Data Mining

Team- STAR

(Final Report.)

Visa Approval Prediction

Authors:

Sridevi Jaidi
Graduate student
Tagliatela College of Engineering,
University of New Haven, West Haven,
sjaid1@unh.newhaven.edu

Shyam sunder Reddy Beeram
Graduate student
Tagliatela College of Engineering,
University of New Haven, West Haven,
sbeer4@unh.newhaven.edu

Priyanka Nandigam
Graduate student
Tagliatela College of Engineering,
University of New Haven, West Haven,
pnand3@unh.newhaven.edu

Abstract

Over 2 million visa petitions are filed by the employers each year and only 65000 petitions are approved. So, the goal is to explore the petitions filed and their outcomes for the past six years i.e., from 2011 to 2016, and to find a pattern to predict the outcome by using a predictive model developed using Machine Learning techniques.

In order to predict the case status of the applicants, we will be feeding the model with the dataset which contains the required fields by which the machine can classify the case status as certified or denied.

Content:

1. Introduction.
2. Related work.
3. Proposed Method.
4. The experimental Results.
5. Discussions.
6. Conclusion and Future works.
7. Appendix for link (GitHub).
8. References.

Introduction

Over 2 million visa petitions are filed by the employers each year and only 65000 petitions are approved. So, the goal is to explore the petitions filed and their outcomes for the past six years i.e., from 2011 to 2016, and to find a pattern to predict the outcome by using a predictive model developed using Machine Learning techniques.

H-1B visa applications that are filed by many professional foreign nationals every year. Here, we framed the problem as a classification problem and applied it in order to output a predicted case status of the application. The input to our algorithm is the attributes of the applicant. H-1B is a type of non-immigrant visa in the United States that allows foreign nationals to work in occupations that require specialized knowledge and a bachelor's degree or higher in the specific specialty.

This visa requires the applicant to have a job offer from an employer in the US before they can file an application to the US immigration service (USCIS). We believe that this prediction algorithm could be a useful resource both for the future H-1B visa applicants and the employers who are considering sponsoring them.

The Aim of the project is that Can we predict visa status of the applicants, by feeding the model with the dataset which contains the required fields by which the machine can classify the visa status as certified or denied.

Dataset Description

Attributes:

First, we describe the key elements of the data. The data set includes 40 columns in each year's records and the column names completely changed after 2015. My first step was to rename the columns in older records for the relevant columns to match with the newer records. The relevant columns include:

1. EMPLOYER_NAME: Name of employer submitting the H1-B application. Used in comparing salaries and number of applications of various employers.

2. JOB_TITLE: Title of the job using which we can filter specific job positions for e.g., Data Scientist, Data Engineer etc.

3. PREVAILING_WAGE: The prevailing wage for a job position is defined as the average wage paid to similarly employed workers in the requested occupation in the area of intended employment. The prevailing wage is based on the employer's minimum requirements for the position. (Source). This column will be one of the key metrics of the data analysis.

4. WORKSITE_CITY, WORKSITE_STATE: The foreign worker's intended area of employment. We will explore the relationship between prevailing wage for Data Scientist position across different locations.

5. CASE_STATUS: Status associated with the last significant event or decision. Valid values include "Certified," "Certified Withdrawn," Denied," and "Withdrawn". This feature will help us analyze what share of the H-1B visa is taken by different employers/ job positions.

Related work.

Review:1

- **Title**: Prediction of H1B Visa Using Machine Learning Algorithms
- **AuthorNames**: Debabrata Swain Kushankur Chakraborty; Anay Dombe; Ashitosh Ashture; Nandakishor Valakunde Debabrata Swain Kushankur Chakraborty; Anay Dombe; Ashitosh Ashture; Nandakishor Valakunde
- **Affiliation**: Vishwakarma Institute of Technology, Pune [2018 International Conference on Advanced Computation and Telecommunication (ICACAT)]
- **Publication date**: 19 December 2019
- **Name of publisher**: IEEE

Review:2

- **Title**: A Hybrid Machine Learning Model Approach to H-1B Visa
- **Author names**: A. Singh Chadha and A. Shitole,
- **Affiliation**: International Institute of Information Technology (I2IT), Pune, India. [2021 3rd International Conference on Electrical, Control and Instrumentation Engineering (ICECIE)]
- **Publication** date: 07 January 2022
- **Name of publisher**: IEEE

