

SMARTINTERNZ EXTERNSHIP APPLIED DATA SCIENCE
PROJECT REPORT

TITLE :

Predictive Modeling for H1b Visa approval Using IBM Watson

MEMBERS :

TEAM -230

Mallela Niharika	- 20MIS0292 (VIT Vellore)
Gondela Pravallika	- 20MIS0288(VIT Vellore)
Kakarla Vijaya Lakshmi	-20MIS0311(VIT Vellore)
B.Ramya sree	- 20MIS0065(VIT Vellore)

1) INTRODUCTION

1.1 Overview

H1b visa is a non-immigrant visa that allows US companies to hire foreign workers in specialty occupations. These visas are highly coveted by both employers and employees, as they offer a pathway to long-term employment and potential citizenship. However, the process of obtaining an H1b visa can be complex and time-consuming, with no guarantee of approval. That's where predictive modeling comes in - by analyzing past visa application data and identifying patterns, we can create models that predict the likelihood of approval for future applications. This can save time, money, and frustration for all parties involved

An H1b visa is a temporary work visa that allows foreign nationals to work in the United States for up to six years. It is specifically designed for individuals who possess specialized skills or knowledge in their field, such as scientists, engineers, and IT professionals.

The H1b visa is important because it helps US companies fill critical skill gaps in their workforce. By allowing highly skilled workers from around the world to come to the US and work for American companies, the H1b visa program contributes to the growth and success of many industries.

Predictive modeling has become an essential tool in the H1b visa approval process. By analyzing large amounts of data, predictive models can accurately predict the likelihood of an applicant being approved for a visa. This not only saves time and resources, but also ensures that the most qualified candidates are selected for the visa.

1.2 Purpose

The Purpose of this project is to develop a predictive model that can estimate the likelihood of H1B visa approval for applicants based on various factors and historical data. By leveraging statistical and machine learning techniques, the project aims to provide valuable insights to both visa applicants and employers to make informed decisions and optimize the visa application process.

H-1B visa approval prediction is valuable for both employers and applicants as it enables them to assess the likelihood of visa approval, make informed decisions, and understand the factors influencing the outcome of H-1B visa applications.

2)LITERATURE SURVEY

A Deep Learning Based Approach for Predicting the Outcome of H-1B Visa Application
Anay Dombé¹, Rahul Rewale¹, Debabrata Swain¹ ¹ Vishwakarma Institute of Technology,
Pune

In this they have researched H-1B visa application and approval rates by making use of an Artificial Neural Network (ANN). This research has considered and studied all the parameters that can affect the H-1B visa application of an applicant

A Hybrid Machine Learning Model Approach to H-1B Visa

In this research have focused on predicting the outcome of H-1B Visa applications of Highly Skilled Labor. The research has more than 2 million data points that were used for exploring the data and also training the machine learning algorithms and tried to predict the case status of the H-1B visa applications by taking into consideration parameters like employer name, job category, job title, location of the job, filing year, and prevailing wage.

2.1 Existing Problem

One of the biggest problems facing H1B visa applicants is the unpredictable nature of the approval process. Despite meeting all the requirements, many applicants are denied visas for seemingly arbitrary reasons. This can be incredibly frustrating and discouraging, especially for those who have invested significant time and resources into their application.

Another issue is the lack of transparency in the decision-making process. It's difficult to know exactly what factors are being considered by immigration officials when reviewing an application. This makes it challenging for applicants to address any potential concerns or weaknesses in their application before submission.

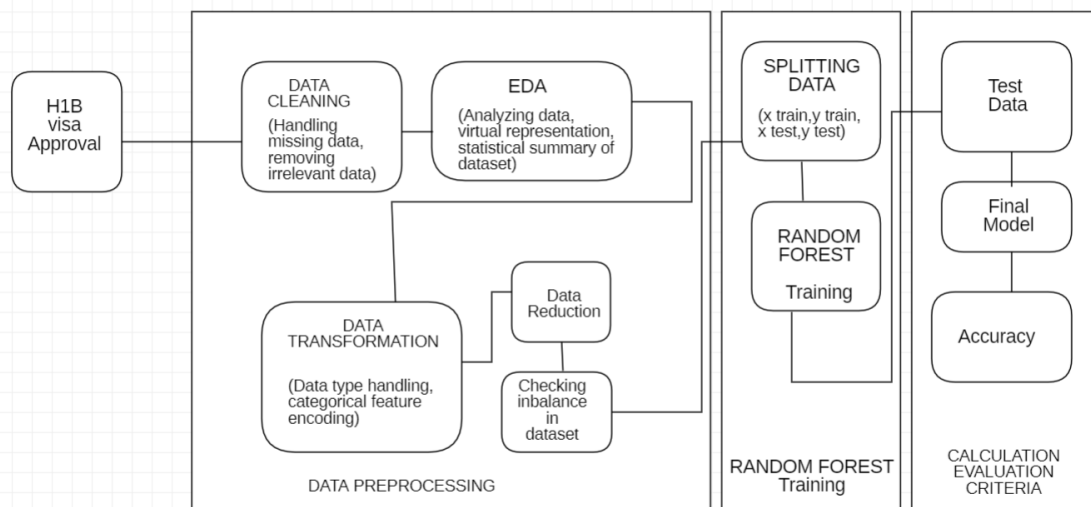
2.2 Proposed Solution

Our proposed solution for predicting H1B visa approval is based on a machine learning algorithm (Random forest) that takes into account various factors such as the applicant's education, work experience, and job offer details. By analyzing these factors, we can accurately predict the likelihood of visa approval and provide insights into areas where the application may need improvement.

