# Predictive Modeling for H1b Visa Approval Using Machine Learning

## Final Project Report

# 1.INTRODUCTION

## 1.1 Project overview

The H-1B visa program in the United States serves as a pivotal avenue for employers seeking highly skilled foreign workers in specialty occupations. Established to address workforce gaps in fields such as technology, engineering, and medicine, the H-1B visa allows U.S. companies to hire non-immigrant professionals for up to six years. This visa category requires that applicants possess at least a bachelor's degree or equivalent experience in their field of expertise. It not only benefits employers by filling crucial roles with qualified individuals but also contributes to the U.S. economy by fostering innovation and competitiveness in key industries. As one of the most sought-after visa categories, the H-1B program plays a crucial role in shaping the landscape of America's labour market and its global standing in technology and other specialized fields.

## 1.2 Objectives

- **Develop Accurate Predictive Models:** Create machine learning models that accurately predict the likelihood of H1B visa approval based on applicant data.
- **Improve Decision-Making:** Provide insights into the factors influencing visa approval to assist applicants, employers, and immigration authorities in making informed decisions.
- **Enhance Efficiency:** Optimize the visa application process by identifying trends and patterns that streamline the approval process and reduce processing times.

# 2. Project Initialization and Planning Phase

## 2.1 Define Problem Statement

Predictive Modeling for H1B Visa Approval Using Machine Learning is to develop a model that accurately predicts the likelihood of H1B visa approval based on applicant data, aiming to assist applicants, immigration authorities, and employers in making informed decisions.

## 2.2 Project Proposal (Proposed solution)

The proposed project, "Predictive Modeling for H1b Visa Approval Using Machine Learning" is to develop a machine learning model that predicts the likelihood of H1B visa approval based on applicant data, leveraging historical visa outcomes, demographic factors, job specifics, and company information. This model aims to enhance decision-making for applicants, employers, and immigration authorities by providing accurate predictions and insights into visa approval factors, thereby optimizing the visa application process and improving transparency.

## 2.3 Initial Project Planning

In Initial Project Planning involves project planning phase, we will define clear objectives to develop a predictive model for H1B visa approval using machine learning techniques. Stakeholders such as immigration experts, data scientists, and potential users will be identified to ensure alignment of project goals and expectations. We will establish a detailed project plan encompassing data collection from sources including USCIS databases and public datasets, followed by rigorous data cleaning and preprocessing to prepare the data for modelling.

# 3. Data Collection and Preprocessing Phase

## 3.1 Data Collection Plan and Raw Data Sources Identified

The Data Collection and Preprocessing Phase for Predictive Modeling for H1B Visa Approval, we will gather applicant data such as demographics, job details, and company information from USCIS records and other relevant sources. Subsequently, rigorous preprocessing will include handling missing values, encoding categorical variables, and normalizing data to ensure it is suitable for machine learning model training.

## 3.2 Data Quality Report

The dataset for Predictive Modeling for H1B Visa Approval will undergo rigorous evaluation for completeness, consistency, and accuracy. This includes handling missing values, addressing outliers, and ensuring data compatibility across all sources to maintain integrity and reliability in model training and evaluation.

## 3.3 Data Exploration and preprocessing

Data Exploration involves Conduct comprehensive analysis of applicant demographics, job details, and visa outcomes to identify patterns and correlations crucial for model development. Cleanse data by handling missing values, encoding categorical variables, and normalizing numerical features to prepare it for effective machine learning model training and evaluation.

# 4. Model Development Phase

## 4.1 Feature Selection Report

The Feature Selection Report Utilize techniques like correlation analysis, feature importance from ensemble models, and domain knowledge to identify and select relevant applicant attributes (e.g., demographics, job specifics) crucial for accurate predictive modeling of H1B visa approval likelihood. This ensures optimal model performance and interpretability while minimizing overfitting.

## 4.2 Model Selection Report

The Model Selection Report Evaluate performance of various machine learning algorithms including logistic regression, random forests, and gradient boosting, using metrics such as accuracy, precision, recall, and ROC curves to determine the most effective model for predicting H1B visa approval likelihood. This report ensures the selection of a robust and accurate model tailored to the project's objectives.

## 4.3 Initial Model Training Code, Model Validation and Evaluation Report

The Initial Model Training Code employs selected algorithms on the loan approval dataset, setting the foundation for predictive modeling. Evaluate RandomForestClassifier's performance using accuracy, precision, recall, and ROC curves on test data to assess its effectiveness in predicting H1B visa approval likelihood.

