the training dataset to make it practical. Naive Bayes assumes all features are conditionally independent given labels. It calculates p(x|y = 1), p(x|y = 0) and p(y) by taking their maximum likelihood estimates in the joint likelihood of the data. While making a prediction, it calculates (4) both for p(y=1) and p(y=0) using the Bayes rule and compares the two.

$$p(y=1|x) = \frac{\prod_{i=1}^{m} p(x_i|y=1)p(y=1)}{\prod_{i=1}^{m} p(x_i|y=1)p(y=1) + \prod_{i=1}^{m} p(x_i|y=0)p(y=0)} \tag{4}$$

3.4 Random Forest

Random Forest Trees is supervised machine learning algorithm. This algorithm works on the process of building multiple decision trees and merging them together. It can be used for regression and classification problems. There is a direct relationship between the results and the number of trees in the forest. Thus, larger the number of trees more will be the accuracy[4]. Random Forests algorithm is a variance minimizing algorithm that uses randomness when making split decision to help avoid overfitting on the training data. If D(x, y) denotes the training dataset, each classification tree in the ensemble is built using a different subset Dθk (x, y) ⊂ D(x, y) of the training dataset. The final output y is obtained by aggregating the results:

$$y = \text{argmax}_{p \in \{h(x_1)..h(x_k)\}} \left\{ \sum_{j=1}^{k} (I(h(x|\theta_j) = p)) \right\} \tag{5}$$

3.5 XG-Boost

XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. Boosting refers to the ensemble learning technique of building many models sequentially, with each new model attempting to correct for the deficiencies in the previous model. XGBoost is normally used to train gradient-boosted decision trees and other gradient boosted models.

3.6 Neural Networks

Neural Networks (NNs) are complex multi-layer structures that consist of units called "neurons". Each neuron introduces a non-linearity with a chosen activation function g(x) after weighing its inputs and propagates the information to the next layer. At the output layer, a prediction y is made. The cost function NN tries to minimize is given in (6) and the vectorized version of forward propagation for a NN with input, one hidden and output layer is shown in (7). Finally, the parameters, W and b, in each layer l are updated as in (8) using backpropagation and some form of gradient descent [3]. In our implementation we used 64 neurons in the input layer and 32 neurons in the hidden layer with one output neuron. The activation functions used are tanh and Sigmoid.

$$\mathcal{L}(\hat{y}, y) = -\left[(1-y)\log(1-\hat{y}) + y \log \hat{y}\right] \tag{6}$$

$$\begin{aligned} Z^{[1]} &= W^{[1]}X + b^{[1]} \\ A^{[1]} &= g(Z^{[1]}) \\ Z^{[2]} &= W^{[2]}A^{[1]} + b^{[2]} \\ \hat{y} &= g(Z^{[2]}) \end{aligned} \tag{7}$$

$$\begin{aligned} W^{[l]} &= W^{[l]} - \alpha \frac{\partial \mathcal{L}}{\partial W^{[l]}} \\ b^{[l]} &= b^{[l]} - \alpha \frac{\partial \mathcal{L}}{\partial b^{[l]}} \end{aligned} \tag{8}$$

3.6 Convolution Neural Networks

A Convolutional Neural Network (CNN) takes input as an image, assign importance to various aspects/objects in the image to differentiate one part from another. The Convolution layer helps in reducing the image size for processing, so that the data which is critical for prediction is not lost.

Layer1: Conv filter = 64, Kernel size = 3, MaxPool = 2
Layer2: Conv filter = 64, Kernel size = 3, MaxPool = 2
Dropout layer: 0.5, Maxpooling layer : pool size = 2
Dense layer: units = 100

Dense layer: units = 2
Output layer

Figure 1: Convolution Neural Network Architecture

The Pooling layer is responsible for reducing the spatial size of the Convolved Feature. Pooling layer helps in decreasing the computational power requirement for data processing [8]. We changed the shape from 2D to 3D to implement a Conv1D CNN. The architecture used is shown in Figure 1.

4 Experimental Evaluation

4.1 Methodology

4.1.1 Dataset

The dataset being used for this project is taken from Kaggle listed under the name \H-1B Visa Petitions 2011-2016 dataset". [https://www.kaggle.com/nsharan/h-1b-visa].

The columns in the dataset are:

1) CASE_STATUS: This provides us with the information of the data classes. The data classes are: (1) Certified (2) Denied (3) Withdrawn (4) Certified -Withdrawn.
2) EMPLOYER_NAME: Details containing the name of the employer.
3) SOC_NAME: The occupation code for the employment.
4) JOB_TITLE: Title of the job
5) FULL_TIME_POSITION: describes whether the position is full time or part time
6) PREVAILING_WAGE: Prevailing wage is the average wage paid to employees with similar qualifications in the intended area of employment.
7) YEAR: The year when the petition is filed.
8) WORKSITE: The address in terms of the city and state of the applicant's job.
9) lon & lat: Exact geographical location of the worksite.

4.1.2 Data preprocessing

The dataset is loaded, and the unnamed column is renamed to CASE_ID after which we check for the null values and drop the columns that do not play an important role in predicting the output. In our implementation we dropped CASE_ID, lon, and lat. We deal with each of the column names separately.

In CASE_STATUS, we drop the rows with null values. The next step is to count the values for each of the 7 individual labels, i.e. Certified, Certified-Withdrawn, Denied, Withdrawn, Pending Quality and Compliance Review – Unassigned, Rejected, Invalidated. Among these labels we drop the labels that are based on the user. With the remaining labels we separate them into two labels, i.e. Certified and Denied. In Figure 2, we notice that there are more petitions under the certified label than the denied label.

Based on the data given, we tried to analyze the number of petitions filed per year in figure 3 and ratio of acceptance per year in Figure 4. As mentioned earlier, we can notice in the graphs as to how the number of applications along with the acceptance criteria increased over the years.

We also analyzed the top 20 designations where the visa is applied the most as shown in Figure 5. According to the plot it is understood that designations like programmer analyst, software engineer, computer programmers and system analysts are the types of jobs where H1B visas are highly applied.