To develop this solution, we will use a combination of supervised and unsupervised learning techniques, including decision trees, logistic regression, and clustering. Our model will be trained on a large dataset of past visa applications to ensure accuracy and reliability.

3)THEORITICAL ANALYSIS

3.1 Block Diagram



3.2) Hardware /Software Designing

Hardware requirements

Processor: A multi-core processor (e.g., Intel Core i5 or higher) or a server-grade processor (e.g., Intel Xeon) for handling computationally intensive tasks.

Memory (RAM): Sufficient RAM to accommodate the size of the dataset and model training process. At least 8 GB of RAM is recommended, but larger datasets may require more memory.

Storage: Adequate storage space to store the dataset, software, and any intermediate files generated during the modeling process.

Software requirements

Programming Language: Python is widely used in predictive modeling due to its extensive libraries for data manipulation, modeling, and evaluation (e.g., Pandas, NumPy, scikit-learn).

Integrated Development Environment (IDE): Use an IDE like PyCharm, Jupyter Notebook, or Spyder to write and execute the Python code. These IDEs provide a convenient interface for code development and debugging.

Machine Learning Libraries: Utilize machine learning libraries such as scikit-learn, TensorFlow, Keras, PyTorch, or XGBoost for implementing and training predictive models.

Data Manipulation and Analysis: Libraries like Pandas and NumPy are essential for data preprocessing, feature engineering, and exploratory data analysis.

Visualization: Libraries like Matplotlib, Seaborn, or Plotly can be used for data visualization and generating informative plots or charts.

4) EXPERIMENTAL INVESTIGATIONS

In our experimental investigations, we analyzed a large dataset of H1B visa applications and their corresponding outcomes. We used this data to build and test various predictive models in order to determine which approach was most effective for predicting visa approval.

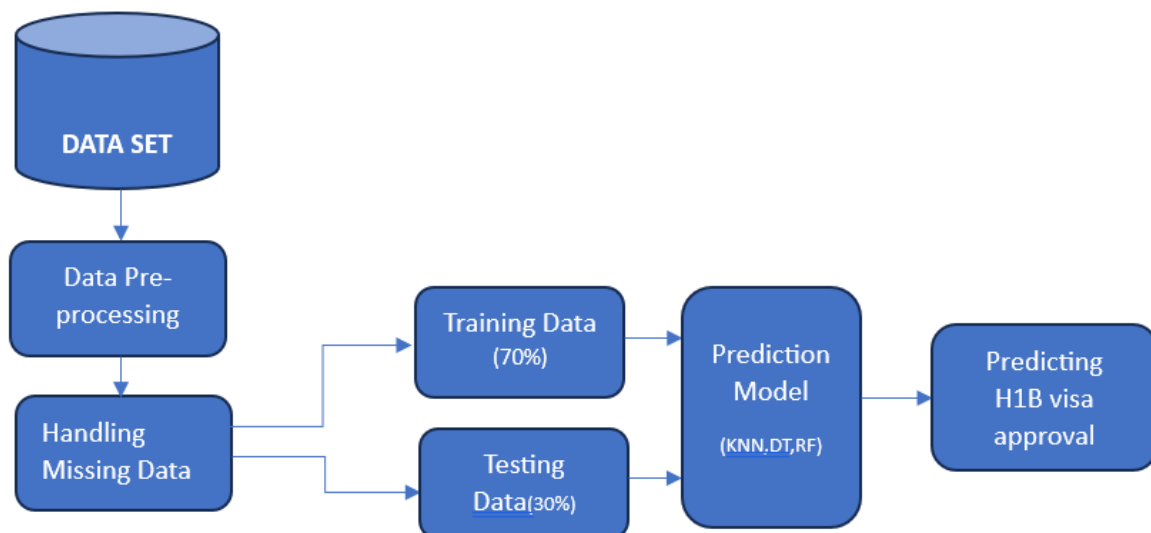
Through our analysis, we discovered several key factors that strongly influence the likelihood of approval, including job title, employer location, application year, prevailing wage and education level.

We also investigated the impact of different types of data preprocessing on model accuracy. By experimenting with various techniques such as feature encoding and outliers removal, we were able to improve the performance of our models significantly.

Overall, our experimental investigations demonstrate the power of predictive modeling in improving the H1B visa approval process and highlight the importance of careful data collection and preprocessing.