# 5.Model Optimization and Tuning Phase

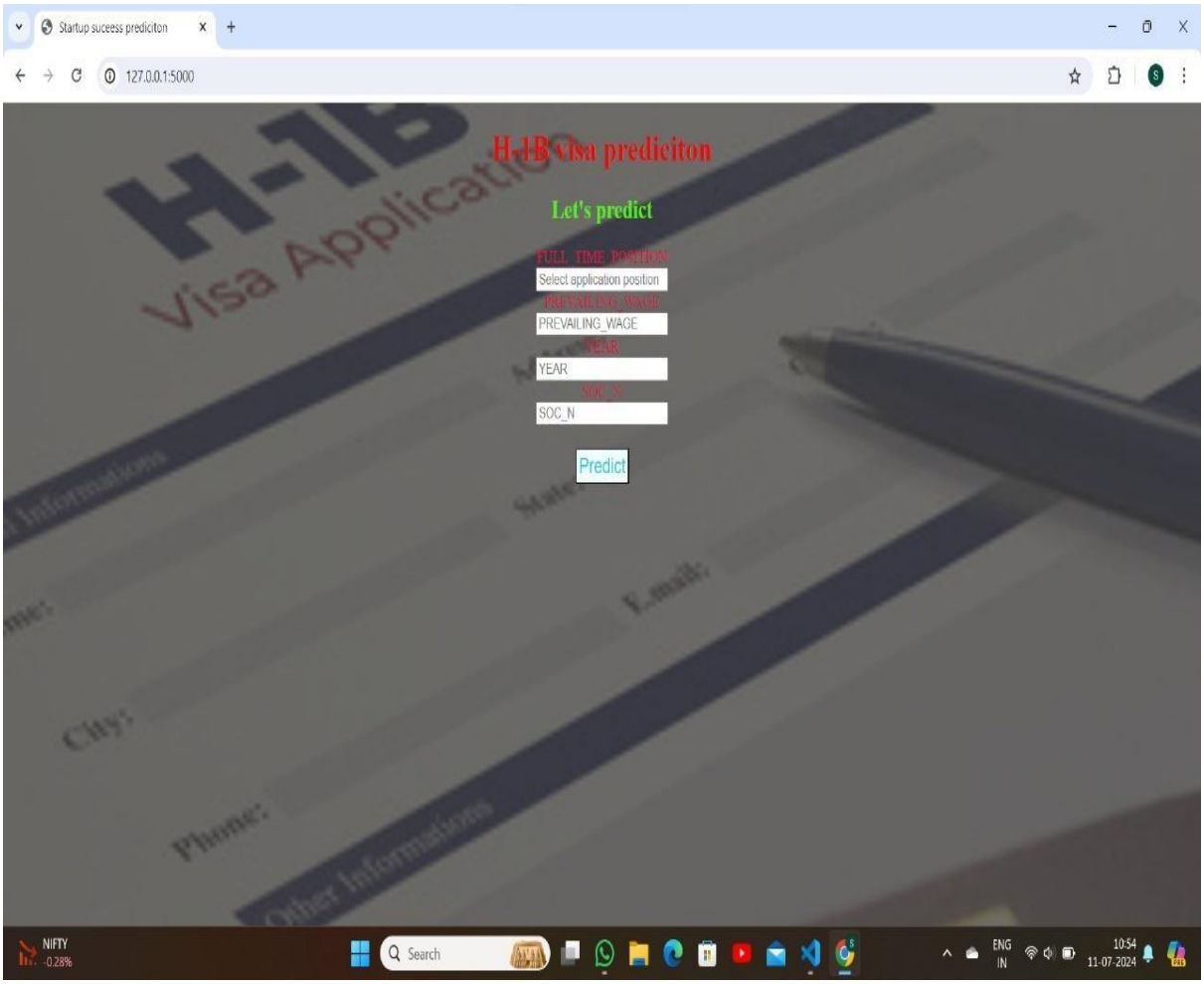
## Final Model Selection Justification

- The Random Forest model is the final model chosen because of its best overall performance compared to the other models.
- It captures the variance in the data very well with minimal prediction error.