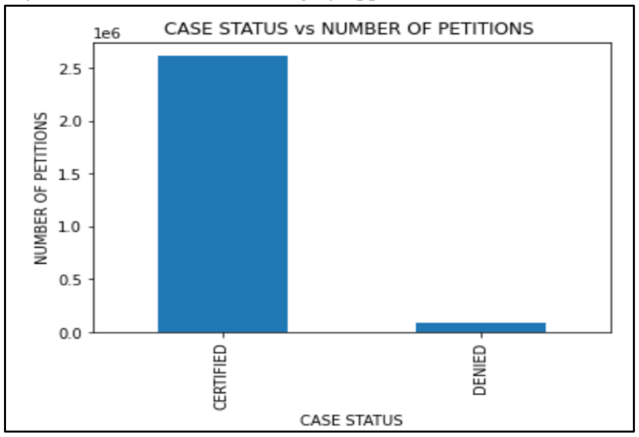


Figure 2: Certified/ Denied vs Number of petitions

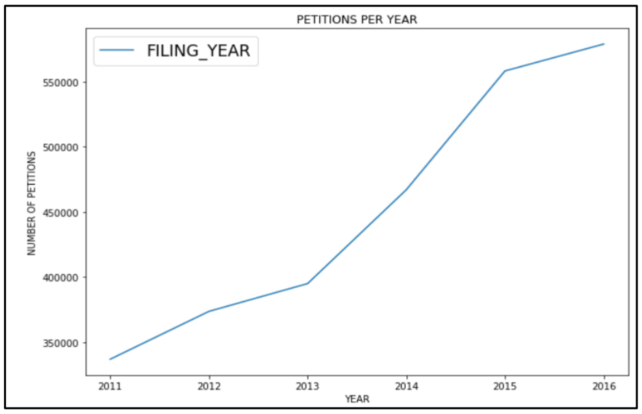


Figure 3: Number of petitions per year

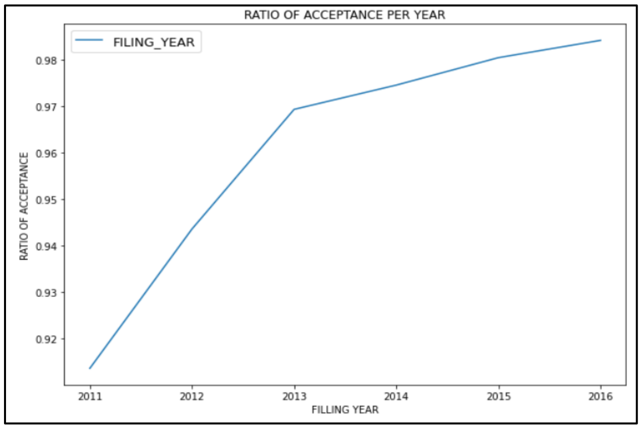


Figure 4: Ratio of acceptance per year

We were also able to analyze the top 20 cities from where the applications were filed, it is seen that places like New York, Texas and San Francisco are the areas where maximum applications are filed as seen in Figure 6.

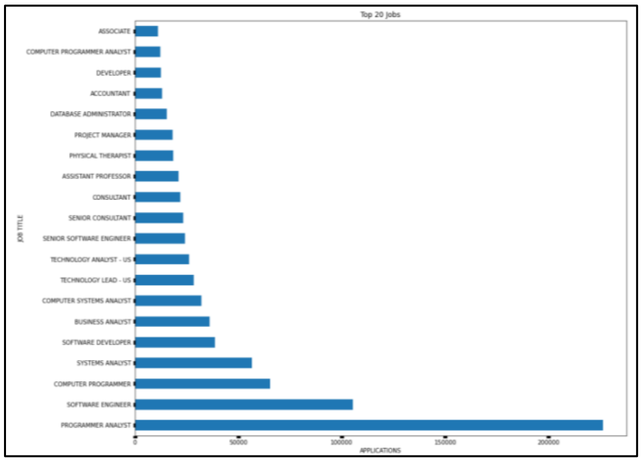


Figure 5: Job Title VS Applications

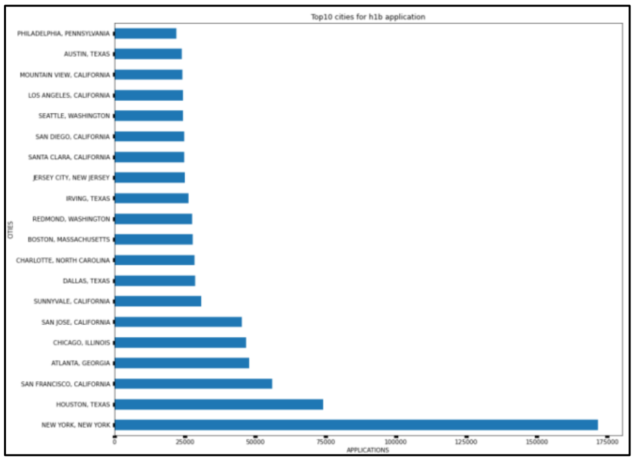


Figure 6: Top 20 cities for visa applications

We performed basic statistical operations like mean and mode. Mode was used for the columns EMPLOYER_NAME, SOC_NAME, JOB_TITLE, FULL_TIME_POSITION to replace the null values for object datatype by the mode of that column. Mean was used for the PREVAILING_WAGE column to replace the null values with the mean of the data as is of float datatype.

When analyzing the prevailing wage, we checked for the minimum, maximum, the mode and the median to understand the range of salaries that were involved according to our dataset and checked for the outliers after which we took the values of the 25th and the 80th quantile to overcome the outliers spotted in Figure 7.

When noticed that one column is more weighted than other columns, we perform transformation. When different columns have different range of values, statistical measures do not give a proper result and hence we used a log method and then performed a cube root transformation on the logarithmic values of the column.