Review 3:

- **Title**: An allotment of H1B work visa in USA using machine learning
- **Author names**: Pooja Thakur, Mandeep Singh , Harpreet Singh , Prashant Singh Rana
- **Affiliation**: Chandigarh University, Mohali, India
- **Publication date**: August 2018
- **Name of publisher**: Google scholar, [Science Publishing corporation]

Review 4:

- **Title**: Predicting the outcome of H-1B visa using ANN algorithm
- **Author names**: Raghav Khaterpal, Harit Ahuja, Jatin Goel, Karanveer Singh, Rahul Manoj
- **Affiliation**: SRM Institute of Science and Technology
- **Publication date:** May 2020
- **Name of publisher**: International Journal of Recent Technology and Engineering (IJRTE)

Review 5:

- **Title**: H1B VISA APPROVAL USING MACHINE LEARNING ALGORITHM
- **Author names**: Mrs. A. Durga Bhavani, Guddeti Bharath, Dubbaka Tharun Reddy
- **Affiliation:** Anurag University, Telangana, India
- **Publication date**: April 2022
- **Name of publisher**: Journal of Emerging Technologies and Innovative Research

Performance Metrics:

Performance metrics using different data mining algorithms based on accuracy of various researchers:

- Logistic Regression: 89%
- Bagging Classifier: 89%
- Gradient Descent Classifier: 86%
- Naive Bayes:77%
- Random Forest Classifier: 95%

Proposed Method:

(Project Flow)

After downloading the dataset, we follow the following method:
Preprocess or clean the data: The data is preprocessed, normalized, and ready for training.

Analyze the pre-processed data: Handling the null values.Handling the categorical values if any. Normalize the data if required. Identifying the dependent and independent variables. Split the dataset into train and test sets.

Train the machine with preprocessed data using an appropriate machine learning algorithm: We will be initially considering the Random Forest Classifier model and fit the data. By using Random Forest Classifier we are fitting our model in order to obtain the prediction. Then we predict the model.

Save the model and its dependencies. Now, we check for the accuracy and lastly we dump the file into pickle and save our model.

Pickle- This module is used for serializing and de-serializing a Python object structure. Any object in Python can be pickled so that it can be saved on disk.

Build a Web application using flask that integrates with the model built: Building an application to integrate the model After the model is built, we will be integrating it to a web application so that normal users can also use it to know if any website is phishing or safe in a no-code manner. In the application, the user provides some parameters and that result in displaying the case status certified or denied.

Experimental Results:

The algorithms can be chosen according to the objective. As the dataset which we are using is a **Classification dataset** we can use the following algorithms,

- Logistic Regression
- Random Forest Regression / Classification
- Decision Tree Regression / Classification
- K-Nearest Neighbors
- Support Vector Machine

Performance Metrics:

Performance metrics using different data mining algorithms based on accuracy of various researchers:

- Logistic Regression: 85%
- K nearest neighbors: 83%
- Support vector Machine: 82%
- Decision tree:77%
- Random Forest Classifier: 89%

	precision	recall	f1-score	support
0	0.88	0.99	0.93	784458
1	0.48	0.09	0.16	60711
2	0.24	0.04	0.07	27545
3	0.15	0.01	0.02	27253
4	0.00	0.00	0.00	6
6	0.00	0.00	0.00	1
accuracy			0.87	899974
macro avg	0.29	0.19	0.20	899974
weighted avg	0.81	0.87	0.82	899974

The list of exploration techniques (statistical and visualization techniques) used in this project are:

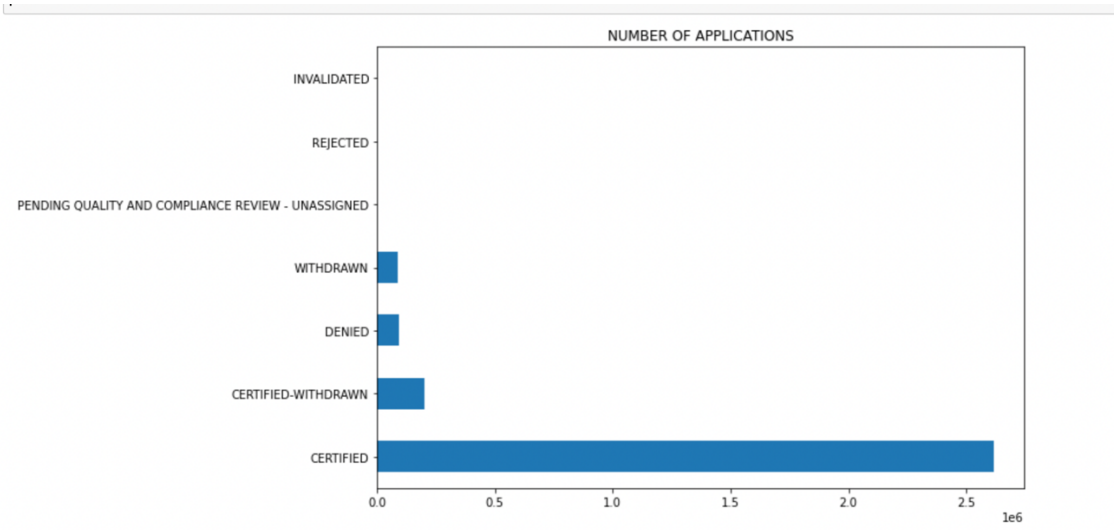
- Bar graph
- Histogram
- White grid graph
- Drawing plot
- Co-relation & Heatmap

After displaying five rows in the dataset, we understand how the dataset and its attributes. And also, we print information about data frame including the index type and columns non null values and memory usage. We also code to get to get a Series containing counts of unique values.

Then after cleaning the data, we start visualizing data in different perspectives.

Bar graph(i):

This is for the analysis of the case status of the applicants.

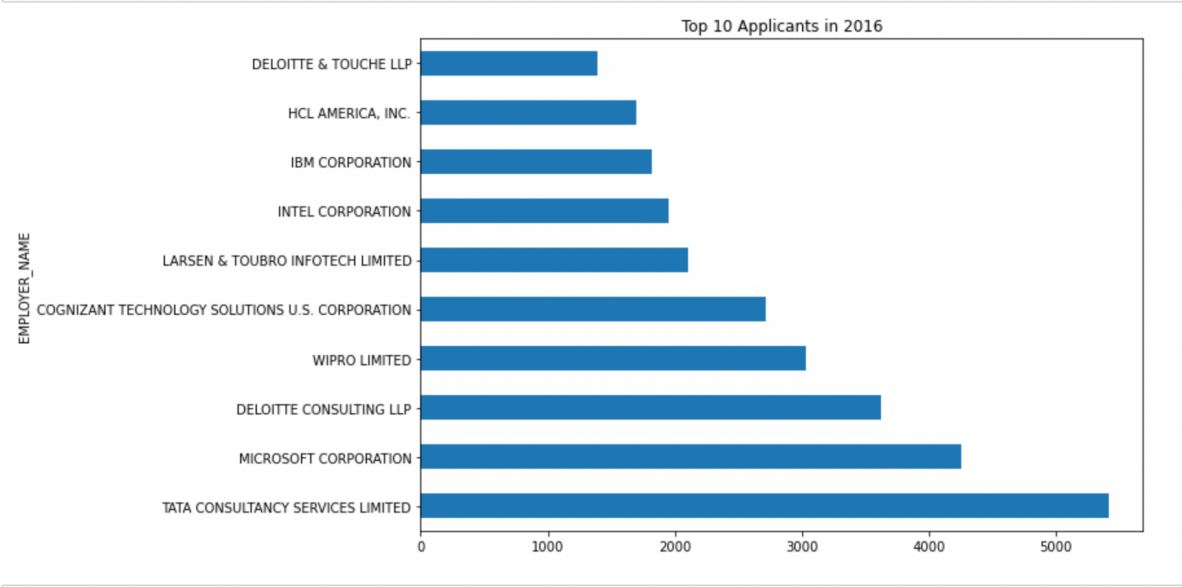


Number of Applications per year

The graph which we plotted here is a bar graph to know the case status of the applicants per year.

Bar graph (ii)

This is for the analysis of applicants who belong to top 10 companies in 2016.



The graph which we plotted here is a bar graph to know the employer who has highest no of applicants in 2016.

