

# **PREDICTIVE MODELING FOR H1B VISA APPROVAL USING IBM WATSON**

## **1. INTRODUCTION**

### **1.1 OVERVIEW**

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 the Guided Project, our goal is to predict the outcome of 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.

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.

### **1.2 PURPOSE**

We'll be able to understand the problem to classify if it is a regression or a classification kind of problem. We will be able to know how to pre-process/clean the data using different data pre-processing techniques. We will be able to analyze or get insights into data through visualization. Applying different algorithms according to the dataset. We 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. The generated model should be able

to accurately classify the H1B application. The model shall allow the user to perform a preliminary analysis of their application based on H1B application decisions made in the recent past.

## **2. LITERATURE SURVEY**

### **2.1 EXISTING PROBLEM**

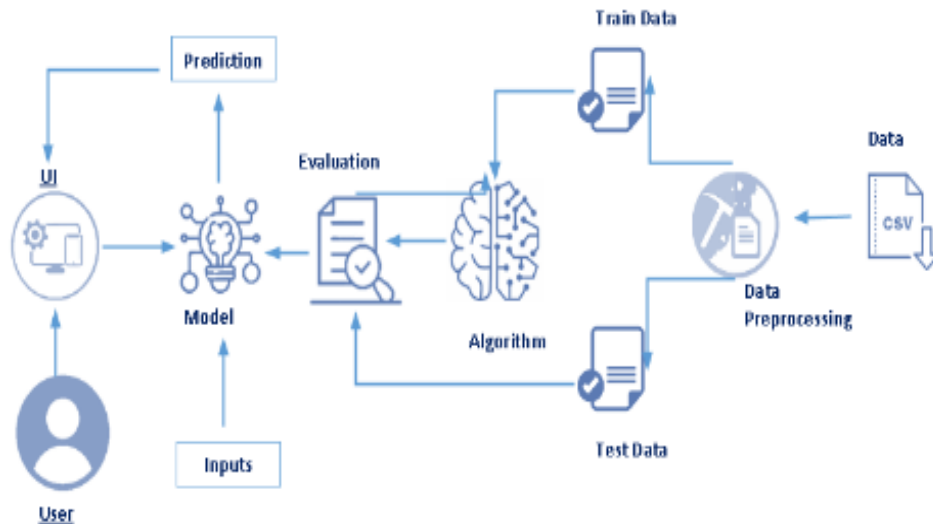
The H-1B visa applications that are filed by many professional foreign nationals every year. Given complex procedures and paperwork that are required, most companies delegate H-1B applications to an immigration attorney. Each application needs to adhere to a proper format and satisfy conditions required for each job category in order to be successfully certified. The whole process is usually achieved by law professionals communicating with the applicant and the employer multiple times. A denied application not only incurs additional cost for each individuals' time, but could also potentially lower the applicant's chance to be successfully certified for the same position. Hence, it is imperative that each filed application satisfies requirements specified by USCIS and is comparable to other applications from the same industry. For my capstone project, I created a supervised classifier models (Random Forest) evaluated performance of each of them, and decided the model based on performance and feasibility. I started by figuring out the set of features that are common between data source from different years, resolving naming conflict, and discarding infeasible and least useful features. 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.

### **2.2 PROPOSED SOLUTION**

Our aim is to predict the outcome of H-1B visa applications that are filed by many high-skilled foreign nationals every year. Explore and understand the dataset using a suite of line plots for the series data and histogram for the data distributions. We framed the problem as a classification problem and applied Random Forest Classifier in order to output a predicted case status of the application.

### 3. THEORITICAL ANALYSIS

#### 3.1BLOCK DIAGRAM



#### 3.2HARDWARE/SOFTWARE DESIGNING

##### SOFTWARE DESIGNING:

1. Jupyter Notebook Environment
2. Spyder
3. Machine Learning Algorithms
4. Python (pandas, numpy, matplotlib, seaborn, sklearn)
5. HTML
6. Flask

We developed this Visa Approval status prediction by using the Python language which is a interpreted and high level programming language and using the Machine Learning algorithms. for coding we used the Jupyter Notebook environment of the Anaconda distributions and the Spyder, it is an integrated scientific programming in the python language. For creating an user interface for the prediction we used the Flask. It is a micro web framework written in Python. It is classified as a micro frame work because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or

any other components where pre-existing third-party libraries provide common functions, and a scripting language to create a webpage is HTML by creating the templates to use in the functions of the Flask and HTML.

## **HARDWARE REQUIRMENTS**

- Processor : Intel Core i3
- Hard Disk Space : Min 100 GB
- Ram : 4 GB
- Display : 14.1 “Color Monitor(LCD, CRT or LED)

## **4. EXPERIMENTAL INVESTIGATIONS**

H-1B visas are a category of employment-based, non-immigrant visas for temporary foreign workers in the United States. For a foreign national to apply for H1-B visa, a US employer must offer them a job and submit a petition for a H-1B visa to the US immigration department. This is also the most common visa status applied for and held by international students once they complete college or higher education and begin working in a full-time position. This dataset contains five year's worth of H-1B petition data, with approximately 3 million records overall. The columns in the dataset include case status, employer name, worksite coordinates, job title, prevailing wage, occupation code, and year filed.

EMPLOYER\_NAME: Name of employer submitting application.