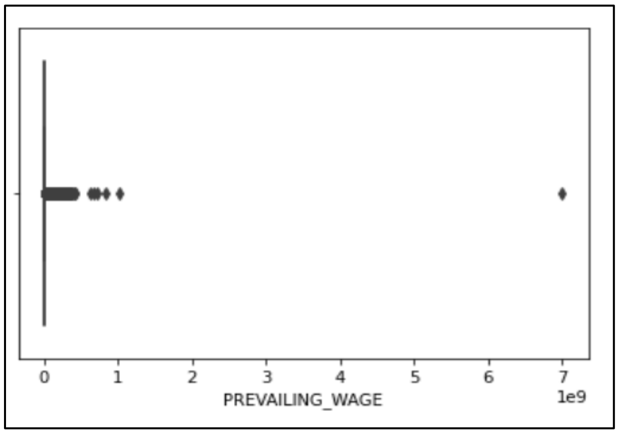


Figure 7: Prevailing wage outliers

For the Employer column, in order to correctly map and check the University string, you should first convert all the strings into the same case and hence we converted all the values to lower case. All the data with the keyword university, limited liability company, corporation are put under EMPLOYER_NAME. All the data with the non-university will be put under EMPLOYER.

For SOC_NAME, there are many values, so a new function is constructed which will include the applicant's significant profession, mapping it to the SOC NAME value. As we can see in Figure 7, Computer Occupation has the highest number of H1B applications. Since most of the information related to Visa have the State to be an important factor, we split the states from the WORKSITE column. A graph is plotted and according to the analysis in figure 8, California has the highest number of petitions, followed by Texas and then New York.

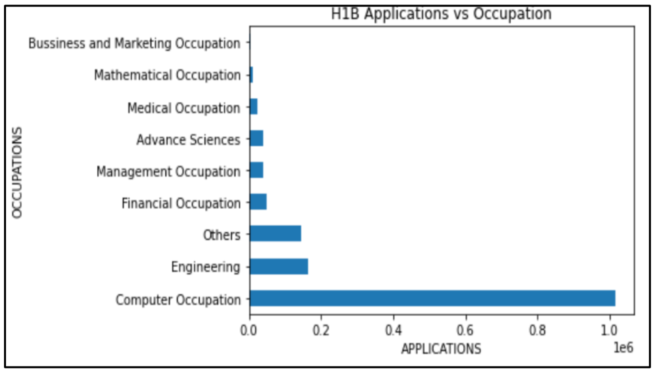


Figure 8: H1B applications Vs occupation

After this process, we drop all the duplicate columns and map the target variables. Since we try to convert the target to binary, we map CERTIFIED as 1 and DENIED as 0. One hot encoding is performed on FULL_TIME_POSITION, YEAR, EMPLOYER, OCCUPATION, STATE.

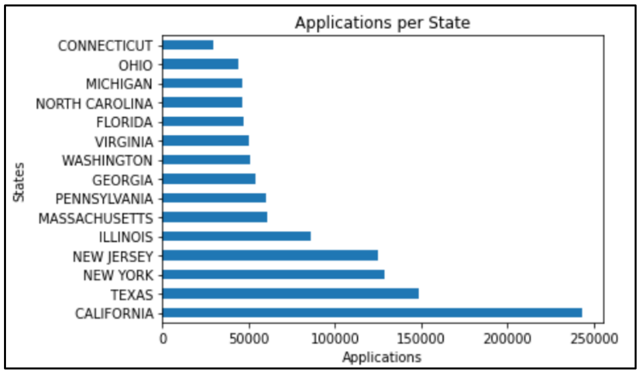


Figure 9: Applications per State

The next step is to split the data into x and y, where Standard Scaler preprocessing has been done on the data related to 'x'. StandardScaler standardizes a feature by subtracting the mean and then scaling to unit variance. It transforms the data in a manner that it has mean as 0 and standard deviation as 1. According to the data we have, we can notice that the number of 1's is greater than the number of 0's.

In order to reduce the huge difference in the size of the data, we perform Downsampling. After trying different techniques, like Smote and Near Miss, we used the Near miss technique in our work. It aims to balance class distribution by randomly eliminating majority class examples. To prevent problem of information loss in most under-sampling techniques, near-neighbor methods are widely used. Further the data is split with an 80:20 ratio.

4.1.3 Feature Selection

Feature selection is performed to enable the machine learning algorithm to train faster. It reduces the complexity of a model and makes it easier to interpret.

For the Logistic Regression and Logistic Regression with L1 penalty, we used the inbuilt attribute coef_ of the Logistic regression model. We used the inbuilt function feature_importances_ for the random forest model and the XGBoost model.

4.1.4 Metrics

We used F1 score to be the basis to the comparison of the work that was done earlier to the work that we did. We implemented Logistic Regression, Random forest, XG boost, Naïve Bayes as implemented in base project.

Models	Existing_Work_F1Score	New_Work_F1Score
Naïve Bayes	0.73	0.85
Random Forest	0.79	0.96
XG Boost	0.80	0.90

Figure 10: Comparison of existing work with our implementation using F1 score to be the metric

4.2 Results

4.2.1 Logistic Regression

We were able to secure an accuracy of 0.857 for the Logistic regression model.

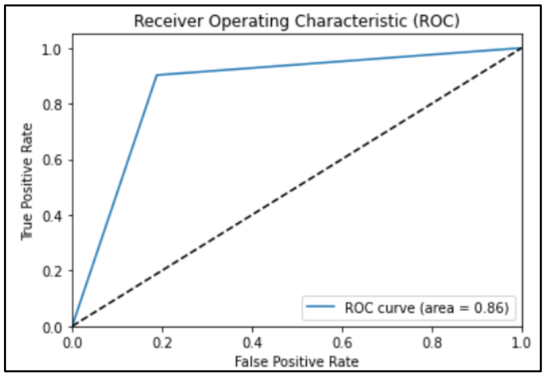
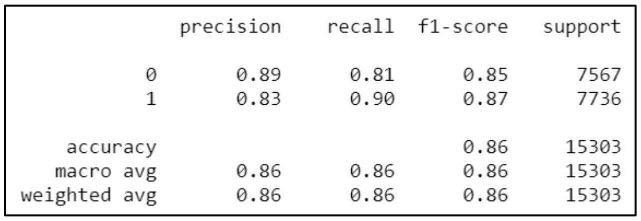