# 6. RESULTS

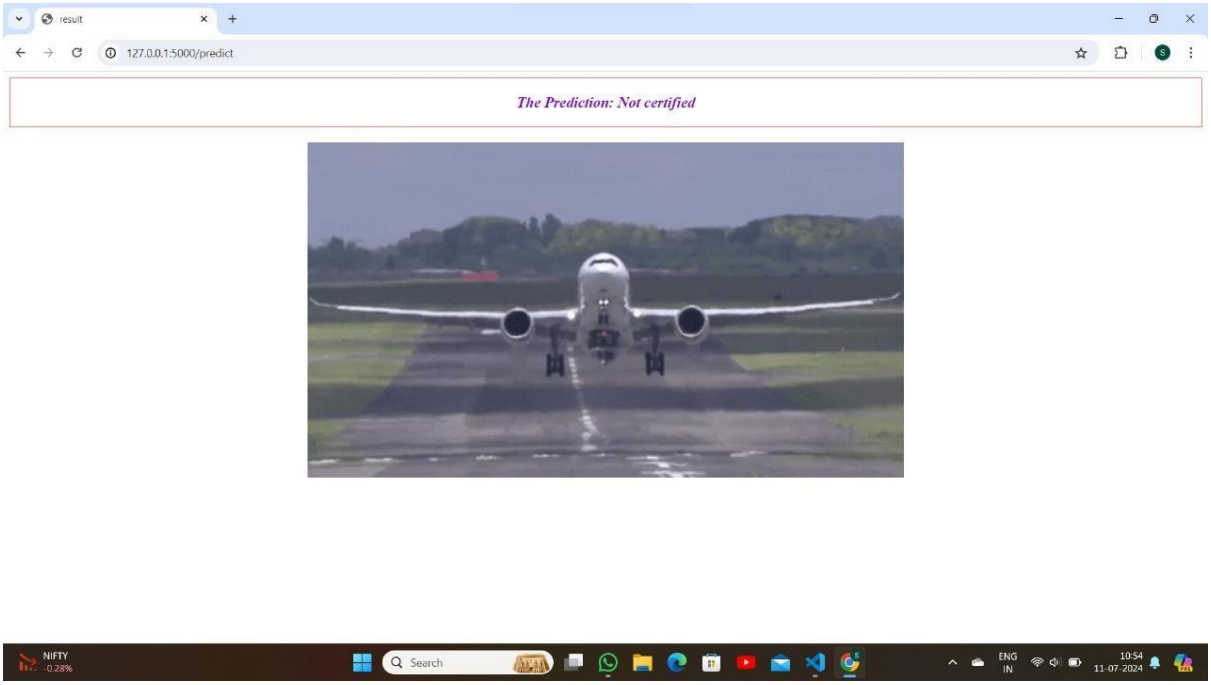
## 6.1 Output Screenshots

## HOME PAGE



# PREDICTION PAGE

# OUTPUT PAGE



# 7.ADVANTAGES AND DISADVANTAGES

## ADVANTAGES:

**1. Insight Generation:** It provides insights into the factors influencing H1B visa approval decisions. Machine learning models can uncover non-obvious patterns and correlations in data that human analysts might overlook.

**2. Automation and Efficiency:** Machine learning models can automate the prediction process once trained, making it faster and more efficient than manual review.

**3. Predictive Accuracy:** With proper data preprocessing, feature selection, and model tuning, machine learning models can achieve high predictive accuracy.

## DISADVANTAGES:

**1. Data Limitations:** The availability and quality of data can significantly impact the model's accuracy and reliability.

**2. Interpretability Issues:** Some machine learning models, such as complex neural networks, can be difficult to interpret.

**3. Model Maintenance:** Machine learning models require ongoing maintenance and updates to remain effective.

# 8.CONCLUSION

In conclusion, a mini project on predictive modelling for H1B visa approval using machine learning represents a valuable application in immigration policy and decision-making. By leveraging historical data and advanced algorithms, the project aims to automate and optimize the visa application process, providing stakeholders with data-driven insights to enhance efficiency, fairness, and transparency. Through this initiative, we aim to contribute to improved decision-making for applicants, employers, and immigration authorities, fostering a more informed and equitable visa approval system.

# 9.FUTURE SCOPE

The Future Scope of the predictive modeling for H1B visa approval using machine learning includes:

1. **Enhanced Accuracy:** Continuously improving model accuracy through refined data collection, feature engineering, and advanced machine learning techniques.

2. **Integration Of New Data Sources**: Incorporating real-time data sources and updates to adapt to changing immigration policies and economic trends.

3. **Automation and Efficiency:** Further automating the visa application review process to reduce processing times and administrative burdens.

**4. Policy Insights:** Providing policymakers with insights into the impact of visa policies and potential adjustments based on predictive modeling outcomes.