SOC\_NAME: Occupational name associated with the SOC CODE which is an occupational code associated

with the job being requested for temporary labour condition, as classified by the Standard Occupational Classification (SOC) System.

JOB\_TITLE: Title of the job

FULL\_TIME\_POSTION: There are 2 categories for this feature: Y= Full time position and N = Part Time Position

PREWAILING WAGE: the average wage paid to employees with similar qualifications in the intended area of employment.

YEAR: The year of filing the petition

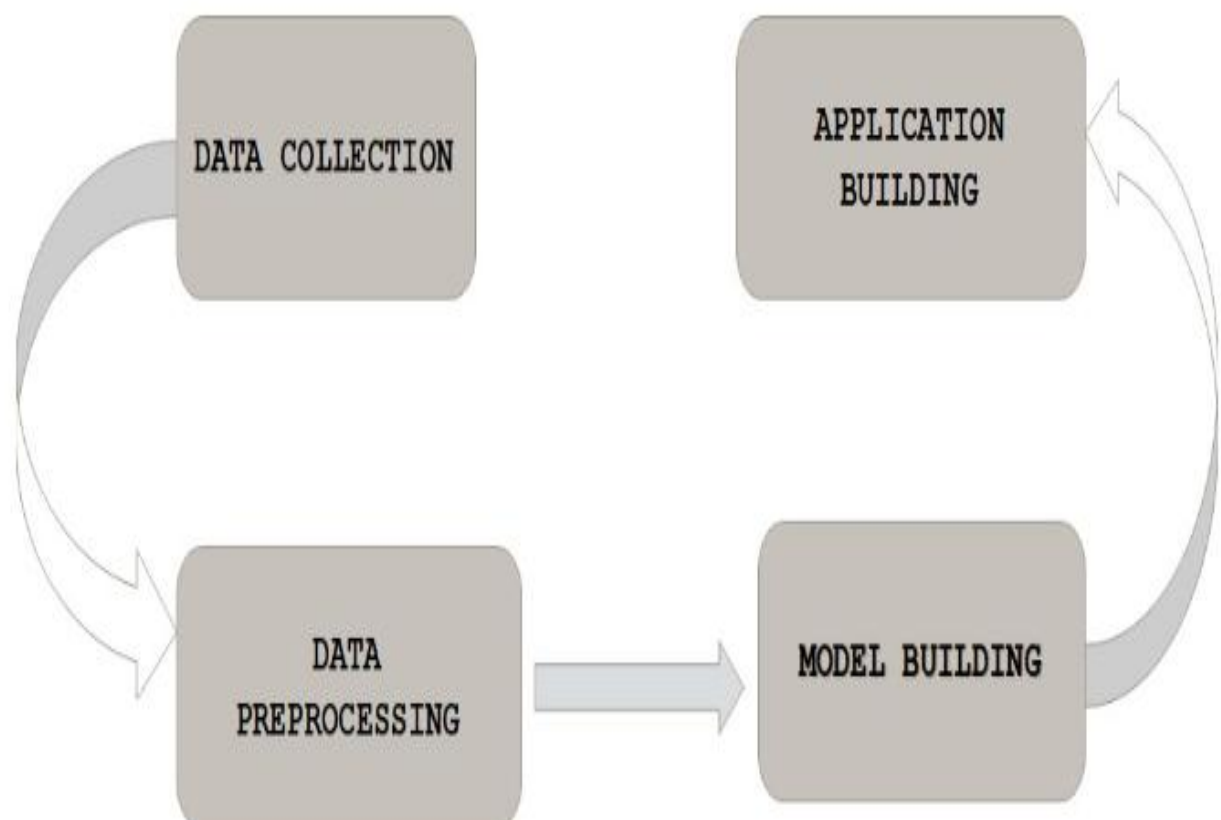
**WORKSITE:** City and state of the applicant's job.

