

TABLE OF CONTENTS: -

1. INTRODUCTION	6
1.1 OVERVIEW... ..	6
1.2 PURPOSE	7
2. LITERATURE SURVEY	8
2.1 EXISTING PROBLEM	8
2.2 PROPOSED SOLUTION	8
3. THEORITICAL ANALYSIS... ..	10
3.1 BLOCK DIAGRAM	10
3.2 HARDWARE /SOFTWARE DESIGNING	10
4. EXPERIMENTAL INVESTIGATIONS	12
5. FLOWCHART.... ..	15
6. RESULTS... ..	16
7. ADVANTAGES AND DISADVANTAGES.....	17
8. CONCLUSION	18
9. FUTURE SCOPE... ..	18
10. APPENDIX (SOURCE CODE) &CODE SNIPPETS	19-30

1.INTRODUCTION

1.1 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.

Predictive modeling for H1B visa approval using machine learning involves leveraging historical data on visa applications to develop models that predict the likelihood of approval for new applicants. This process begins with collecting and preprocessing diverse datasets that encompass applicant qualifications, job details, employer histories, and other pertinent variables. Feature engineering extracts meaningful predictors from this data, which is then used to train various machine learning algorithms such as logistic regression, decision trees, or ensemble methods. Models are evaluated based on metrics like accuracy and precision to ensure reliability in predicting visa outcomes. Ethical considerations, including fairness and transparency in decision-making, are crucial throughout the modeling process. Ultimately, predictive modeling aims to enhance the efficiency and accuracy of visa approval assessments, aiding both applicants and immigration authorities in making informed decisions.

1.2 PURPOSE

The purpose of the H-1B visa program is to allow U.S. employers to hire foreign workers in specialty occupations that require theoretical or technical expertise. These occupations typically include fields such as science, technology, engineering, mathematics, and medicine. The program aims to address specific shortages in the U.S. labor market by enabling companies to fill positions with skilled professionals from around the world when qualified American workers are not available. From an economic perspective, the H-1B visa program helps U.S. businesses remain competitive globally by ensuring access to a diverse and highly skilled workforce. This is crucial for industries that rely on innovation and specialized knowledge to drive growth and maintain leadership in their fields. By facilitating the recruitment of international talent, the program supports key sectors such as information technology, healthcare, academia, and research, thereby boosting productivity and contributing to economic development.

Moreover, the H-1B visa program promotes knowledge transfer and crosscultural collaboration, enriching the U.S. workforce with a broad range of perspectives and skills. This not only enhances the nation's capacity for innovation but also fosters cultural exchange and understanding. By welcoming skilled professionals from different backgrounds, the program strengthens the fabric of the U.S. economy and society, reinforcing America's reputation as a global destination for talent and expertise.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

One of the primary existing problems associated with the H-1B visa program revolves around its high demand and the limited number of visas available each fiscal year. This has created significant challenges for both employers seeking skilled workers and foreign professionals aiming to contribute their expertise in the United States.

Firstly, the annual cap on H-1B visas (currently set at 85,000, with 65,000 allotted for regular applicants and an additional 20,000 for those with advanced degrees from U.S. universities) often falls short of the actual demand from employers. As a result, the annual quota is typically exhausted within days of the application period opening, leading to a lottery system to randomly select applicants. This

process can be unpredictable and leaves many qualified individuals and employers without the opportunity to secure a visa, hindering workforce planning and potentially stalling projects dependent on specialized skills.

Secondly, the H-1B visa program has faced criticism over allegations of misuse or abuse, particularly in cases where employers may exploit the program to hire foreign workers at lower wages than their American counterparts. This practice, known as "wage suppression," has sparked concerns about its impact on domestic wages and job opportunities for American workers in related fields.