# 10. APPENDIX

## 10.1. Source Code

## Code Snippets

```python
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib as plt
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
```

Python

```python
df = pd.read_csv("h1b_kaggle.csv")
df.shape
df.head()
```

Python

| | Unnamed: 0 | CASE_STATUS | EMPLOYER_NAME | SOC_NAME | JOB_TITLE | FULL_TIME_POSITION | PREVAILING_WAGE | YEAR | WORKSITE | lon | lat |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | CERTIFIED-WITHDRAWN | UNIVERSITY OF MICHIGAN | BIOCHEMISTS AND BIOPHYSICISTS | POSTDOCTORAL RESEARCH FELLOW | N | 36067.0 | 2016.0 | ANN ARBOR, MICHIGAN | -83.743038 | 42.280826 |
| 1 | 2 | CERTIFIED-WITHDRAWN | GOODMAN NETWORKS, INC. | CHIEF EXECUTIVES | CHIEF OPERATING OFFICER | Y | 242674.0 | 2016.0 | PLANO, TEXAS | -96.698886 | 33.019843 |
| 2 | 3 | CERTIFIED-WITHDRAWN | PORTS AMERICA GROUP, INC. | CHIEF EXECUTIVES | CHIEF PROCESS OFFICER | Y | 193066.0 | 2016.0 | JERSEY CITY, NEW JERSEY | -74.077642 | 40.728158 |
| 3 | 4 | CERTIFIED-WITHDRAWN | GATES CORPORATION, A WHOLLY-OWNED SUBSIDIARY O... | CHIEF EXECUTIVES | REGIONAL PRESIDEN, AMERICAS | Y | 220314.0 | 2016.0 | DENVER, COLORADO | -104.990251 | 39.739236 |
| 4 | 5 | WITHDRAWN | PEABODY INVESTMENTS CORP. | CHIEF EXECUTIVES | PRESIDENT MONGOLIA AND INDIA | Y | 157518.4 | 2016.0 | ST. LOUIS, MISSOURI | -90.199404 | 38.627003 |

```python
df.info()
```

Python

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3002458 entries, 0 to 3002457
Data columns (total 11 columns):
 #   Column             Dtype
---  ------             -----
 0   Unnamed: 0         int64
 1   CASE_STATUS        object
 2   EMPLOYER_NAME      object
 3   SOC_NAME           object
 4   JOB_TITLE          object
 5   FULL_TIME_POSITION object
 6   PREVAILING_WAGE    float64
 7   YEAR               float64
 8   WORKSITE           object
 9   lon                float64
 10  lat                float64
dtypes: float64(4), int64(1), object(6)
memory usage: 252.0+ MB
```

```python
df.CASE_STATUS.value_counts()
```

```
CASE_STATUS
CERTIFIED                                      2615623
CERTIFIED-WITHDRAWN                             202659
DENIED                                           94346
WITHDRAWN                                        89799
PENDING QUALITY AND COMPLIANCE REVIEW - UNASSIGNED  15
REJECTED                                             2
INVALIDATED                                          1
Name: count, dtype: int64
```

```
df.isnull().sum()
```

```
Unnamed: 0             0
CASE_STATUS           13
EMPLOYER_NAME         59
SOC_NAME           17734
JOB_TITLE             43
FULL_TIME_POSITION    15
PREVAILING_WAGE       85
YEAR                  13
WORKSITE               0
lon               107242
lat               107242
dtype: int64
```

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

Python

```
df['CASE_STATUS'] = df['CASE_STATUS'].map({'CERTIFIED':0, 'CERTIFIED-WITHDRAWN': 1, 'DENIED': 2, 'WITHDRAWN': 3, 'PENDING QUALITY AND COMPLIANCE REVIEW UNASSIGNED' : 4,
'REJECTED': 5, 'INVALIDATED': 6})
```

Python

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

Python

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

Python

| | Unnamed: 0 | CASE_STATUS | EMPLOYER_NAME | SOC_NAME | JOB_TITLE | FULL_TIME_POSITION | PREVAILING_WAGE | YEAR | WORKSITE | lon | lat |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1.0 | UNIVERSITY OF MICHIGAN | BIOCHEMISTS AND BIOPHYSICISTS | POSTDOCTORAL RESEARCH FELLOW | 0 | 36067.0 | 2016.0 | ANN ARBOR, MICHIGAN | -83.743038 | 42.280826 |
| 1 | 2 | 1.0 | GOODMAN NETWORKS, INC. | CHIEF EXECUTIVES | CHIEF OPERATING OFFICER | 1 | 242674.0 | 2016.0 | PLANO, TEXAS | -96.698886 | 33.019843 |
| 2 | 3 | 1.0 | PORTS AMERICA GROUP, INC. | CHIEF EXECUTIVES | CHIEF PROCESS OFFICER | 1 | 193066.0 | 2016.0 | JERSEY CITY, NEW JERSEY | -74.077642 | 40.728158 |
| 3 | 4 | 1.0 | GATES CORPORATION, A WHOLLY-OWNED SUBSIDIARY O... | CHIEF EXECUTIVES | REGIONAL PRESIDEN, AMERICAS | 1 | 220314.0 | 2016.0 | DENVER, COLORADO | -104.990251 | 39.739236 |
| 4 | 5 | 3.0 | PEABODY INVESTMENTS CORP. | CHIEF EXECUTIVES | PRESIDENT MONGOLIA AND INDIA | 1 | 157518.4 | 2016.0 | ST. LOUIS, MISSOURI | -90.199404 | 38.627003 |