**LONGITUDE & LATITUDE:** Exact geographical location of the worksite.

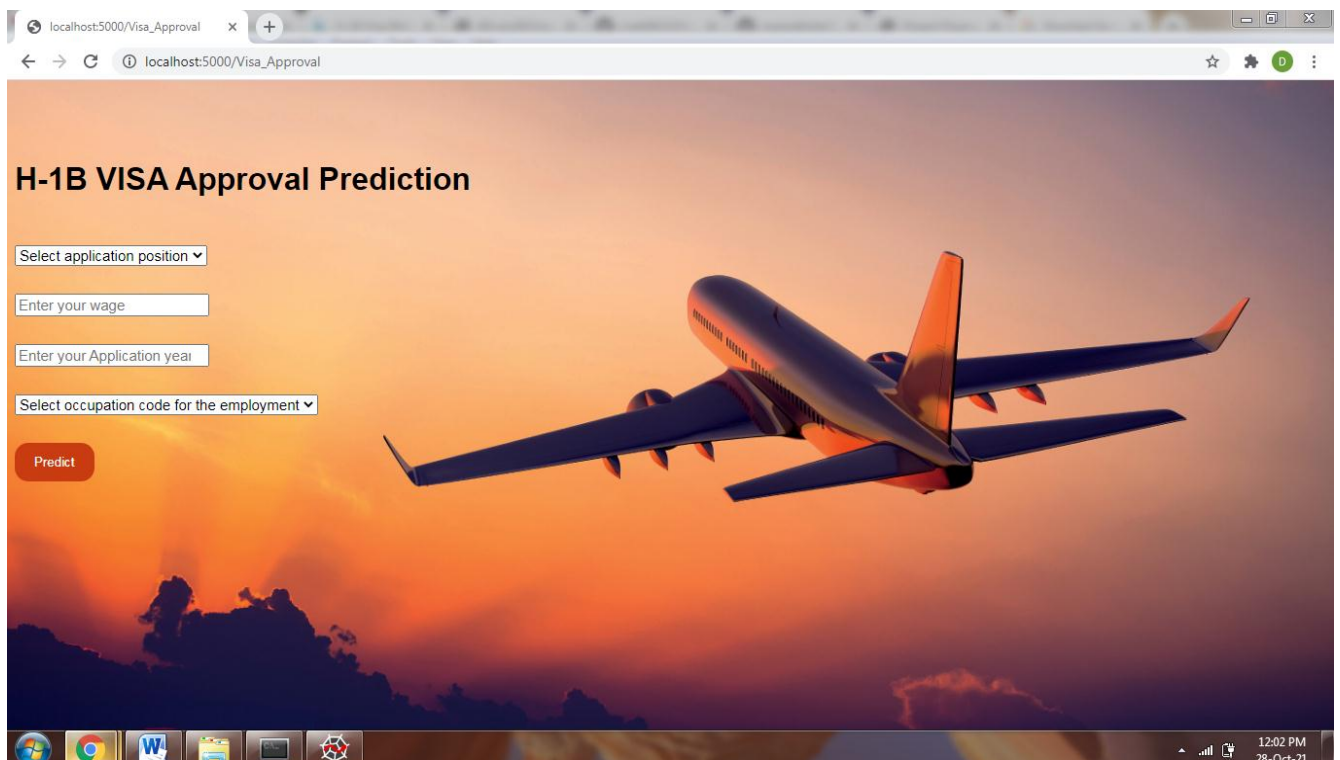
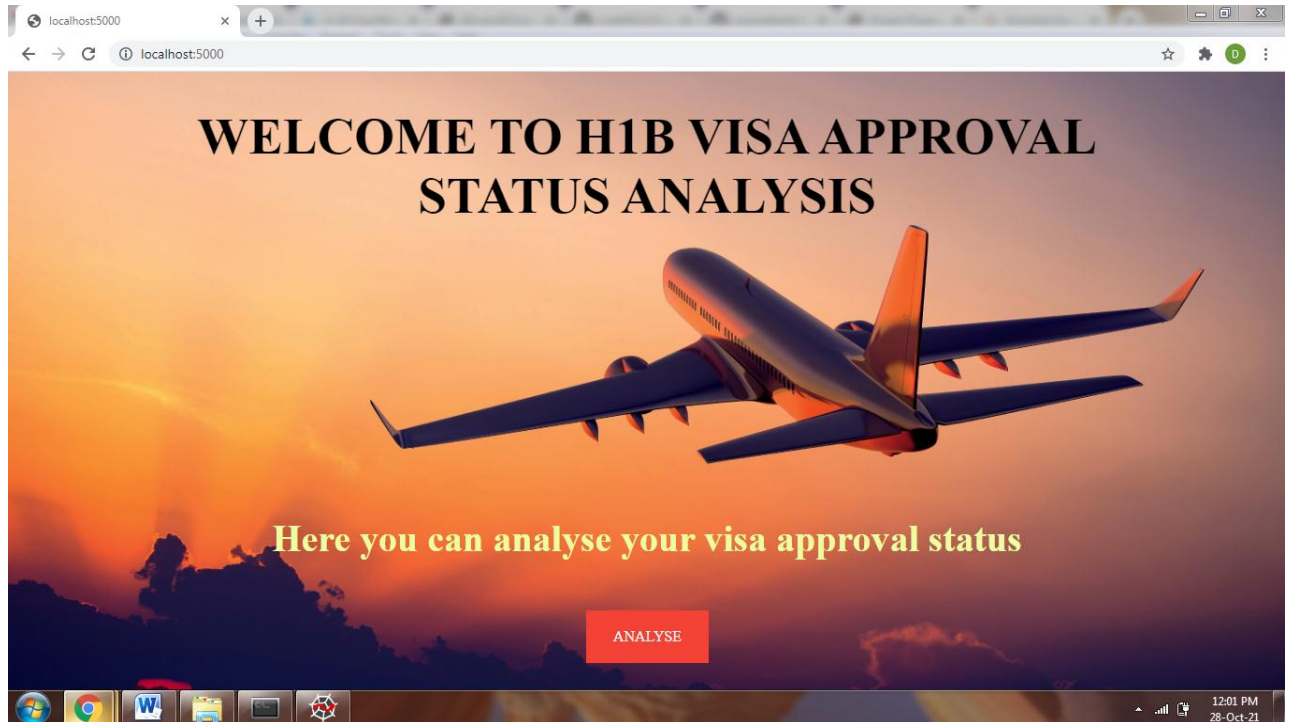
The 1 label in the dataset is divided into 7 classes: (1) CERTIFIED (2) CERTIFIED-WITHDRAWN (3) DENIED (4)WITHDRAWN (5) PENDING QUALITY AND COMPLIANCE REVIEW – UNASSIGNED(6) REJECTED (7)INVALIDATED.