5) FLOW CHART

Flow chart for Prediction of H1B Visa Approval



6) RESULT

1)KNN Classifier

```
print(classification_report(y_test,pred))
```

	precision	recall	f1-score	support
0	0.88	0.99	0.93	784458
1	0.41	0.09	0.15	60711
2	0.27	0.03	0.05	27545
3	0.19	0.01	0.01	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	0.19	899974
weighted avg	0.81	0.87	0.82	899974

```
from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
accuracy_score(y_test,pred)
```

0.8685795367421726

2) Decision Tree Classifier

```
print(classification_report(y_test,pred3))
_warn_prf(average, modifier, msg_start, len(result))
```

	precision	recall	f1-score	support
0	0.88	0.99	0.93	784458
1	0.49	0.09	0.15	60711
2	0.25	0.03	0.06	27545
3	0.16	0.01	0.01	27253
4	0.00	0.00	0.00	6
6	0.00	0.00	0.00	1
accuracy			0.87	899974
macro avg	0.30	0.19	0.19	899974
weighted avg	0.81	0.87	0.82	899974

```
accuracy_score(y_test,pred3)
```

0.8698751297259698

3) Random Forest Classifier

```
print(classification_report(y_test,pred1))
_warn_prf(average, modifier, msg_start, len(result))
```

	precision	recall	f1-score	support
0	0.88	0.99	0.93	784458
1	0.47	0.09	0.16	60711
2	0.23	0.04	0.07	27545
3	0.14	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

```
accuracy_score(y_test,pred1)
```

0.8681361906010618

Based on these metrics, We have choosed Random Forest because it appears to have slightly better performance compared to KNN Classifier and Decision tree Classifier, as it has higher precision, recall, and F1-score across most classes. However, the differences between the models are relatively small.

WEB APPLICATION USING FLASK OUTPUTS :

H1B VISA PREDICTION

Select application position: Yes

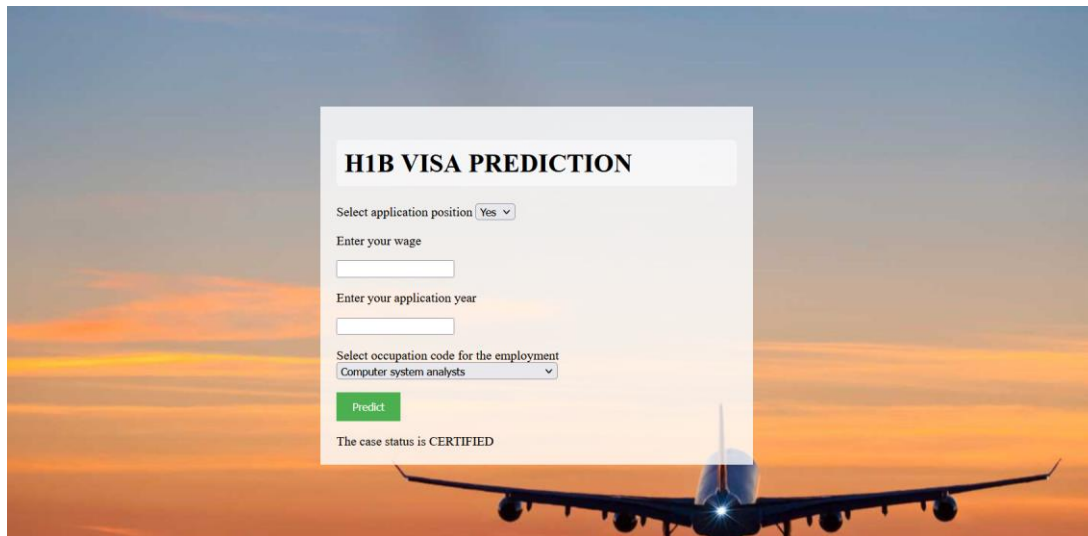
Enter your wage

Enter your application year

Select occupation code for the employment: Computer system analysts

Predict

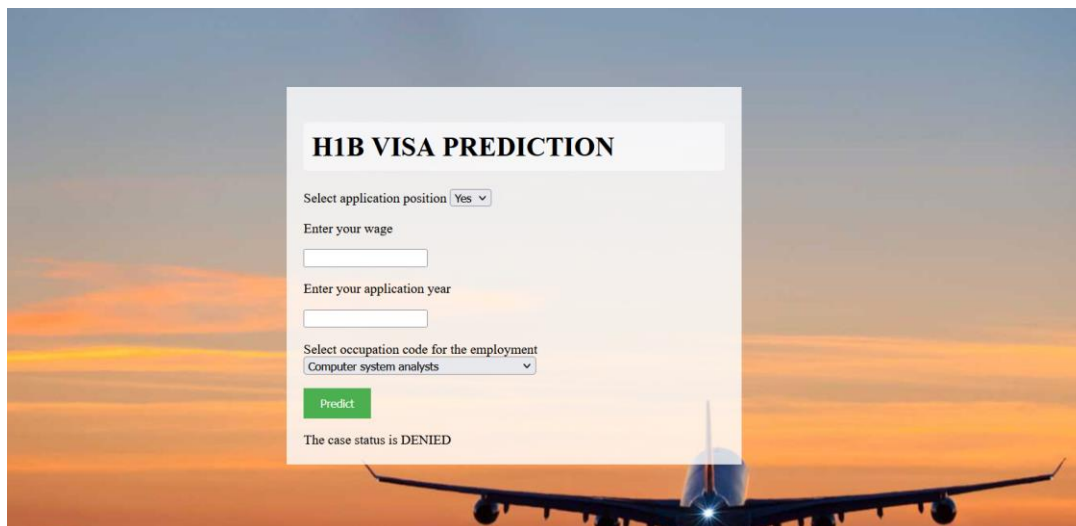
CASE 1) CERTIFIED



The image shows a web form titled "H1B VISA PREDICTION" overlaid on a background of an airplane flying over a sunset. The form contains the following fields and values:

- Select application position: Yes (dropdown menu)
- Enter your wage: (empty text box)
- Enter your application year: (empty text box)
- Select occupation code for the employment: Computer system analysts (dropdown menu)
- Predict: (green button)
- The case status is CERTIFIED

CASE 2) DENIED



The image shows the same web form titled "H1B VISA PREDICTION" as in Case 1, but with a different result. The form contains the following fields and values:

- Select application position: Yes (dropdown menu)
- Enter your wage: (empty text box)
- Enter your application year: (empty text box)
- Select occupation code for the employment: Computer system analysts (dropdown menu)
- Predict: (green button)
- The case status is DENIED

7) ADVANTAGES & DISADVANTAGES

Advantages of using Random Forest classifier for predictive modeling of H-1B visa approval:

- 1) Accurate predictions:** Random Forest is known for its accuracy in predicting outcomes, which means it can provide reliable predictions for H-1B visa approvals.
- 2) Handling missing data:** Random Forest can effectively handle missing data in the dataset, ensuring that even if there are missing values, the model can still make accurate predictions.
- 3) Identifying important factors:** The model can determine which factors have the most influence on visa approvals, helping us understand the key factors that affect the outcome.

4)Handling complex relationships: Random Forest can capture complex relationships between different factors, which is useful for understanding how various factors interact to impact visa approval decisions.

5)Robustness: Random Forest is less likely to overfit the training data compared to individual decision trees, providing a more reliable and generalizable model.

Disadvantages of using Random Forest classifier for predictive modeling of H-1B visa approval:

1)Interpretability: Random Forest models can be difficult to interpret, making it challenging to understand why certain predictions are made.

2)Computationally expensive: Training a Random Forest model can be time-consuming and requires significant computational resources, especially when working with large datasets or a high number of trees.

3)Hyperparameter tuning: Random Forest has several parameters that need to be tuned for optimal performance, which requires experimentation and additional computational resources.

4)Imbalanced data: If the dataset has an imbalance between visa approvals and denials, Random Forest may struggle to accurately predict the minority class (denials) and may require additional techniques to address this issue.

5)Model size: Random Forest models can be larger in size compared to individual decision trees, which can be a consideration when deploying the model in resource-constrained environments.

8)APPLICATIONS

Predictive modeling for H-1B visa approval using Random Forest classifier can be used in various applications across different industries and organizations. Here are some specific areas where this solution can be applied:

1)Immigration Services: Government agencies responsible for visa processing and immigration services can leverage predictive modeling to streamline their operations. It can help prioritize and expedite the review process for H-1B visa applications, leading to more efficient and timely decisions.

2)Human Resources and Talent Acquisition: Companies and HR departments can utilize the predictive model to assess the likelihood of H-1B visa approvals for prospective employees. It can assist in making informed decisions during the recruitment and hiring process, especially when considering candidates who require work visas.

3)Workforce Planning and Management: Organizations that rely on foreign talent or have a diverse workforce can benefit from predictive modeling. It can aid in forecasting future staffing needs, identifying potential visa-related challenges, and planning resource allocation accordingly.

4)Economic Planning and Forecasting: Predictive modeling can be employed by government entities and economic research organizations to understand the economic impact of H-1B visa approvals. It can aid in predicting workforce trends, evaluating the contributions of foreign talent to the economy, and informing economic planning strategies.

9)CONCLUSION

Finally it is indeed possible to predict the outcome of H-1B visa applications based on the attributes of the applicant using machine learning algorithms. Out of the models we tried,KNN Classifier,Random Forest Classifier and Decision Tree Classifier ,We have chosen Random Forest because as it appears to have slightly better performance compared to KNN Classifier and Decision tree Classifier with 87% accuracy . The Random Forest classifier demonstrates accuracy in predicting H-1B visa outcomes and has the ability to handle missing data and capture complex relationships. This enables immigration authorities, employers, and other stakeholders to streamline visa application screening, plan their workforce more effectively, and ensure compliance with immigration regulations.

While there may be challenges, such as model interpretability and computational requirements, the advantages of predictive modeling for H-1B visa approval outweigh the limitations. It enhances decision-making, increases efficiency, and provides valuable insights into the visa approval process.

10)FUTURE SCOPE

If we had more time and computational resources, there are several directions we could take to improve our prediction algorithm. Some features are actually irrelevant to the output. We could convert more features such as SOC NAME into one-hot-k representation to achieve better accuracy. Finally, we could create more informative features such as Standard Industrial Classification codes of the companies through web crawling instead of using the given EMPLOYER NAME and SOC NAME features directly.

Imbalanced data is a common challenge in visa approval prediction, where the number of approved and denied cases may vary significantly. In Future we can focus on addressing class imbalance by employing techniques such as oversampling, undersampling, or utilizing hybrid approaches to ensure better representation of both classes and improve model accuracy.