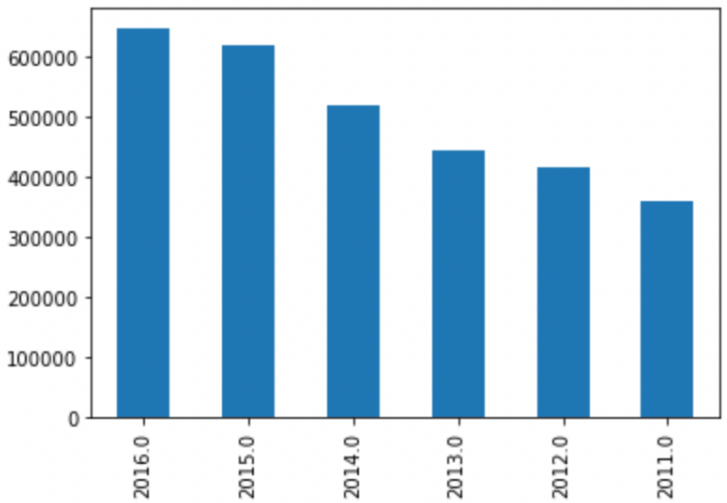
In the project there were few other bar graphs to know the status and employer who has good number of applicants.

Histogram:

This is for the analysis of no of applicants per year.

```
: df.YEAR.value_counts().plot(kind = 'bar')
```

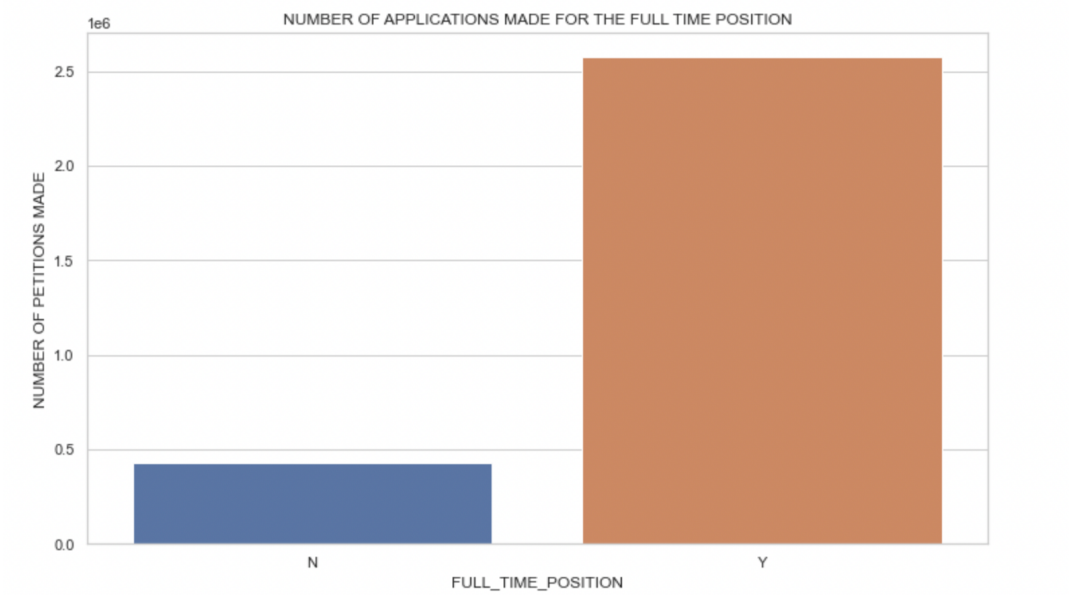
```
: <AxesSubplot:>
```



The histogram shows the data of no of applicants filed visa in that respective year.

White grid graph:

This is for the analysis of Number of applicants made for the full time position.