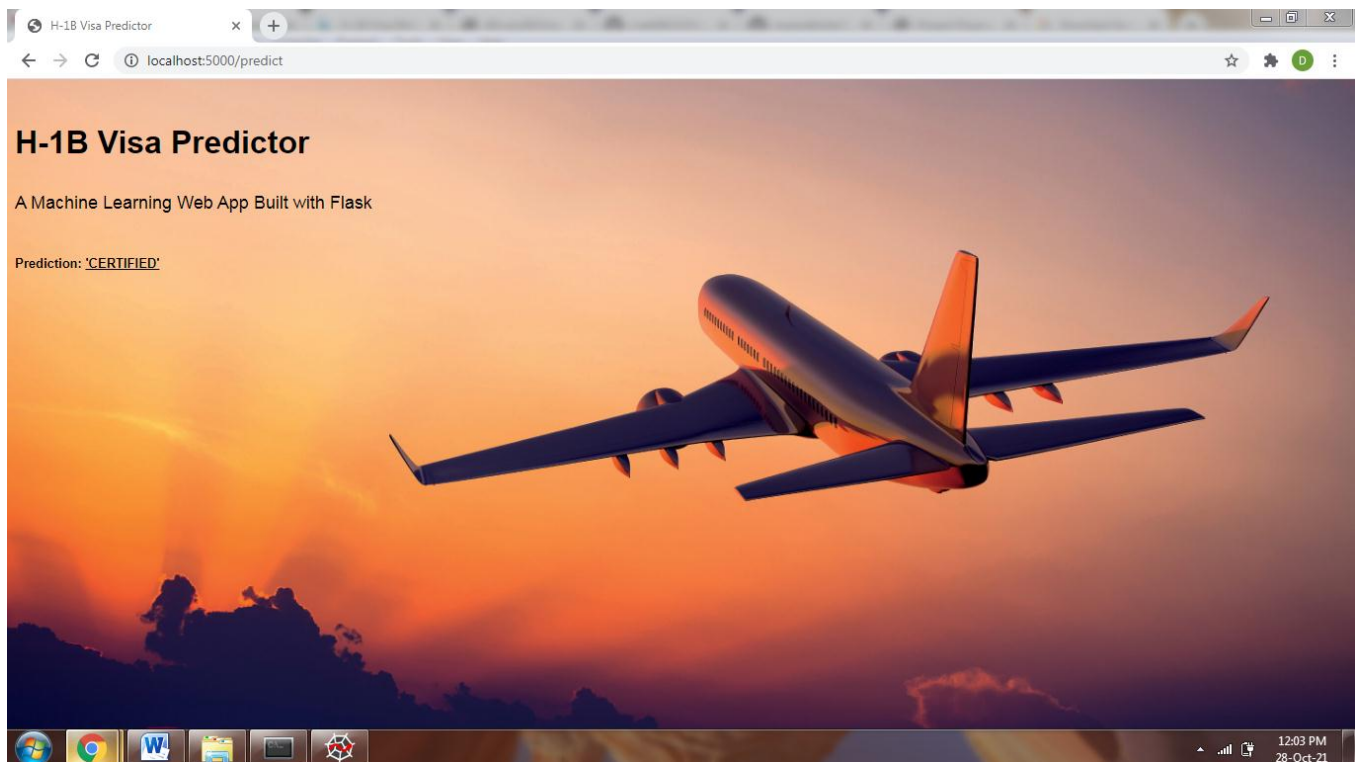
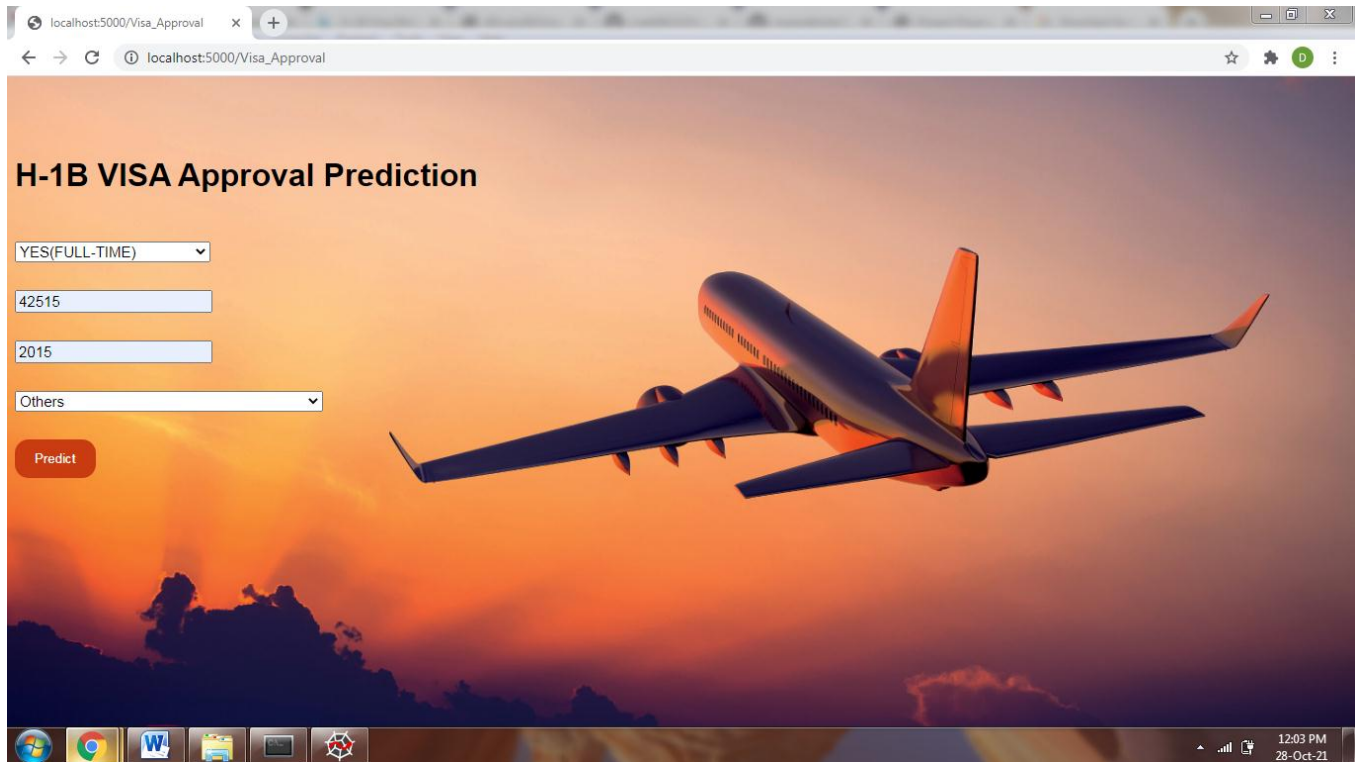
Then pre-process/clean the data using different data pre-processing techniques. We will be able to analyze or get insights into data through visualization. Applying different algorithms according to the dataset. Supervised classifier model (Random Forest Classifier Algorithm) evaluated the performance. site. To use the classifier model approach to predict the H1b visa approval at a new site, several steps are required.