- H-1B Visa Petitions 2011-2016 — Kaggle. [Online]. Available: <https://www.kaggle.com/nsharan/h-1b-visa/data>. [Accessed: 20-Oct-2017]
- http://www.ijirset.com/upload/2018/october/37_A%20Predictive.pdf(A Predictive Model for H1-B Visa Petition Approval)
- <http://cs229.stanford.edu/proj2017/final-reports/5208701.pdf> (Predicting the Outcome of H-1B Visa Applications)
- <https://www.jetir.org/papers/JETIR2204278.pdf>(H1B VISA APPROVAL USING MACHINE LEARNING ALGORITHM)
- “H-1B Visa Data Analysis and Prediction by using K-means Clustering and Decision Tree Algorithms.” [Online]. Available:
- <https://github.com/Jinglin-LI/H1B-VisaPrediction-by-Machine-LearningAlgorithm/blob/master/H1B\%20Prediction\%20Research\%20Report.p>
- PredictingCaseStatusof H-1B Visa Petition<https://cseweb.ucsd.edu/classes/wi17/cse258-a/reports/a054.p>

A.SOURCE CODE

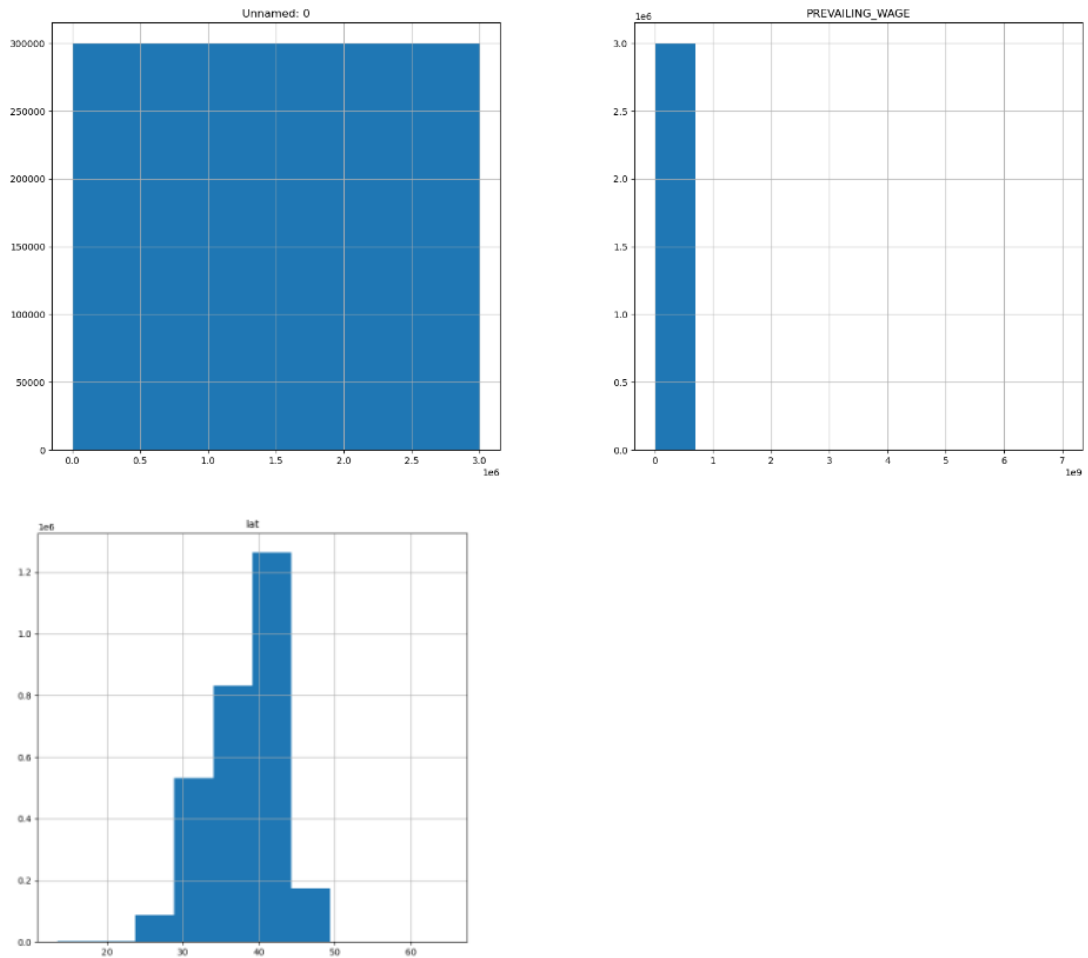
Importing the Libraries

Data Preprocessing

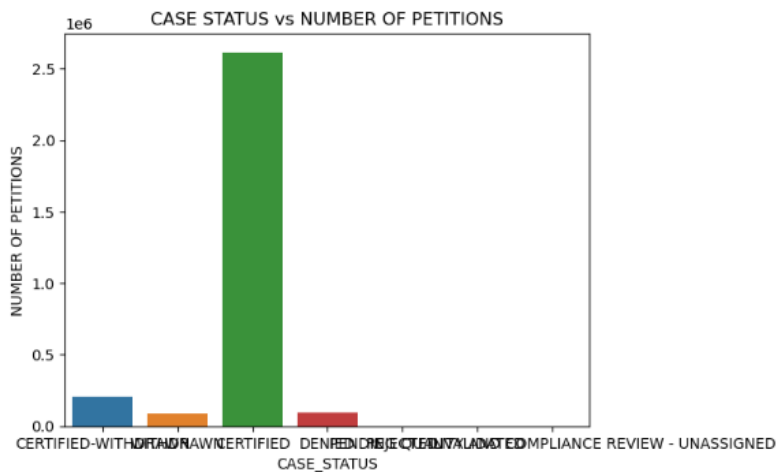
[illegible]

```
In [10]: #Univariate Analysis
#Histogram
data.hist(figsize=(20,30))
```

```
Out[10]: array([[<Axes: title='center': 'Unnamed: 0'>,