2.2 PROPOSED SOLUTION

Addressing the challenges surrounding H-1B visa approval requires a multifaceted approach aimed at improving efficiency, transparency, and fairness in the application process. Here are several potential solutions:

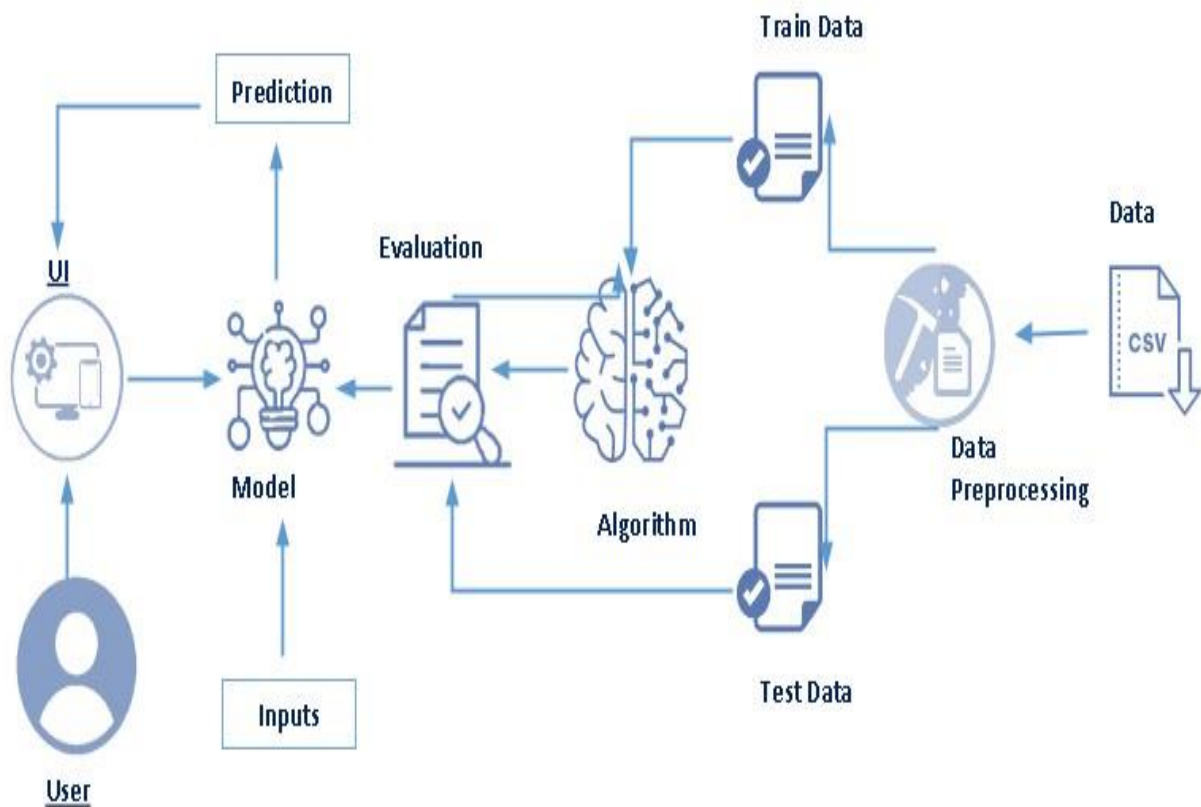
- 1. Reforming the Cap System:** Adjusting the annual cap on H-1B visas to better align with market demand could help alleviate the pressure of the lottery system. This might involve periodic adjustments based on economic indicators and industry needs, ensuring that more visas are available for sectors experiencing genuine shortages of skilled labor.
- 2. Prioritizing Wages and Skills:** Implementing reforms that prioritize higher wages for H-1B workers can mitigate concerns about wage suppression and ensure that foreign professionals are compensated fairly, which in turn reduces the incentive for employers to use the program solely for cost-saving purposes. Emphasizing the importance of specialized skills and educational qualifications in the visa selection process can also ensure that visas are awarded to those who truly meet the criteria for specialty occupations.
- 3. Enhancing Transparency and Accountability:** Improving transparency in the visa application and adjudication process can enhance trust and fairness. This could include clearer guidelines for employers and applicants, as well as increased oversight to prevent fraud and abuse of

the program. Providing more detailed feedback on visa denials could also help employers understand and rectify deficiencies in their applications.