Figure 11: ROC curve

The ROC curve shows the trade-off between sensitivity (or TPR) and specificity (1 – FPR). Classifiers that give curves closer to the top-left corner indicate a better performance [7]. From the figure 11 we understand that the curve is away from the 45-degree diagonal of the ROC space, so the test is accurate thereby giving an F1 score of 0.86 as in Figure 12.


```
              precision    recall  f1-score   support

           0       0.89      0.81      0.85      7567
           1       0.83      0.90      0.87      7736

    accuracy                           0.86     15303
   macro avg       0.86      0.86      0.86     15303
weighted avg       0.86      0.86      0.86     15303
```
Figure 12: Classification report for LR

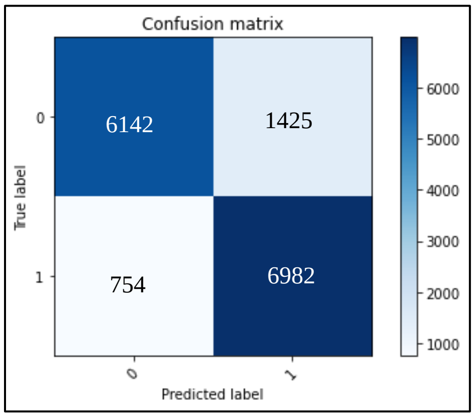


Figure 13:Confusion Matrix for LR

The confusion matrix shows the ways in which the classification model is confused when it makes predictions. 6982 applications were correctly predicted as certified, while 1425 applications were incorrectly predicted as certified. 6142 applications were correctly predicted as denied, while 754 applications were incorrectly predicted as denied.

4.2.2 Logistic Regression with L1(Lasso) penalty

The advantage of this technique is that it shrinks the less important feature's coefficient to zero, thereby removing some features altogether. Since we had many features, we implemented the L1 penalty.

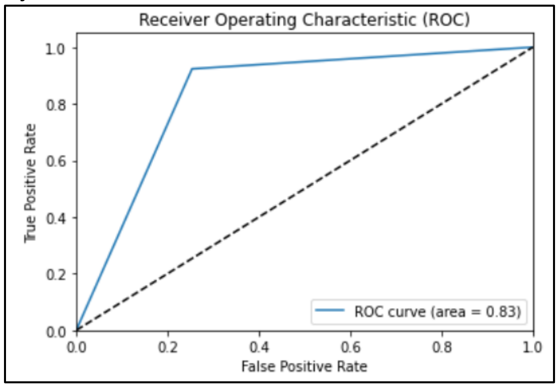
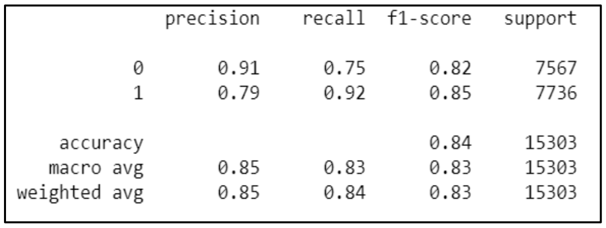


Figure 14: ROC curve

From the figure 14 we understand that test performed is accurate thereby giving an F1 score of 0.84 as in Figure 15.



```
              precision    recall  f1-score   support

           0       0.91      0.75      0.82      7567
           1       0.79      0.92      0.85      7736

    accuracy                           0.84     15303
   macro avg       0.85      0.83      0.83     15303
weighted avg       0.85      0.84      0.83     15303
```

Figure 15: Classification report for LR with L1 penalty

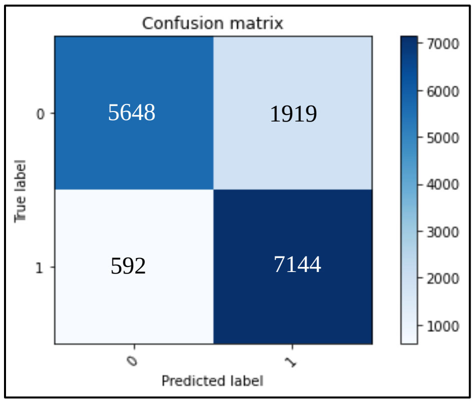


Figure 16: Confusion Matrix for LR with L1 penalty

7144 applications were correctly predicted as certified, while 1919 applications were incorrectly predicted as certified. 5648 applications were correctly predicted as denied, while 592 applications were incorrectly predicted as denied.

4.2.3 Naïve Bayes

The ROC curve does not depend on the class distribution. From the figure 17 we notice that the ROC curve is much closer to the y axis than the previous models discussed thereby giving an F1 score of 0.85 as in Figure 18.

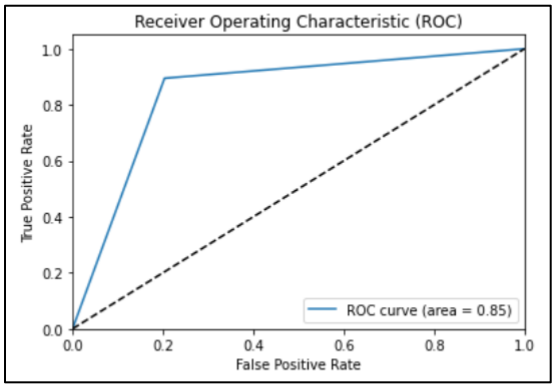
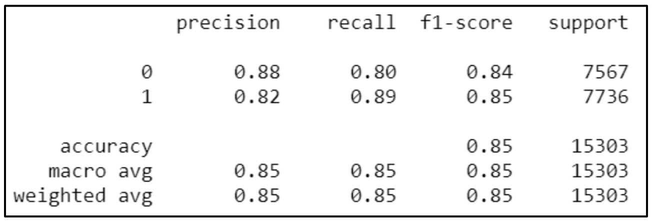