<Axes: title='center': 'PREVAILING_WAGE'>],
[<Axes: title='center': 'YEAR'>,
<Axes: title='center': 'lon'>],
[<Axes: title='center': 'lat'>, <Axes: >]], dtype=object)
```

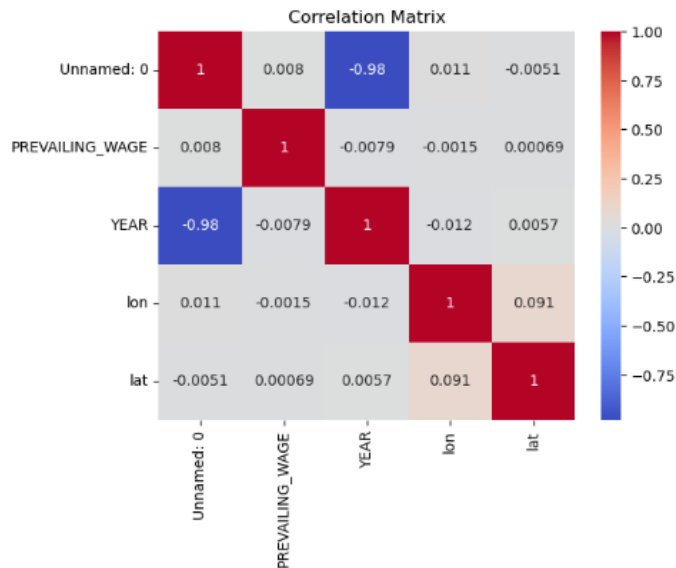


```
In [11]: #Bivariate Analysis
# bar plot
sns.countplot(data=data, x='CASE_STATUS')
plt.title('CASE STATUS vs NUMBER OF PETITIONS')
plt.xlabel('CASE_STATUS')
plt.ylabel('NUMBER OF PETITIONS')
plt.show()
```



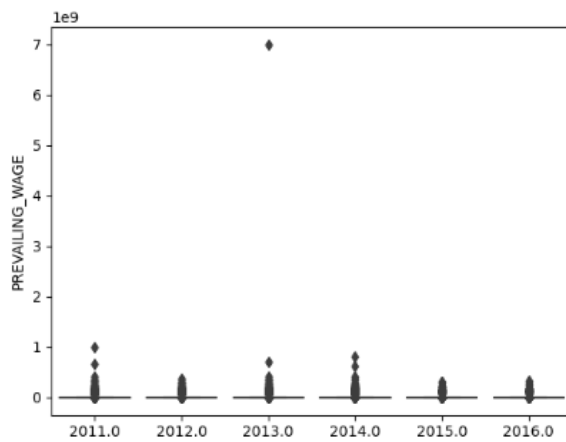
```
In [12]: # Multivariate Analysis
# Heatmap of correlation matrix
corr_matrix = data.corr()
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()

C:\Users\ramya\AppData\Local\Temp\ipykernel_7576\2056294179.py:3: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
  corr_matrix = data.corr()
```

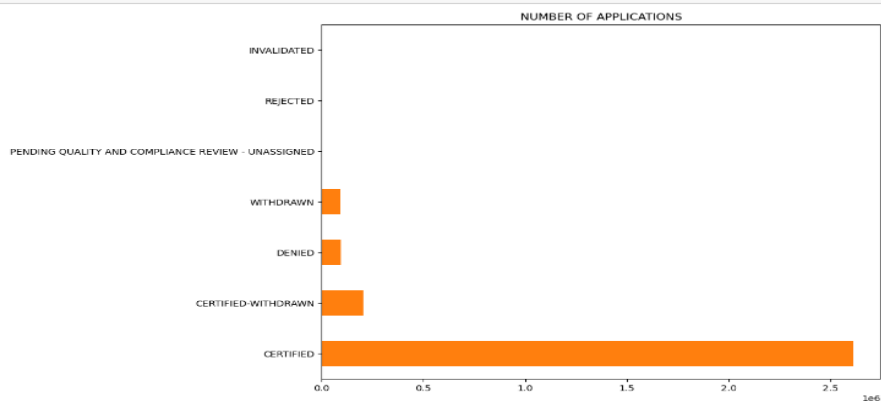


```
In [13]: sns.boxplot(x='YEAR', y='PREVAILING_WAGE', data=data)
```

```
Out[13]: <Axes: xlabel='YEAR', ylabel='PREVAILING_WAGE'>
```



```
In [15]: plt.figure(figsize=(10,8))
data.CASE_STATUS.value_counts().plot(kind='barh', color='C1')
data.sort_values('CASE_STATUS')
plt.title("NUMBER OF APPLICATIONS")
plt.show()
```