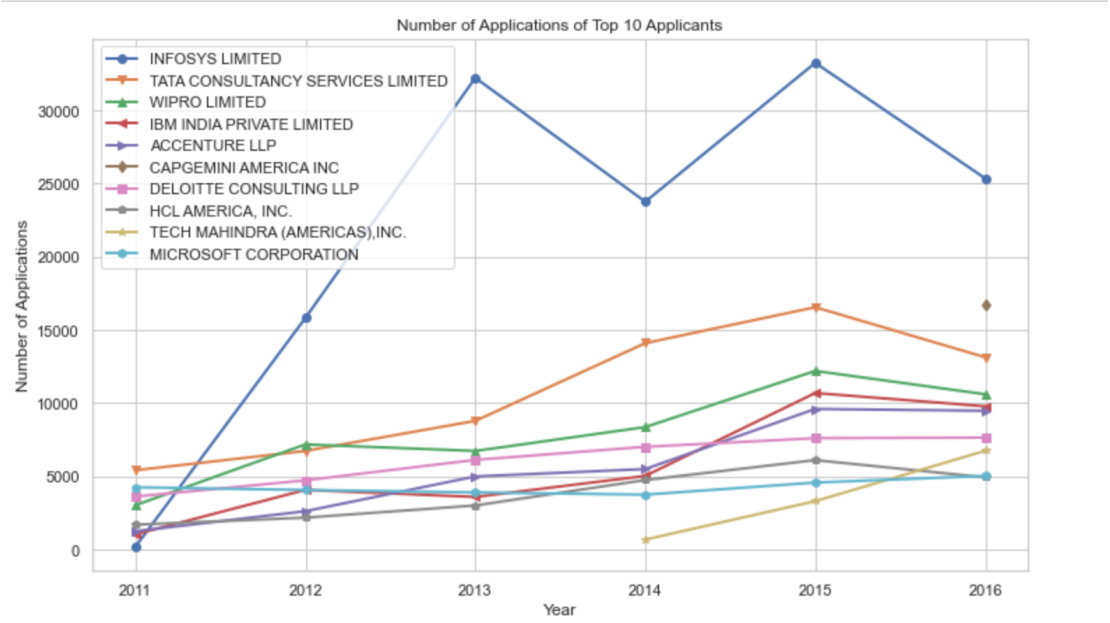


g = sns.countplot(x = 'FULL_TIME_POSITION', data = df)

Here, by using sns we are counting the full time positions and here X-axis is denoted for full time positions.

Drawing Plot:

This is for the analysis of number of applicants of the top 10 companies.



Here the graph tells about the total number of applications of the top 10 companies.
The plot of different color is for the respective company showed in menu graph.

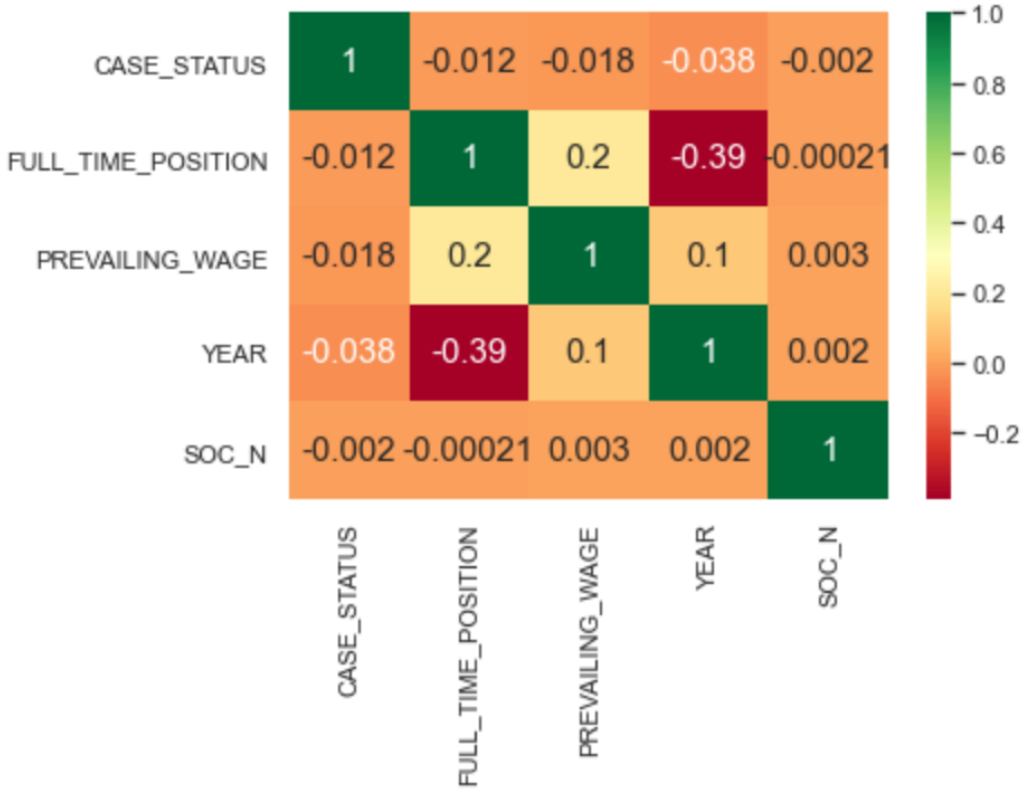
Co-relation & Heatmap:

Correlation is used for measuring the strength and direction of the linear relationship between two continuous random variables x and y. A positive correlation means the variables increase or decrease together. A negative correlation means if one variable increase then the other decrease.