Figure 17: ROC curve for Naive Bayes



```
              precision    recall  f1-score   support

           0       0.88      0.80      0.84      7567
           1       0.82      0.89      0.85      7736

    accuracy                           0.85     15303
   macro avg       0.85      0.85      0.85     15303
weighted avg       0.85      0.85      0.85     15303
```

Figure 18: Classification report for Naïve Bayes

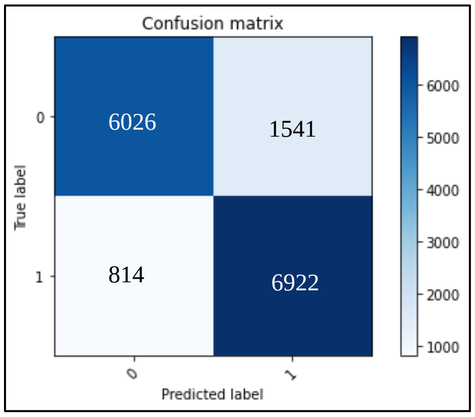


Figure 19: Confusion Matrix for Naive Bayes

6922 applications were correctly predicted as certified, while 1541 applications were incorrectly predicted as certified. 6026 applications were correctly predicted as denied, while 814 applications were incorrectly predicted as denied.

4.2.4 Random Forest

The ROC curve as observed in Figure 20 shows that the tests are the best than the previously observed curves because the curve is completely away from the 45-degree line resulting in the F1 score to be 0.96 as in Figure 21.

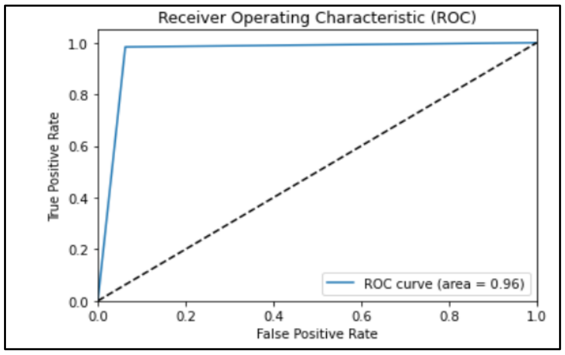
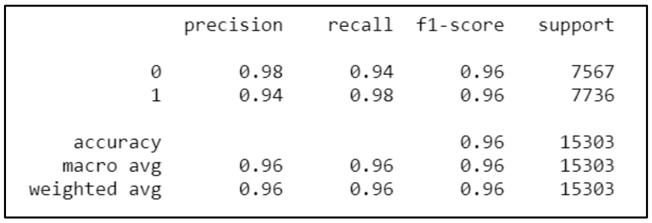


Figure 20: ROC curve



```
              precision    recall  f1-score   support

           0       0.98      0.94      0.96      7567
           1       0.94      0.98      0.96      7736

    accuracy                           0.96     15303
   macro avg       0.96      0.96      0.96     15303
weighted avg       0.96      0.96      0.96     15303
```

Figure 21: Classification report for Random Forest

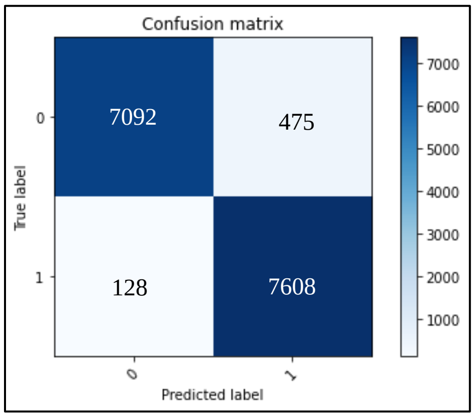


Figure 22: Confusion Matrix for Random Forest

7608 applications were correctly predicted as certified, while 475 applications were incorrectly predicted as certified. 7092 applications were correctly predicted as denied, while 128 applications were incorrectly predicted as denied. This model predicted the most correct values; therefore, it has the highest accuracy.

4.2.5 XG-Boost

While we were able to improve the F1 Score (0.90) as in Figure 23 from the existing work. We notice that XG boost performed good but not in the comparison of Random Forest when comparing figure 20 and figure 21.

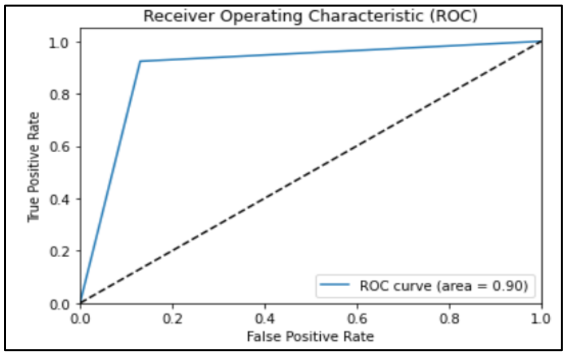
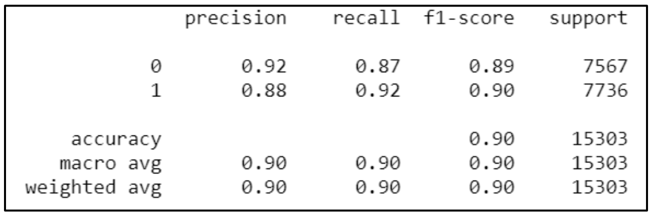


Figure 23: ROC curve



```
              precision    recall  f1-score   support

           0       0.92      0.87      0.89      7567
           1       0.88      0.92      0.90      7736

    accuracy                           0.90     15303
   macro avg       0.90      0.90      0.90     15303
weighted avg       0.90      0.90      0.90     15303
```

Figure 24: Classification report for XGBoost

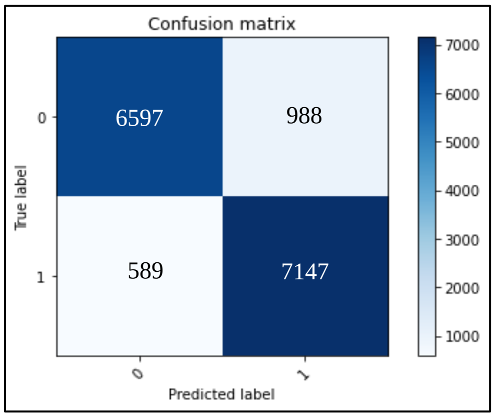


Figure 25: Confusion Matrix for XGBoost

7147 applications were correctly predicted as certified, while 988 applications were incorrectly predicted as certified. 6597 applications were correctly predicted as denied, while 589 applications were incorrectly predicted as denied.