## 5. FLOWCHART



## 6. RESULT





## **7. ADVANTAGES & DISADVANTAGES**

### ***Advantages:***

- Random Forest Classifier give the accurate result of the prediction upto 88% which is the algorithm we used for prediction.
- H-1B visa benefit, and perhaps the main reason for its popularity, is the board requirements associated with qualifying for the visa
- Duration of Stay
- Portability
- Anyone Can Apply
- Dual Intent (pursue legal permanent residency) while under H-1B non-immigrant status.

### ***Disadvantages:***

- Lottery.
- Extensions.
- Due to lottery process,there are strict dates the must be adhgered to during process.
- Fees.

## **8. APPLICATIONS**

- Deployment for the project can be in the form of an interactive Web Based Platform, where users can enter their details as per the model requirement and get the predictions as a result
- A web application can be used to launch the model for direct customer use
- The model can also be extended to predict visa status of othe visa types, subject to availability of dataset.

## **9. CONCLUSION**

In this project, we have established the application to predict the outcome of H-1B visa applications based on the attributes of the applicant and ,several machine learning models like Random Forest Classifier algorithm can be used. Finally, this can be integrated to a web application.



## 10. FUTURE SCOPE

In further Random Forest Classifier algorithm can be applied on other data sets available for visa approvals to further investigate its accuracy. A rigorous analysis of other machine learning algorithms other than these six can also be done in future to investigate the power of machine learning algorithms for visa status prediction. In further study, we will try to conduct experiments on larger data sets or try to tune the model so as to achieve the state -of-art performance of the model and a great UI support system making it complete web application model.