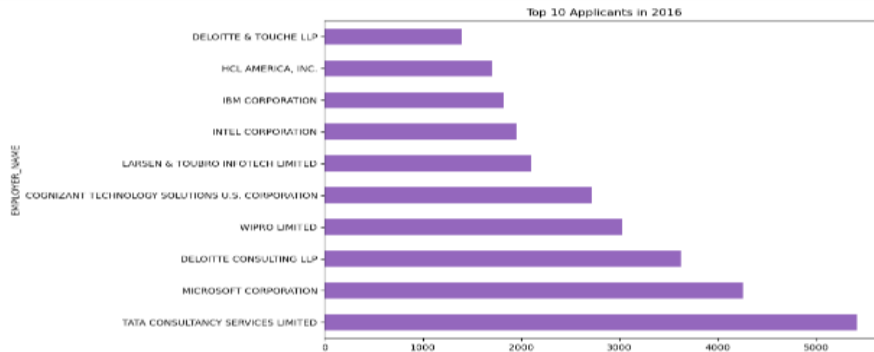
Top 10 applicants in 2011

```

: ##### Top 10 applicants in 2011
plt.figure(figsize=(10,7))

ax1 = data[data["EMPLOYER_NAME"]][data["YEAR"] == 2011].groupby(data["EMPLOYER_NAME"]).count().sort_values(ascending=False).head(10).p
ax1.set_label("")
plt.show()

```



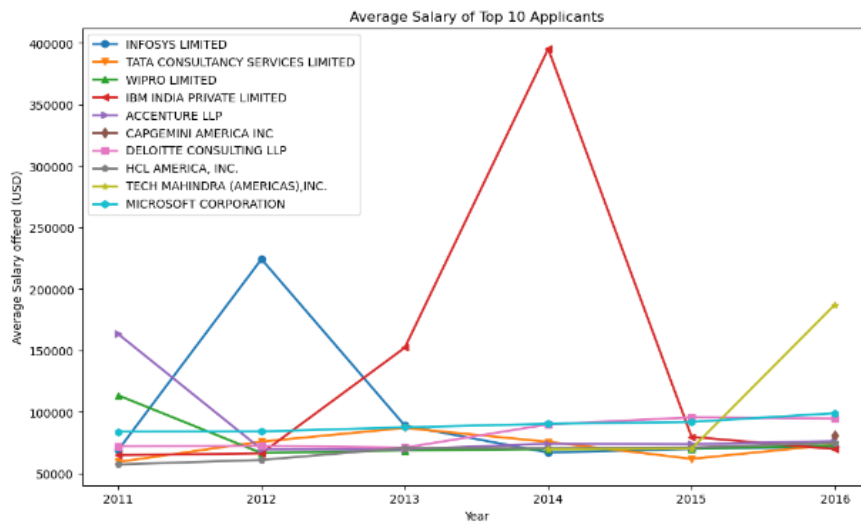
```

]: ##### Average salary of each company
plt.figure(figsize=(12,7))

for company in top_emp:
    tmp = byempyear.mean().loc[company]
    plt.plot(tmp.index.values, tmp["PREVAILING_WAGE"].values, label=company, linewidth=2, marker=markers[top_emp.index(company)])
plt.xlabel("Year")
plt.ylabel("Average Salary offered (USD)")
plt.legend()
plt.title("Average Salary of Top 10 Applicants")
plt.show()

C:\Users\ranya\AppData\Local\Temp\ipykernel_7576\705480231.py:6: FutureWarning: The default value of numeric_only in DataFrameG
rouper.mean is deprecated. In a future version, numeric_only will default to False. Either specify numeric_only or select only
columns which should be valid for the function.
    tmp = byempyear.mean().loc[company]

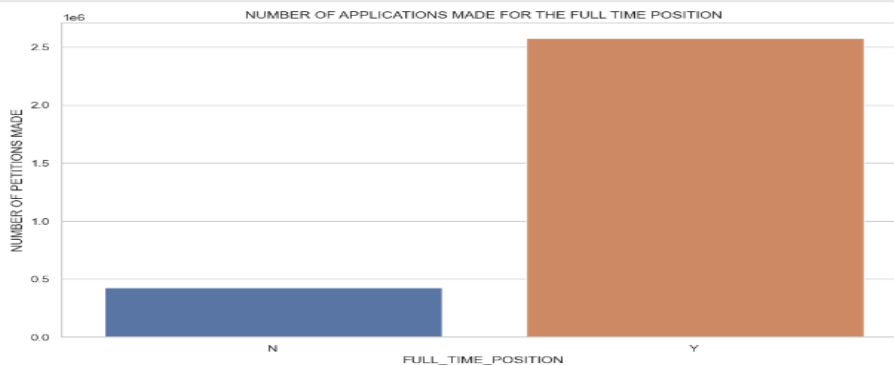
```



```

: plt.figure(figsize=(12,7))
sns.set(style="whitegrid")
g = sns.countplot(x="FULL_TIME_POSITION", data=data)
plt.title("NUMBER OF APPLICATIONS MADE FOR THE FULL TIME POSITION")
plt.ylabel("NUMBER OF PETITIONS MADE")
plt.show()