```
    import sys
    df['SOC_NAME1'] = 'others'
    df['SOC_NAME1'] [df['SOC_NAME'].str.contains('computer', 'software')] = 'it'
    df['SOC_NAME1'] [df['SOC_NAME'].str.contains('chief', 'management')] = 'manager'
    df['SOC_NAME1'] [df['SOC_NAME'].str.contains('mechanical')] = 'mechanical'
    df['SOC_NAME1'] [df['SOC_NAME'].str.contains('database')] = 'database'
    df['SOC_NAME1'][df['SOC_NAME'].str.contains('sales', 'market')] = 'scm'
    df['SOC_NAME1'] [df['SOC_NAME'].str.contains('financial')] = 'finance'
    df['SOC_NAME1'] [df['SOC_NAME'].str.contains('public', 'fundraising')] = 'pr'
    df['SOC_NAME1'] [df['SOC_NAME'].str.contains('education', 'law')] = 'administrative'
    df['SOC_NAME1'][df['SOC_NAME'].str.contains('auditors', 'compliance')] = 'audit'
    df['SOC_NAME1'] [df['SOC_NAME'].str.contains('distribution', 'logistics')] = 'scm'
    df['SOC_NAME1'] [df['SOC_NAME'].str.contains ('recruiters', 'human')] = 'hr'
    df['SOC_NAME1'] [df['SOC_NAME'].str.contains('agricultural', 'farm')] = 'agri'
    df['SOC_NAME1'][df['SOC_NAME'].str.contains('construction', 'architectural')] = 'estate'
    df['SOC_NAME1'] [df['SOC_NAME'].str.contains('forencsic', 'health')] = 'medical'
    df['SOC_NAME1'][df['SOC_NAME'].str.contains('teachers')] = 'education'
```

Python

```
C:\Users\sushm\AppData\Local\Temp\ipykernel_25128\2877348365.py:3: FutureWarning: ChainedAssignmentError: behaviour will change in pandas 3.0!
You are setting values through chained assignment. Currently this works in certain cases, but when using Copy-on-Write (which will become the default behaviour in pandas 3.0) this will
A typical example is when you are setting values in a column of a DataFrame, like:

df["col"][row_indexer] = value

Use `df.loc[row_indexer, "col"] = values` instead, to perform the assignment in a single step and ensure this keeps updating the original `df`.

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

  df['SOC_NAME1'] [df['SOC_NAME'].str.contains('computer', 'software')] = 'it'
C:\Users\sushm\AppData\Local\Temp\ipykernel_25128\2877348365.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

```
    df = df.drop(['Unnamed: 0', 'EMPLOYER_NAME', 'SOC_NAME', 'JOB_TITLE','WORKSITE', 'lon', 'lat'], axis = 1)
```

Python

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

Python

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

Python

```
    import seaborn as sns
    import matplotlib.pyplot as plt

    # Use a valid colormap name
    sns.heatmap(df.corr(), annot=True, cmap="RdYlGn", annot_kws={"size":15})
    plt.show()
```

Python

Click to add a breakpoint

```
df['CASE_STATUS'].fillna(df['CASE_STATUS'].mode()[0])
```

Python

```
    selcols=["FULL_TIME_POSITION", "PREVAILING_WAGE", "YEAR", "SOC_N"]
    x=pd.DataFrame(df,columns=selcols)
    y=pd.DataFrame(df,columns=['CASE_STATUS'])
```

Python

```
    x.columns
```

Python

```
Index(['FULL_TIME_POSITION', 'PREVAILING_WAGE', 'YEAR', 'SOC_N'], dtype='object')
```

```
    x.head(10)
```

Python

```
uni=x['SOC_N'].unique()
print(uni)
```

```
[2 1 0]
```

```
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)
```

```
#from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
rf = DecisionTreeClassifier()
rf.fit(x_train, y_train)
```

```
DecisionTreeClassifier
DecisionTreeClassifier()
```

```
y_pred_rf =rf.predict(x_test)
print(y_pred_rf)
```

```
[0. 0. 0. ... 0. 0. 0.]
```

```
y_pred_rf =rf.predict(x_test)
print(y_pred_rf)
```