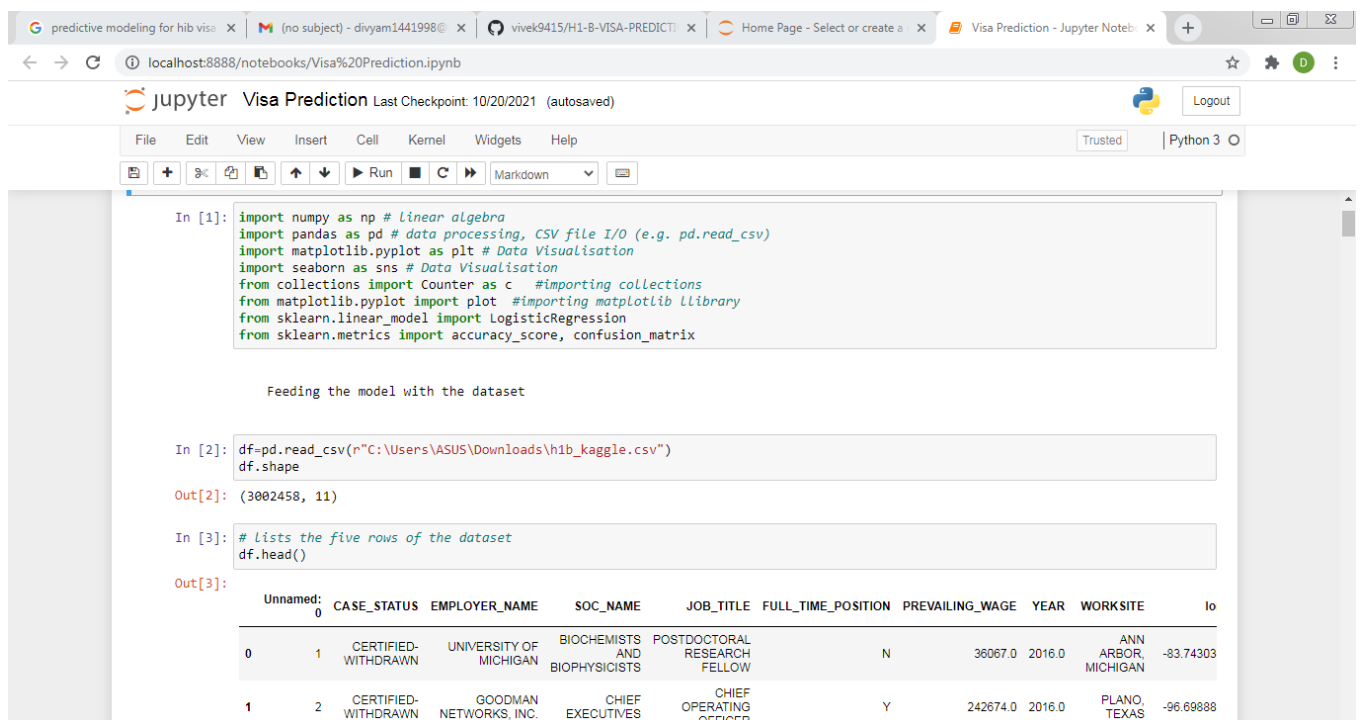
## 11.BIBILOGRAPHY

<https://towardsdatascience.com/predicting-h-1b-status-using-random-forest-dc199a6d254c>

<https://smartinternz.com/guided-project/visa-approval-prediction>

## APPENDIX

### Python(Source Code)



```
In [1]: import numpy as np # Linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt # Data Visualisation
import seaborn as sns # Data Visualisation
from collections import Counter as c #importing collections
from matplotlib.pyplot import plot #importing matplotlib library
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix

Feeding the model with the dataset

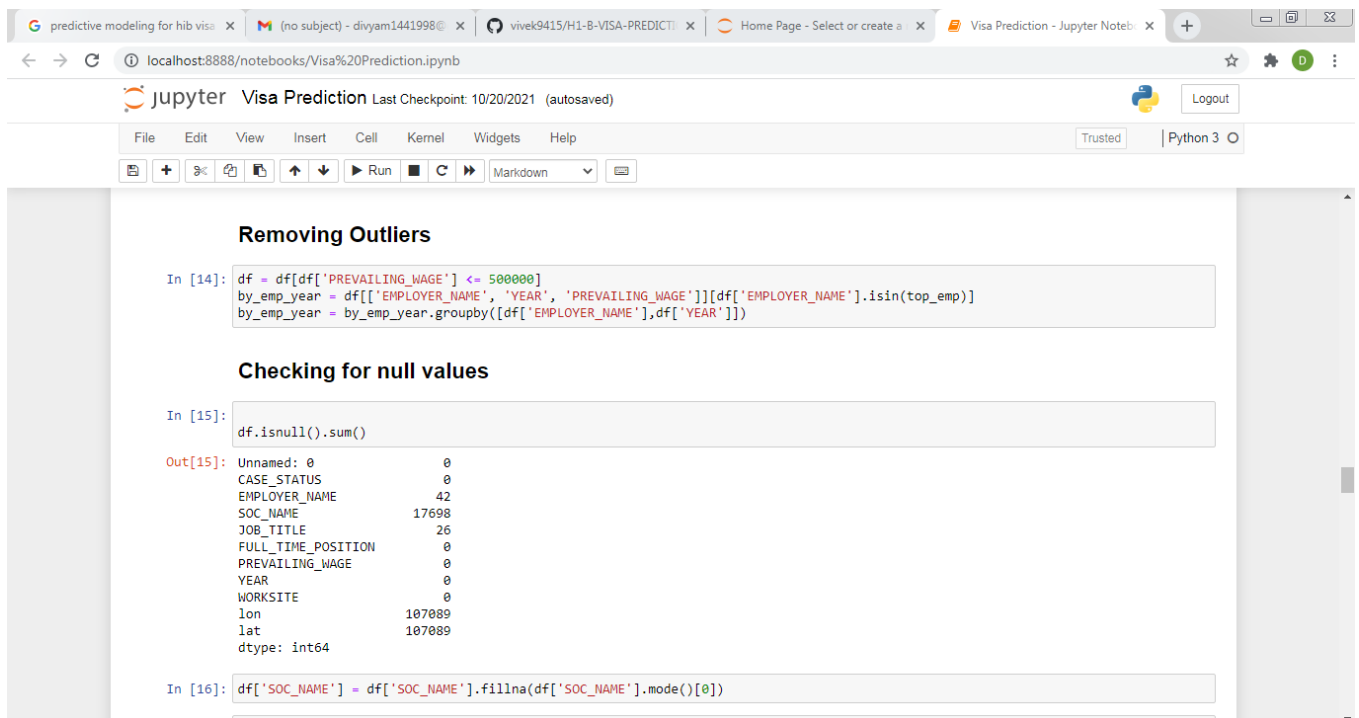
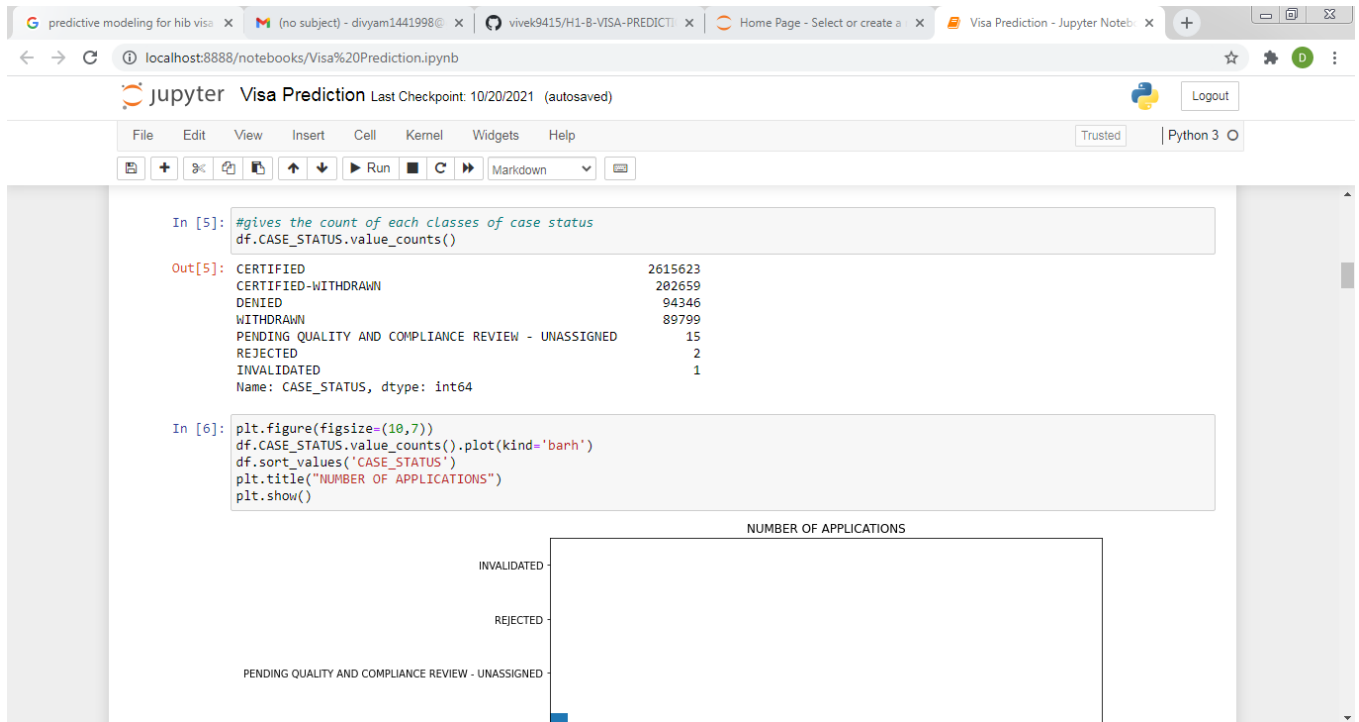
In [2]: df=pd.read_csv(r"C:\Users\ASUS\Downloads\h1b_kaggle.csv")
df.shape

Out[2]: (3002458, 11)

In [3]: # Lists the five rows of the dataset
df.head()

Out[3]:
```