4.2.3 Neural Networks

Approach 1:

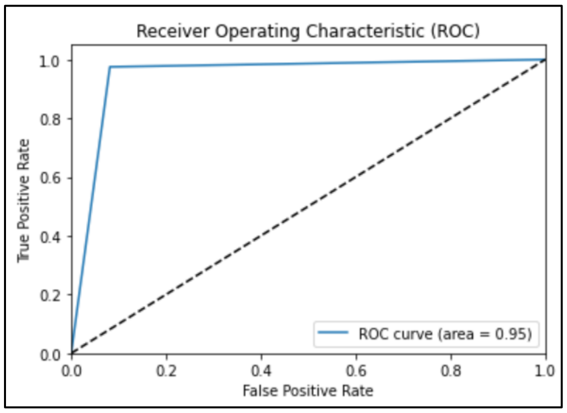
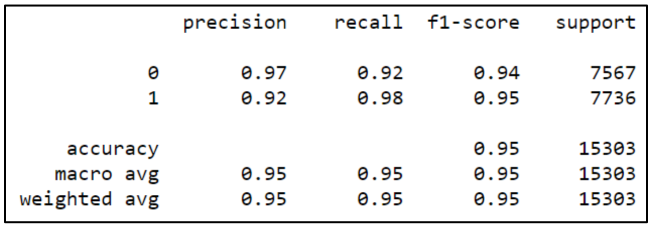


Figure 26: ROC Curve

In addition to all the existing work for predicting the H1B visa status, we tried to implement neural networks with the combination of (64, 32, 1) neurons and activation layers to be tanh and adam. We further notice an F1 score of 0.94 from the Figure 21 and a ROC curve in figure 22 as good as the best model so far, i.e. Random Forest



```
              precision    recall  f1-score   support

           0       0.97      0.92      0.94      7567
           1       0.92      0.98      0.95      7736

    accuracy                           0.95     15303
   macro avg       0.95      0.95      0.95     15303
weighted avg       0.95      0.95      0.95     15303
```

Figure 27: Classification report for Neural Network

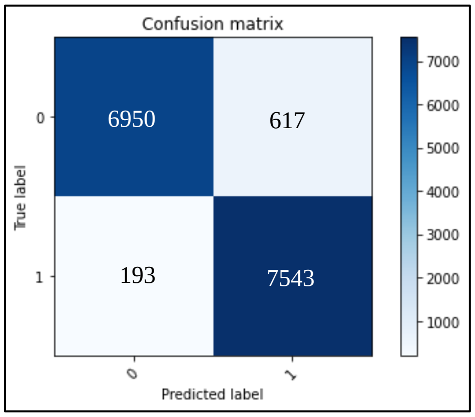


Figure 28: Confusion Matrix for Neural Networks

7543 applications were correctly predicted as certified, while 617 applications were incorrectly predicted as certified. 6950 applications were correctly predicted as denied, while 193 applications were incorrectly predicted as denied.

Approach 2:

We trained the Neural Networks model with the top few features from the Logistic Regression, L1 Logistic Regression, Random Forest and XGBOOST with the combination of (64, 32, 1) with the activation layers tanh and adam.

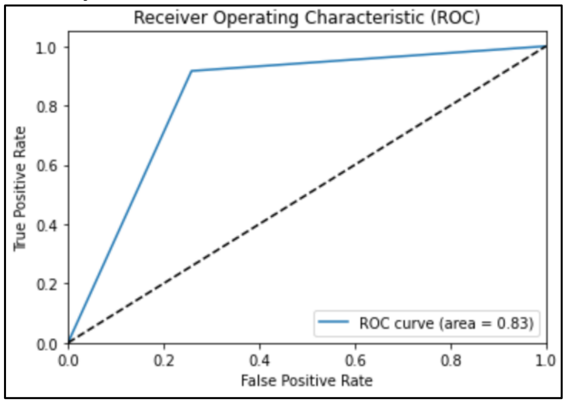
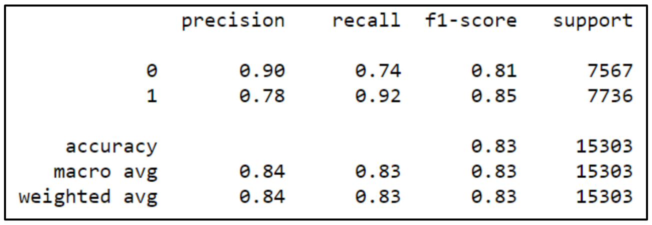


Figure 29: ROC Curve

We were not able to obtain higher F1 score as compared to all features as it can be seen in Figure 29 and Figure 30. This concluded that all the feature was equally important for the outcome of the result.



```
              precision    recall  f1-score   support

           0       0.90      0.74      0.81      7567
           1       0.78      0.92      0.85      7736

    accuracy                           0.83     15303
   macro avg       0.84      0.83      0.83     15303
weighted avg       0.84      0.83      0.83     15303
```

Figure 30: Classification report for NN - Approach 2

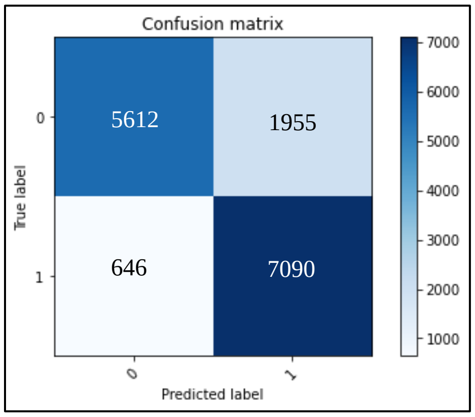


Figure 31: Classification report for NN - Approach 2

7090 applications were correctly predicted as certified, while 1955 applications were incorrectly predicted as certified. 5612 applications were correctly predicted as denied, while 646 applications were incorrectly predicted as denied.