Python

```
[0. 0. 0. ... 0. 0. 0.]
```

```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred_rf))
```

Python

```
c:\Users\sushm\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\metrics\_classification.py:1531: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
c:\Users\sushm\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\metrics\_classification.py:1531: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
              precision    recall  f1-score   support

         0.0       0.88      0.99      0.93    784464
         1.0       0.49      0.09      0.15     60711
         2.0       0.25      0.03      0.06     27545
         3.0       0.16      0.01      0.01     27253
         6.0       0.00      0.00      0.00         1

    accuracy                           0.87    899974
   macro avg       0.35      0.22      0.23    899974
weighted avg       0.81      0.87      0.82    899974

c:\Users\sushm\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\metrics\_classification.py:1531: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

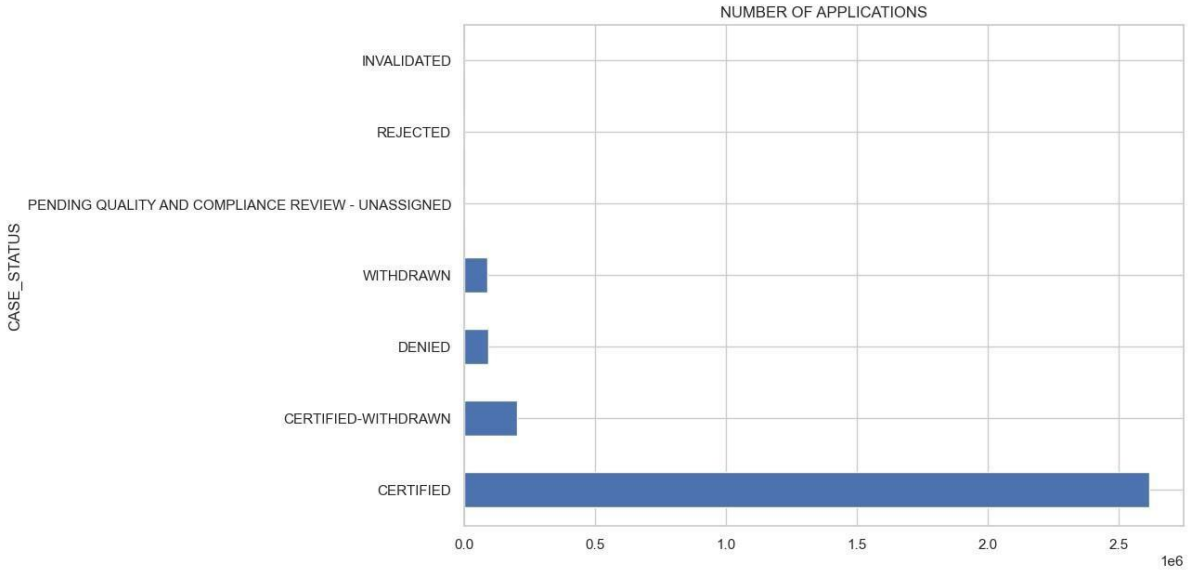
```
(variable) y_pred_rf: ndarray
c(y_pred_rf)
```

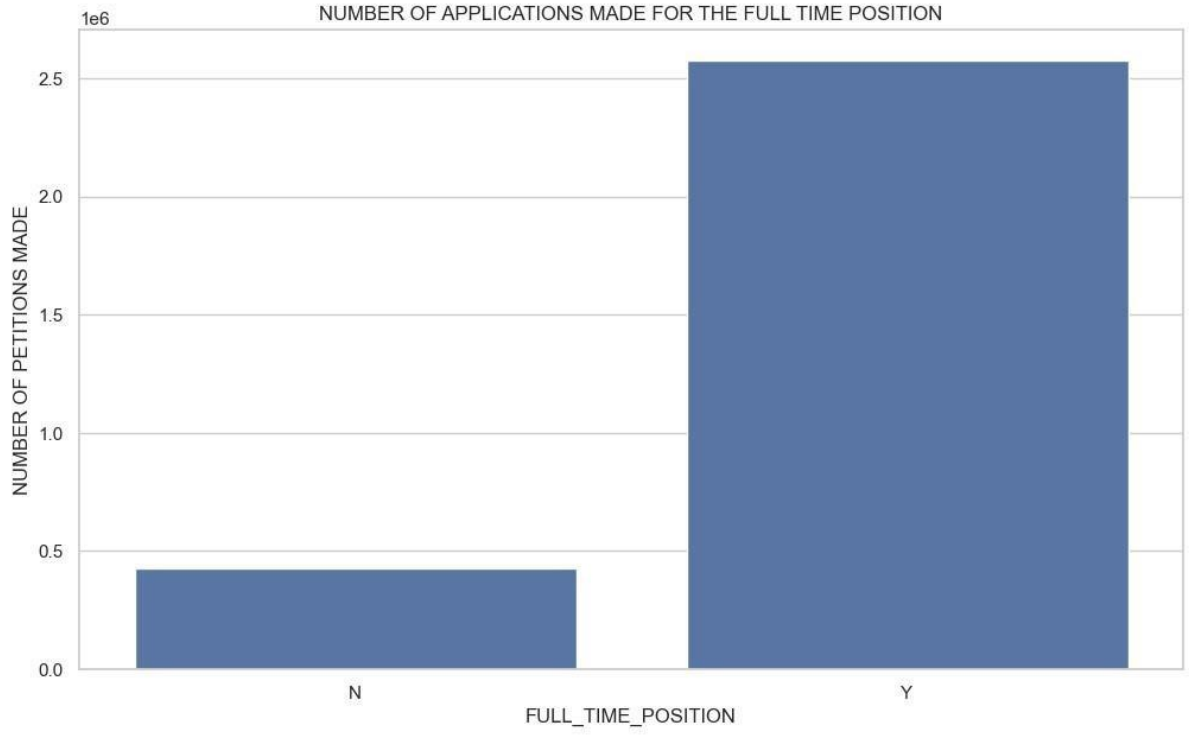
```
Counter({np.float64(0.0): 883608,
         np.float64(1.0): 11260,
         np.float64(2.0): 3791,
         np.float64(3.0): 1315})
```

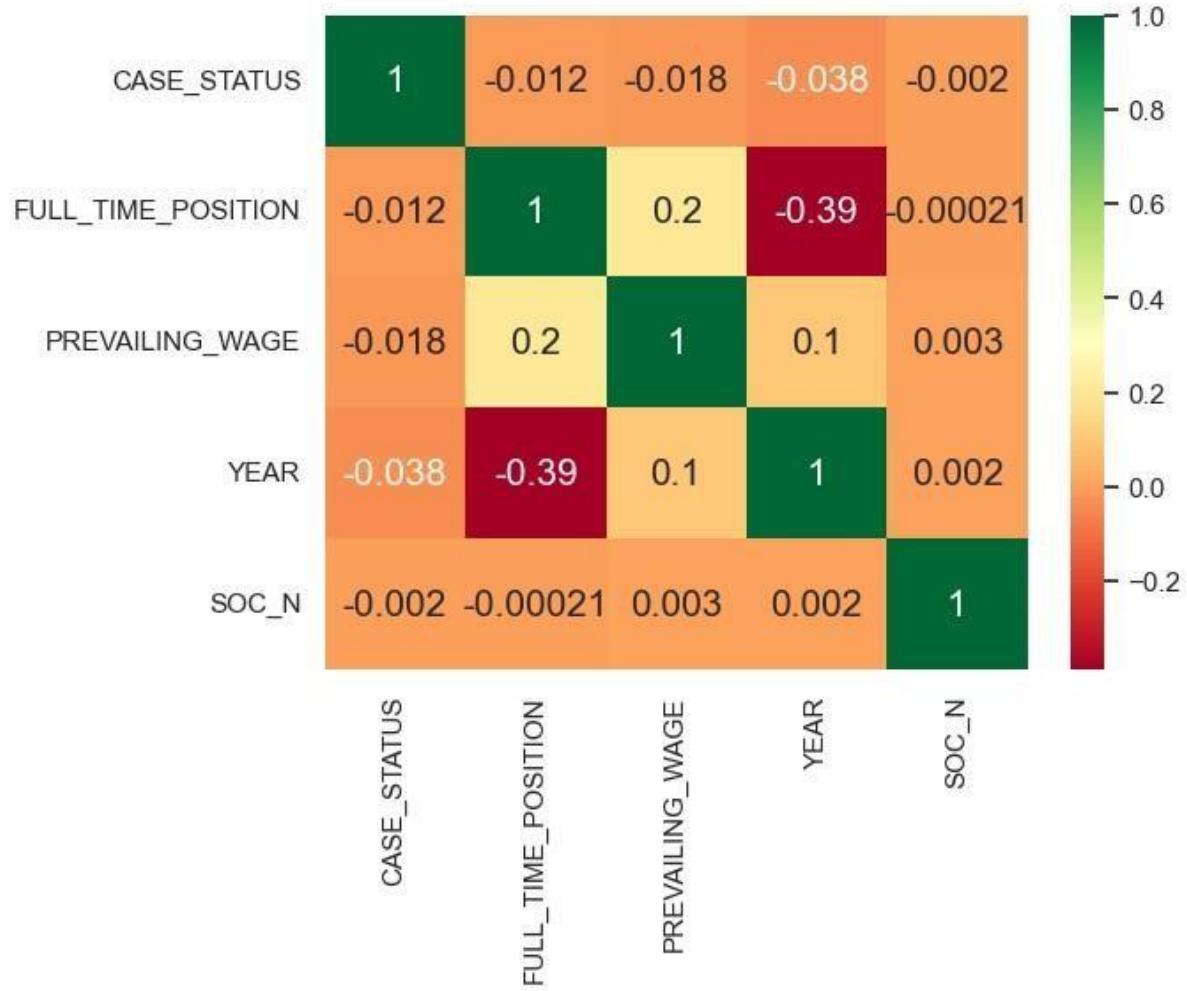
```
accuracy = accuracy_score(y_test,y_pred_rf)
accuracy
```