Unnamed: 0	CASE_STATUS	EMPLOYER_NAME	SOC_NAME	JOB_TITLE	FULL_TIME_POSITION	PREVAILING_WAGE	YEAR	WORKSITE	lo	
0	1	CERTIFIED-WITHDRAWN	UNIVERSITY OF MICHIGAN	BIOCHEMISTS AND BIOPHYSICISTS	POSTDOCTORAL RESEARCH FELLOW	N	36067.0	2016.0	ANN ARBOR, MICHIGAN	-83.74303
1	2	CERTIFIED-WITHDRAWN	GOODMAN NETWORKS, INC.	CHIEF EXECUTIVES	CHIEF OPERATING OFFICER	Y	242674.0	2016.0	PLANO, TEXAS	-96.69888



predictive modeling for hib visa x (no subject) - divyam1441998@ x vivek9415/H1-B-VISA-PREDICTI x Home Page - Select or create a x Visa Prediction - Jupyter Noteb x

localhost:8888/notebooks/Visa%20Prediction.ipynb

jupyter Visa Prediction Last Checkpoint: 10/20/2021 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [16]: `df['SOC_NAME'] = df['SOC_NAME'].fillna(df['SOC_NAME'].mode()[0])`

In [17]: `df.isnull().sum()`

Out[17]:

Unnamed: 0	0
CASE_STATUS	0
EMPLOYER_NAME	42
SOC_NAME	0
JOB_TITLE	26
FULL_TIME_POSITION	0
PREVAILING_WAGE	0
YEAR	0
WORKSITE	0
lon	107089
lat	107089
dtype: int64	

**Label encoding the CASE\_STATUS feature**

In [18]: `df['CASE_STATUS'] = df['CASE_STATUS'].map({'CERTIFIED' : 0, 'CERTIFIED-WITHDRAWN' : 1, 'DENIED' : 2, 'WITHDRAWN' : 3, 'PENDING QUAL`

In [19]: `df['FULL_TIME_POSITION'] = df['FULL_TIME_POSITION'].map({'N' : 0, 'Y' : 1})`  
`df.head()`

Out[19]:

Unnamed: 0	CASE_STATUS	EMPLOYER_NAME	SOC_NAME	JOB_TITLE	FULL_TIME_POSITION	PREVAILING_WAGE	YEAR	WORKSITE	lon	lat
0										

predictive modeling for hib visa x (no subject) - divyam1441998@ x vivek9415/H1-B-VISA-PREDICTI x Home Page - Select or create a x Visa Prediction - Jupyter Noteb x

localhost:8888/notebooks/Visa%20Prediction.ipynb

jupyter Visa Prediction Last Checkpoint: 10/20/2021 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [22]: `from sklearn import preprocessing`  
`le = preprocessing.LabelEncoder()`  
`le.fit(df.SOC_NAME1)`  
`# print list(le.classes_)`  
`df['SOC_N']=le.transform(df['SOC_NAME1'])`

In [23]: `df = df.drop(['SOC_NAME1'], axis=1)`

In [24]: `sns.heatmap(df.corr(), annot=True, cmap="RdYlGn", annot_kws={"size":15})`

Out[24]: `<matplotlib.axes._subplots.AxesSubplot at 0x16e45e80>`

CASE_STATUS	1	-0.012	-0.018	-0.038	-0.002
FULL_TIME_POSITION	-0.012	1	0.2	-0.39	-0.00021
PREVAILING_WAGE	-0.018	0.2	1	0.1	0.003
YEAR	-0.038	-0.39	0.1	1	0.002
SOC_N	-0.002	-0.00021	0.003	0.002	1

```
predictive modeling for hib vis... (no subject) - divyam1441998@ x vivek9415/H1-B-VISA-PREDICTI... Home Page - Select or create a... Visa Prediction - Jupyter Noteb...
localhost:8888/notebooks/Visa%20Prediction.ipynb
jupyter Visa Prediction Last Checkpoint: 10/20/2021 (autosaved)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
In [25]: selcols=["FULL_TIME_POSITION","PREVAILING_WAGE","YEAR","SOC_N"]
x=pd.DataFrame(df,columns=selcols)
y=pd.DataFrame(df,columns=['CASE_STATUS'])

In [26]: 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)

##Model Fitting by using Random Forest Classifier

In [27]: from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier()
rf.fit(x_train, y_train)

C:\Users\ASUS\Anaconda3\envs\tensorflow\lib\site-packages\ipykernel_launcher.py:3: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
This is separate from the ipykernel package so we can avoid doing imports until

Out[27]: RandomForestClassifier()

In [28]: y_pred_rf = rf.predict(x_test)
print(y_pred_rf)

[0 0 0 ... 0 0 0]

In [29]: from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred_rf))
```

```
predictive modeling for hib vis... (no subject) - divyam1441998@ x vivek9415/H1-B-VISA-PREDICTI... Home Page - Select or create a... Visa Prediction - Jupyter Noteb...
localhost:8888/notebooks/Visa%20Prediction.ipynb
jupyter Visa Prediction Last Checkpoint: 10/20/2021 (autosaved)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
In [29]: from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred_rf))

          0          1          2          3         support
accuracy      0.88      0.99      0.93      0.87      784458
macro avg      0.47      0.09      0.16      0.20      60711
weighted avg    0.25      0.04      0.07      0.20      27545
          3          4          6
accuracy      0.14      0.01      0.02      0.00         6
macro avg      0.00      0.00      0.00      0.00         1
weighted avg    0.00      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

C:\Users\ASUS\Anaconda3\envs\tensorflow\lib\site-packages\sklearn\metrics\_classification.py:1248: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))

In [30]: c(y_pred_rf)

Out[30]: Counter({0: 881670, 1: 12081, 2: 4528, 3: 1695})

In [31]: accuracy = accuracy_score(y_test,y_pred_rf)
accuracy

Out[31]: 0.8688162102460738

In [32]: import pickle
pickle.dump(rf,open('Visarf.pkl','wb'))
```

```
Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help

ditor - D:\VisaApprovalPrediction-main\Flask App\app.py

home.html Visa_Approval.html resultVA.html app.py app1.py

8 import numpy as np
9 import pandas as pd
10 from flask import Flask, request, render_template
11 import pickle
12 import os
13 import requests
14 import json
15
16 app = Flask(__name__)
17 model = pickle.load(open('D:\VisaApprovalPrediction\Training\Visarf.pkl', 'rb'))
18
19
20 # NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
21 API_KEY = "qkiuVnigPTSH8XW33P1j6CfW50UqPjZ2I3noxm7PYEr"
22 token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey": API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
23 mltoken = token_response.json()["access_token"]
24
25 header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
26
27
28 @app.route('/')
29 def home():
30     return render_template('home.html')
31
32 @app.route('/Visa_Approval')
33 def Visa_Approval():
34     return render_template('Visa_Approval.html')
35
36 @app.route('/predict', methods=['POST'])
37 def predict():
38     input_features = [float(x) for x in request.form.values()]
39     features_value = np.array(input_features)
40
41     payload_scoring = {"input_data": [{"field": ['FULL_TIME_POSITION', 'PREVAILING_WAGE', 'YEAR', 'SOC_N'], "values": [input_features]}]}
42
43     response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/9078763e-b479-4774-9abe-a28dab485e9/predictions?version=2021-10-26', json=payload_scoring)
44     print(response_scoring)
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 9 Column: 20 Memory: 74 %
```