4.2.4 Convolution Neural Network (CNN)

Approach 1:

Additionally, to the Neural Networks implementation we tried to do CNN for the non-image dataset used in this project. In figure 32, we notice that the ROC curve is as good as our best models so far, i.e. Random Forest and Neural Networks with an F1 score of 0.93 as observed in figure 33.

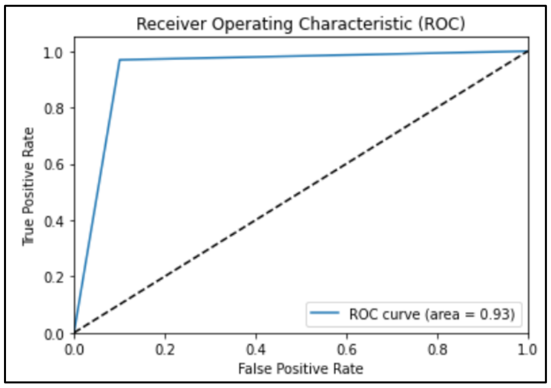
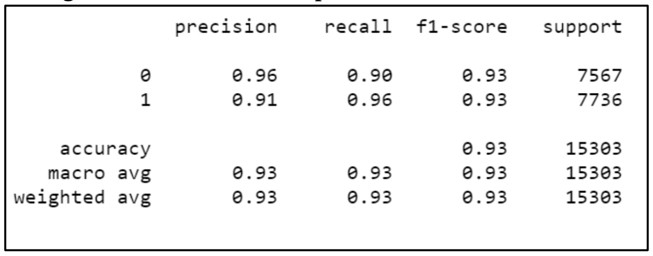


Figure 32: ROC Curve

Figure 33:Classification report for Convolution Neural



```
              precision    recall  f1-score   support

           0       0.96      0.90      0.93      7567
           1       0.91      0.96      0.93      7736

    accuracy                           0.93     15303
   macro avg       0.93      0.93      0.93     15303
weighted avg       0.93      0.93      0.93     15303
```

Network (CNN)

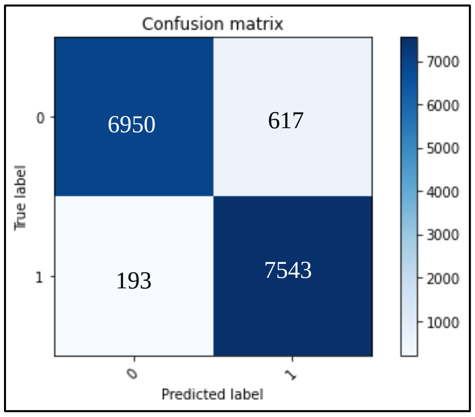
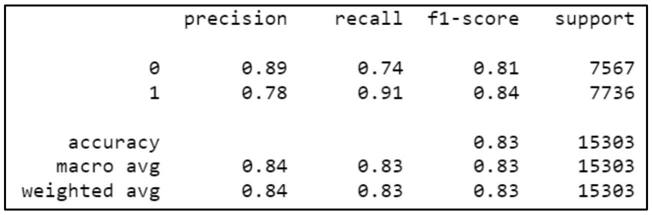


Figure 34: Confusion Matrix for Convolution Neural Network (CNN)

7543 applications were correctly predicted as certified, while 617 applications were incorrectly predicted as certified. 6950 applications were correctly predicted as denied, while 193 applications were incorrectly predicted as denied.

Approach 2:

On training the Convolution Neural Networks (CNN) with the top few features from Logistic Regression, L1 Logistic Regression, Random Forest and XGBOOST we noticed that the F1 score is 0.83 as in figure 35 and the ROC curve as in figure 36.



```
              precision    recall  f1-score   support

           0       0.89      0.74      0.81      7567
           1       0.78      0.91      0.84      7736

    accuracy                           0.83     15303
   macro avg       0.84      0.83      0.83     15303
weighted avg       0.84      0.83      0.83     15303
```

Figure 35: Classification report for CNN – Approach 2

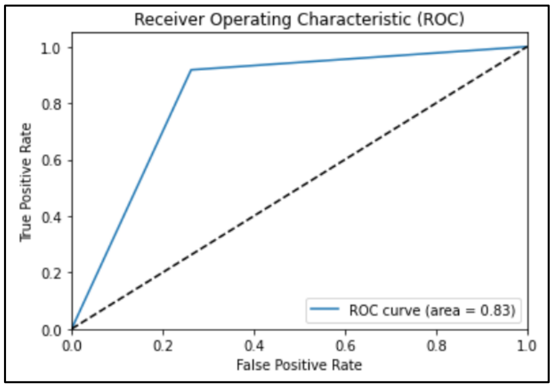


Figure 36: ROC curve

We can see here that we were not able to improve our accuracy and F1 score for the top features. We achieved a much better F1 score for CNN with all the features. This concluded that all features were equally important for the prediction of the outcome.

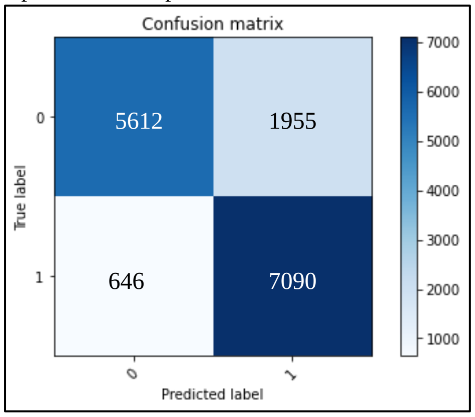
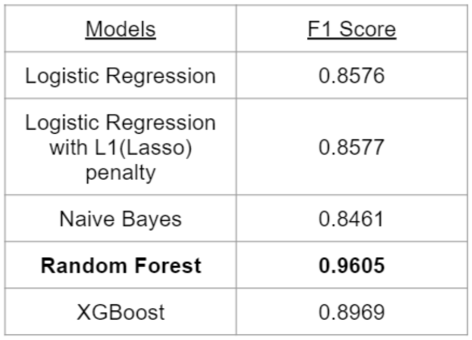


Figure 37: Confusion Matrix for CNN - Approach 2

7090 applications were correctly predicted as certified, while 1955 applications were incorrectly predicted as certified. 5612 applications were correctly predicted as denied, while 646 applications were incorrectly predicted as denied. When we use the top features, we get the same values for prediction as given in Figure 37 and Figure 31.