```
0.8698217948518513
```

```
import pickle
pickle.dump(rf,open('Visarf.pkl','wb'))
```






## INDEX.HTML

```
<!DOCTYPE html>

<html lang="en">

 <head>

 <meta charset="UTF-8" />

 <meta name="viewport" content="width=device-width”, initial-scale=1.0">

 <link rel="stylesheet" href="style.css" />

 <title>Startup sucess prediciton</title>

 <style>  body {  background: url("static/1.jpg") center;

height: 100%; background-position: center; background-
```

```
size: cover; background-repeat: no-repeat; position:
sticky;
}
h1 { color: rgb(236, 11, 11);
}
.btn {
margin-top: 20px; padding: 3px; background-
color: azure; font-size: larger;
color: rgb (17, 208, 214); cursor: pointer;
}
form { color: crimson; align-content:
center; text-align: center;
}
</style>
</head>
<body>
<h1 style="text-align: center;">H-1B visa prediciton</h1>
<h2 style="color: rgb (76, 245, 14); text-align: center">Let's predict</h2>
<div class="inputs">
<form action="{{url_for('predict')}}" method="post">
<label>FULL_TIME_POSITION</label><br />
<input type="text"
name="'FULL_TIME_POSITION"
placeholder="Select application position"
/><br />
<label>PREVAILING_WAGE</label><br />
<input type="text"
name="PREVAILING_WAGE" placeholder="PREVAILING_WAGE "
/><br />
<label>YEAR</label><br />
<input type="text" name="YEAR"
placeholder="YEAR"
```

```
/><br />
<label>SOC_N</label><br />
<input type="text"
 name="SOC_N"  placeholder="SOC_N"
 /><br />
 <a href="result.html"
 ><button class="btn" type="submit">Predict</button></a
 >
 </form>
 </div>
 <br /><br />
 <section>
 <h3 style="color: blueviolet; text-align: center">
 {{prediction_text}}
 </h3>
 </section>
 </body>
   </html
```

## FLASK PAGE

```
<html>
 <head>
<title>result</title>
 <style> /*body { background-color: rgba (166, 122, 122, 0.893);
}*/
.output { padding: 20px; border: 1px solid
red;  text-align:  center;  color:  rgb(124,  0,
241);
 font-style: italic;  font-size: larger;
 }
 .result {   display:  block;   margin-left:
auto;  margin-right: auto;  width: 50%;
```

```
}
</style>
</head>
<body>
<h3 class="output"> {{prediction_text}} </h3>
<img class="result" src="static/2.gif" alt="prediction" width="200" />
</body>
  </html>
```

## RESULT PAGE

```
<html>
 <head>
<title>result</title>
<style> /*body {background-color: rgba (166, 122, 122, 0.893);
}*/
 . output {padding: 20px; border: 1px solid
red; text-align: center; color: rgb (124, 0,
241);
 font-style: italic;  font-size: larger;
 }
 .result {   display: block; margin-left:
auto; margin-right: auto; width: 50%;
 }
 </style>
 </head>
 <body>
 <h3 class="output"> {{prediction_text}} </h3>
 <img class="result" src="static/2.gif" alt="prediction" width="200" />
 </body>
</html>
```

## 10.2 GitHub and project Demo link:

GitHub link: Click Here

Project Demo link: Click Here