Correlation values can be computed using the 'corr()' method of the Data Frame and rendered using heatmap.

According to our prediction we can drop columns which are not required or preferred.

```
<AxesSubplot:>
```



Discussion

After we build model, our discussion includes how to optimize that and build & integrating with front end, such that a normal user can test the model.

Parameters:

Model Parameters: These are the parameters in the model that must be determined using the training data set. These are the fitted parameters.

Hyperparameters: These are adjustable parameters that must be tuned in order to obtain a model with optimal performance.

The parameters in our project were:

```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, random_state = 42)
```

```
In [30]: rf.get_params()

Out[30]: {'bootstrap': True,
          'ccp_alpha': 0.0,
          'class_weight': None,
          'criterion': 'gini',
          'max_depth': 4,
          'max_features': 4,
          'max_leaf_nodes': 20,
          'max_samples': None,
          'min_impurity_decrease': 0.0,
          'min_impurity_split': None,
          'min_samples_leaf': 1,
          'min_samples_split': 2,
          'min_weight_fraction_leaf': 0.0,
          'n_estimators': 100,
          'n_jobs': None,
          'oob_score': False,
          'random_state': 123,
          'verbose': 0,
          'warm_start': False}
```

Techniques used for Optimizing parameters:

- **Stochastic Gradient Descent**: **Stochastic gradient descent** is an optimization algorithm often used in machine learning applications to find the model parameters that correspond to the best fit between predicted and actual outputs. It's an inexact but powerful technique.

- **Random search:** Random search (RS) is a family of numerical optimization methods that do not require the gradient of the problem to be optimized, and RS can hence be used on functions that are not continuous or differentiable.

Optimization techniques Enhance Data Mining Techniques:

The algorithms can be chosen according to the objective. As the dataset which we are using is a **Classification dataset** we can use the following algorithms,

- Logistic Regression
- Random Forest Regression / Classification
- Decision Tree Regression / Classification
- K-Nearest Neighbors
- Support Vector Machine

Optimization techniques effecting Performance Metrics:

Performance metrics using different data mining algorithms based on accuracy of various researchers:

As we are using Random Forest classifier, the optimizing techniques were stochastic gradient descent and random search.

```
In [86]: s.get_params()

Out[86]: {'alpha': 0.01,
          'average': False,
          'class_weight': None,
          'early_stopping': False,
          'epsilon': 0.1,
          'eta0': 0.0,
          'fit_intercept': True,
          'l1_ratio': 0.1,
          'learning_rate': 'optimal',
          'loss': 'hinge',
          'max_iter': 1000,
          'n_iter_no_change': 5,
          'n_jobs': None,
          'penalty': 'l2',
          'power_t': 0.5,
          'random_state': 123,
          'shuffle': True,
          'tol': 0.001,
          'validation_fraction': 0.1,
          'verbose': 0,
          'warm_start': False}

In [45]: print(accuracy_score(y_test,pred1))

         0.8690851069030883
```

Random search:

```
In [80]: rf.get_params()

Out[80]: {'bootstrap': True,
          'ccp_alpha': 0.0,
          'class_weight': None,
          'criterion': 'gini',
          'max_depth': 4,
          'max_features': 4,
          'max_leaf_nodes': 20,
          'max_samples': None,
          'min_impurity_decrease': 0.0,
          'min_impurity_split': None,
          'min_samples_leaf': 1,
          'min_samples_split': 2,
          'min_weight_fraction_leaf': 0.0,
          'n_estimators': 100,
          'n_jobs': None,
          'oob_score': False,
          'random_state': 123,
          'verbose': 0,
          'warm_start': False}
```

```
In [85]: gs.best_score_

Out[85]: 0.8719657437555712
```

Accuracy using the optimizing technique random search is 87 and using stochastic gradient descent is 86.