4.3 Results Comparison

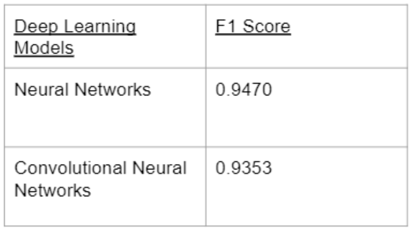


Models	F1 Score
Logistic Regression	0.8576
Logistic Regression with L1(Lasso) penalty	0.8577
Naive Bayes	0.8461
Random Forest	**0.9605**
XGBoost	0.8969

Figure 38: Comparison of our Machine learning models

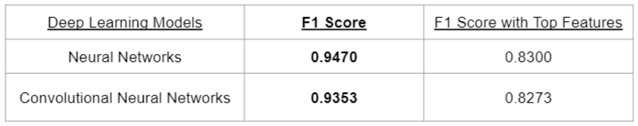
We got the best F1 score from the Random Forest model with score of 0.96, while Naïve Bayes performed the least with score of 0.84.

As an additional feature implementation, we performed Deep Learning models, Neural Networks and Convolution Neural Networks. The results show that Neural networks performed better than CNN.



Deep Learning Models	F1 Score
Neural Networks	0.9470
Convolutional Neural Networks	0.9353

Figure 39: Comparing our Deep Learning models



Deep Learning Models	F1 Score	F1 Score with Top Features
Neural Networks	**0.9470**	0.8300
Convolutional Neural Networks	**0.9353**	0.8273

Figure 40: Comparing Deep Learning models after selecting top features

When we compared the results after selecting the top features, we were not able to improve our score. This concluded that our all the features were equally important for the prediction.

RELATED WORK

We were able to find 3 reports that were relevant to our topic and used the same dataset as we had from Kaggle. First report [9] by Madhana Sohan Kumar conducted a analyzes of the dataset and performed Naïve Bayes, Random Forest and XGBoost as their models. Their analysis gave us the insight for the data to be used in the classification models.

Second report [10] performed the data on Logistic Regression and Random Forest Classifier using the multiclass Classification. They performed the cases analysis strategies that were useful, and we tried to implement them ourselves to improve the overall performance of the models.

The students from Stanford also conducted a project [11] to predict the outcome by using Naive Bayes, Logistic Regression, SVM and Neural Network as their models. The way they trained neural networks helped us on training the data with the deep learning models.

CONCLUSION

A wide spectrum of learning models is available to predict H1B visa status. The three models applied to find the status and their performance in the paper are Naive Bayes, Random Forest and XGBoost [12]. Our method for Random Forest, Neural Network and CNN gave much better results for predicting the H1B status. While all the three models gave good results. We found out that Random Forest was the best implementation.

WORK DIVISION

Name: **Prerak Shah**
Tasks performed:

- Removed rows with null values and removed duplicate rows.
- Performed feature engineering.
- Performed feature selection methods and ran on deep learning models.
- CNN model implementation.
- Performed Logistic Regression with L1 penalty.
- Worked on the Report and Presentation.

Name: **Sai Sri Meghana Dharmapuri**
Tasks performed:

- Performed feature normalization and encoding.
- Split data into train and test data.
- Outperformed the results of the 3 models in the paper.
- Neural Network model implementation.
- Performed Logistic Regression.
- Worked on the Report and Presentation.

LEARNING EXPERIENCE

- Handling imbalanced data was a tough task. The imbalanced data could produce result with inclination towards the majority class. We used Downsampling of data to handle the imbalanced data.
- Categorizing feature columns was useful. We used feature engineering techniques.

REFERENCES

[1] U.S. Citizenship and Immigration Services, “H-1B Specialty Occupations, DOD Cooperative Research and Development Project Workers and Fashion Models,” updated March 2016, https://www.uscis.gov/working-united-states/temporary-workers/h-1b-specialty-occupations-dodcooperative-research-and-development-project-workers-and-fashion-models.

[2] Dombe, Anay, Rahul Rewale, and Debabrata Swain. "A Deep Learning-Based Approach for Predicting the Outcome of H-1B Visa Application." Machine Learning and Information Processing. Springer, Singapore, 2020. 193-202.

[3] Gunel, Beliz, and Onur Cezmi Mutlu. "Predicting the Outcome of H-1B Visa Applications."

[4] Swain, Debabrata, et al. "Prediction of H1B Visa Using Machine Learning Algorithms." 2018 International Conference on Advanced Computation and Telecommunication (ICACAT). IEEE, 2018.

[5] https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d

[6] https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[7] https://www.displayr.com/what-is-a-roc-curve-how-to-interpret-it/

[8] Ng, Andrew Y. "Feature selection, L 1 vs. L 2 regularization, and rotational invariance." Proceedings of the twenty-first international conference on Machine learning. 2004.

[9] http://www.ijirset.com/upload/2018/october/37_A%20Predictive.pdf

[10] D. Swain, K. Chakraborty, A. Dombe, A. Ashture and N. Valakunde, "Prediction of H1B Visa Using Machine Learning Algorithms," 2018 International Conference on Advanced Computation and Telecommunication (ICACAT), Bhopal, India, 2018.

[11] http://cs229.stanford.edu/proj2017/final-reports/5208701.pdf

[12] Madhana Sohan Kumar1, N Naresh2 “A Predictive Model for H1-B Visa Petition Approval” 2018 International Journal of Innovative Research in Science, Engineering and Technology. 2018

[13] https://www.americanimmigrationcouncil.org/research/h1b-visa-program-fact-sheet