```



Train-Test Split

Splitting the data into train and test

```
In [49]: 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 [50]: x.columns
Out[50]: Index(['FULL_TIME_POSITION', 'PREVAILING_WAGE', 'YEAR', 'SOC_N'], dtype='object')
```

2) Model Building

Random Forest Classifier

```
In [51]: from sklearn.ensemble import RandomForestClassifier
rf=RandomForestClassifier(n_estimators=10,criterion='entropy',random_state=0)

In [52]: #training the model
rf.fit(x_train,y_train)

Out[52]: RandomForestClassifier(criterion='entropy', n_estimators=10, random_state=0)
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.
```

```
In [53]: #test the model
pred1=rf.predict(x_test)

In [54]: pred1
Out[54]: array(['CERTIFIED', 'CERTIFIED', 'CERTIFIED', ..., 'CERTIFIED',
                'CERTIFIED', 'CERTIFIED'], dtype=object)
```

```
In [55]: x_test
Out[55]:
```

	FULL_TIME_POSITION	PREVAILING_WAGE	YEAR	SOC_N
520002	0	61818.0	2016.0	2
1128791	1	54018.0	2015.0	2
900630	1	30014.0	2015.0	2
1362279	1	66082.0	2014.0	2
1691055	1	54995.0	2014.0	2
...
108212	0	59717.0	2016.0	2
1821115	1	78250.0	2013.0	2
826367	1	62754.0	2015.0	2
573076	0	50502.4	2016.0	2
1164132	1	70533.0	2015.0	2

899974 rows x 4 columns

```
In [56]: y_test
Out[56]:
```

520002	CERTIFIED
1128791	CERTIFIED
900630	CERTIFIED
1362279	CERTIFIED
1691055	CERTIFIED
...	...
108212	CERTIFIED
1821115	CERTIFIED
826367	CERTIFIED
573076	WITHDRAWN
1164132	CERTIFIED

Name: CASE_STATUS, Length: 899974, dtype: object

```
In [57]: from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
accuracy_score(y_test,pred1)
Out[57]: 0.8681361906010618
```

```
In [58]: confusion_matrix(y_test,pred1)
Out[58]: array([[774182, 5554, 3487, 0, 0, 1315],
                [ 54548, 5736, 238, 0, 0, 197],
                [ 26003, 306, 1118, 0, 0, 126],
                [ 1, 0, 0, 0, 0, 0],
                [ 5, 0, 0, 0, 0, 1],
                [ 26130, 684, 167, 0, 0, 272]], dtype=int64)
```

```

In [57]: from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
accuracy_score(y_test, pred1)

Out[57]: 0.8681361906010618

In [58]: confusion_matrix(y_test, pred1)

Out[58]: array([[774182, 5554, 3407, 0, 0, 1315],
 [ 54540, 5736, 238, 0, 0, 197],
 [ 26003, 306, 1110, 0, 0, 126],
 [ 1, 0, 0, 0, 0, 0],
 [ 5, 0, 0, 0, 0, 1],
 [ 26130, 684, 167, 0, 0, 272]], dtype=int64)

In [59]: print(classification_report(y_test, pred1))

C:\Users\ramya\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1344: 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))
C:\Users\ramya\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1344: 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))

              precision    recall  f1-score   support

    CERTIFIED             0.88       0.99       0.93       784458
CERTIFIED-WITHDRAWN       0.47       0.09       0.16        60711
          DENIED          0.23       0.04       0.07       27545
    INVALIDATED          0.00       0.00       0.00           1
PENDING QUALITY AND COMPLIANCE REVIEW - UNASSIGNED  0.00       0.00       0.00           6
          WITHDRAWN       0.14       0.01       0.02       27253

    accuracy             0.87       0.87       0.87      899974
    macro avg             0.29       0.19       0.20      899974
    weighted avg          0.81       0.87       0.82      899974

C:\Users\ramya\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1344: 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 [60]: import pickle
pickle.dump(rf, open('Visarf.pkl', 'wb'))

In [61]: model = pickle.load(open('Visarf.pkl', 'rb'))

In [62]: print(model.predict([[1, 30014.0, 2015.0, 2]]))

['CERTIFIED']

C:\Users\ramya\anaconda3\lib\site-packages\sklearn\base.py:420: UserWarning: X does not have valid feature names, but RandomForestClassifier was fitted with feature names
  warnings.warn(

In [63]: print(model.predict([[1, 70533.0, 2015.0, 2]]))

['CERTIFIED']

C:\Users\ramya\anaconda3\lib\site-packages\sklearn\base.py:420: UserWarning: X does not have valid feature names, but RandomForestClassifier was fitted with feature names
  warnings.warn(

In [64]: print(model.predict([[0, 61818.0, 2016.0, 2]]))

['CERTIFIED']

```