Application Building:

Building an Application to integrate the model
After the model is built, we will be integrating it to a web application so that normal users can also use the model.

In the application, the user provides the corresponding parameter values and the case status will be predicted.

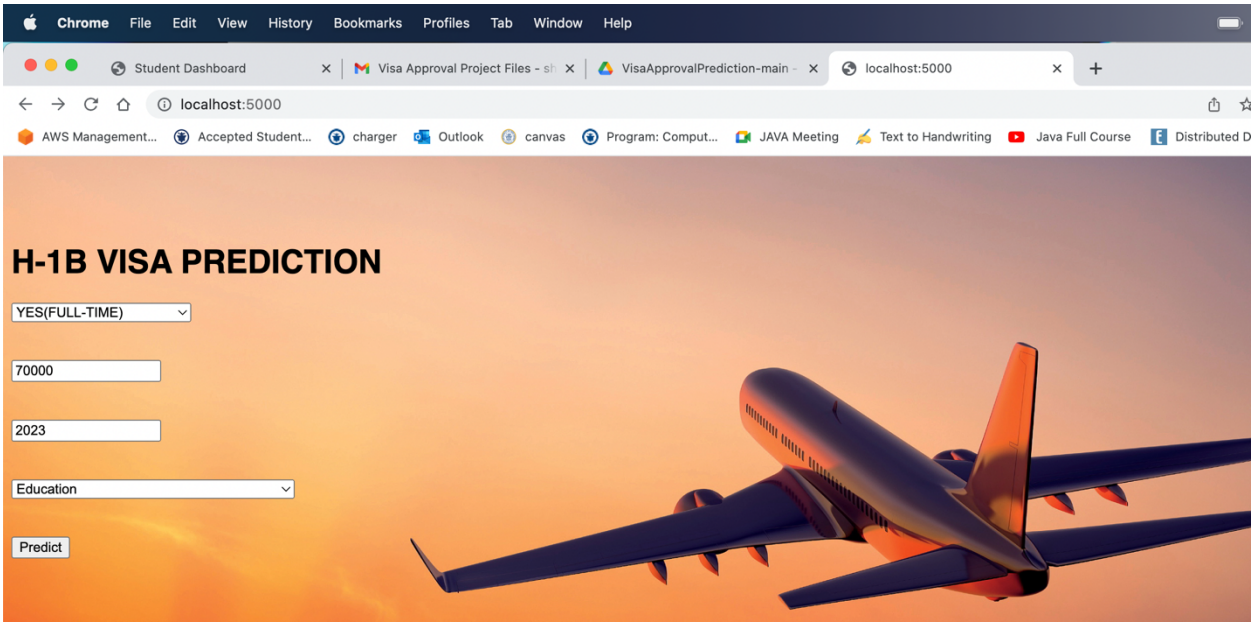
Build The Python Flask App:

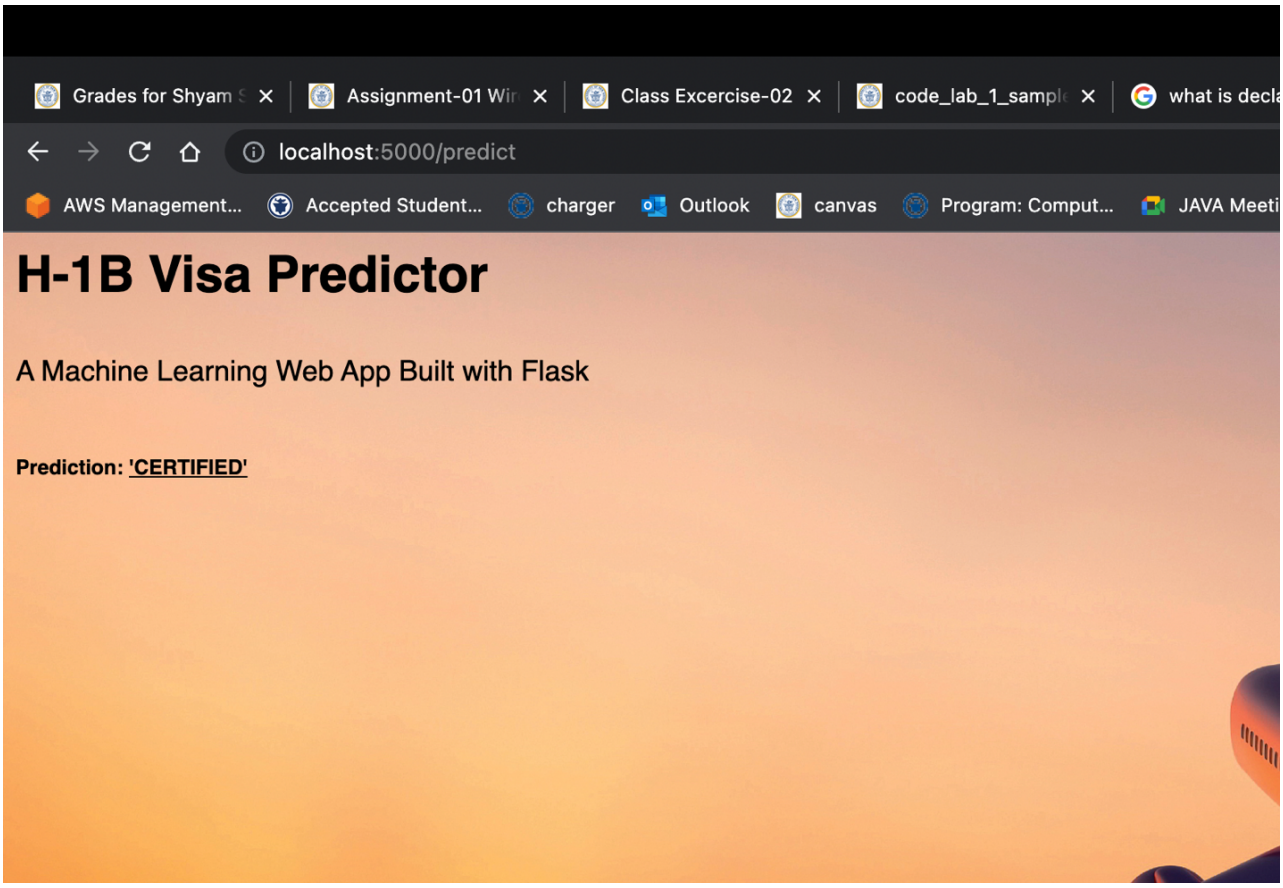
In the flask application, the input parameters are taken from the HTML page These factors are then given to the model to know if the Visa is approved or not and is sent back to the HTML page to notify the user.

Results:

When you run the model using your terminal,

- It will show the local host where your app is running on **http://127.0.0.1.5000/**
- Copy that local host URL and open that URL in the browser. It does navigate me to where you can view your web page.
- Enter the values, click on the predict button and see the result/prediction on the web page.





Conclusion and future work:

To conclude, based on our research we found accuracy is more for Random Forest classifier. We also got to know that Random Forest Classifier predicts better in this case of Prediction.

However by doing this research we are able to understand the problem to classify if it is a regression or a classification kind of problem, and also able to know how to pre-process/clean the data using different data pre-processing techniques. Applying different algorithms according to the dataset. Now will be able to know how to find the accuracy of the model. We will be able to build web applications using the Flask framework.

We believe that this prediction algorithm could be a useful resource both for the future H-1B visa applicants and the employers who are considering sponsoring them.

Appendix:

GitHub repository :

https://github.com/shyambeeram/STAR

References:

Dataset: https://www.kaggle.com/datasets/nsharan/h-1b-visa

1. 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), 2018, pp. 1-7, doi: 10.1109/ICACAT.2018.8933628.

[https://ieeexplore-ieee-org.unh-proxy01.newhaven.edu/document/8933628]

2. A. Singh Chadha and A. Shitole, "A Hybrid Machine Learning Model Approach to H- 1B Visa," 2021 3rd International Conference on Electrical, Control and Instrumentation Engineering (ICECIE), 2021, pp. 1-8, doi: 10.1109/ICECIE52348.2021.9664747. [.https://ieeexplore-ieee-org.unh-proxy01.newhaven.edu/document/9664747]

3. https://www.datacamp.com/tutorial/predicting-H-1B-visa-status-python

4. https://www.kaggle.com/code/akhilkasare/h-1b-visa-prediction-using-machine- learning

5. **Machine learning packages:**

https://www.activestate.com/blog/top-10-python-machine-learning-packages/

core:

https://www.geeksforgeeks.org/supervised-unsupervised-learning/