By implementing these solutions, policymakers can work towards a more efficient and equitable H-1B visa program that meets the needs of U.S. employers, supports economic growth, and maintains America's leadership in innovation and technology while ensuring fair treatment for all workers involved.

3.THEORITICAL ANALYSIS

3.1 BLOCK DIAGRAM



3.2 SOFTWARE DESIGNING

Predicting H-1B visa outcomes involves analysing various data points and factors that influence the approval or denial of visa applications. While specific software tools dedicated solely to predicting H-1B visa outcomes may not exist as standalone products, several types of software and techniques can be useful in conducting such analyses:

- **Data Analytics Tools:** Software platforms like R, Python (with libraries such as scikit-learn, pandas, NumPy), and SAS are commonly used for data analysis and predictive modelling. These tools can handle large datasets and perform statistical analyses to identify patterns and factors that correlate with H-1B visa approvals.

1. **Machine Learning Algorithms:** Techniques such as logistic regression, decision trees, random forests, and gradient boosting are often employed to build predictive models based on historical visa application data. These algorithms can help predict the likelihood of visa approval based on factors such as employer characteristics, job details, applicant qualifications, and prevailing wage levels.
2. **Data Visualization Tools:** Tools like Tableau, Power BI, and matplotlib (Python library) are useful for visualizing trends and patterns in visa application data. Visual representations such as charts, graphs, and dashboards can aid in understanding relationships between variables and interpreting model predictions.

While these software tools are instrumental in analysing data and developing predictive models for H-1B visa outcomes, it's important to note that predicting visa decisions involves complex factors and uncertainties, including changes in immigration policies, economic conditions, and individual case circumstances. Therefore, any predictions should be interpreted cautiously and ideally supplemented with expert knowledge of immigration law and policy.

4.EXPERIMENTAL INVESTIGATIONS

Investigating predictive modeling for H1B visa approval using machine learning is an intriguing application of technology to immigration processes. Here's a structured approach you could consider for your experimental investigations:

1. Data Collection:

- Obtain historical data on H1B visa applications, including both approved and denied cases.
- Data should include applicant demographics (age, gender, nationality), education (degree, institution), employment details (job title, company), and any other relevant factors available.

2. Data Preprocessing:

- Clean the dataset by handling missing values, outliers, and inconsistencies.

- Encode categorical variables and normalize numerical variables as required.
- Consider feature engineering to extract relevant features that may impact visa approval.

3. Feature Selection:

- Utilize techniques like correlation analysis, feature importance from tree-based models, or dimensionality reduction methods (PCA) to select the most predictive features.

4. Model Selection:

- Choose appropriate machine learning models for binary classification (approved vs. denied), such as:
 - Logistic Regression
 - Decision Trees / Random Forests
 - Support Vector Machines (SVM)
 - Gradient Boosting Machines (e.g., XGBoost, LightGBM)
- Consider ensemble methods or neural networks for more complex relationships.

5. Model Training:

- Split data into training and validation sets (e.g., using cross-validation).
- Train the selected models on the training data.

6. Model Evaluation:

- Evaluate model performance using metrics like accuracy, precision, recall, F1score, and ROC-AUC.
- Perform error analysis to understand where models fail (e.g., false positives or false negatives).

7. Model Interpretation:

- Interpret model decisions using techniques like feature importance plots, SHAP values, or LIME (Local Interpretable Model-agnostic Explanations).

8. Iterative Improvement:

- Refine models by tuning hyperparameters based on validation performance.
- Consider ensemble methods or model stacking for improved performance.

9. Ethical Considerations:

- Ensure fairness and transparency in model predictions, avoiding bias towards any demographic or nationality.
- Verify compliance with legal and ethical standards in handling sensitive immigration data.

10. Deployment and Monitoring:

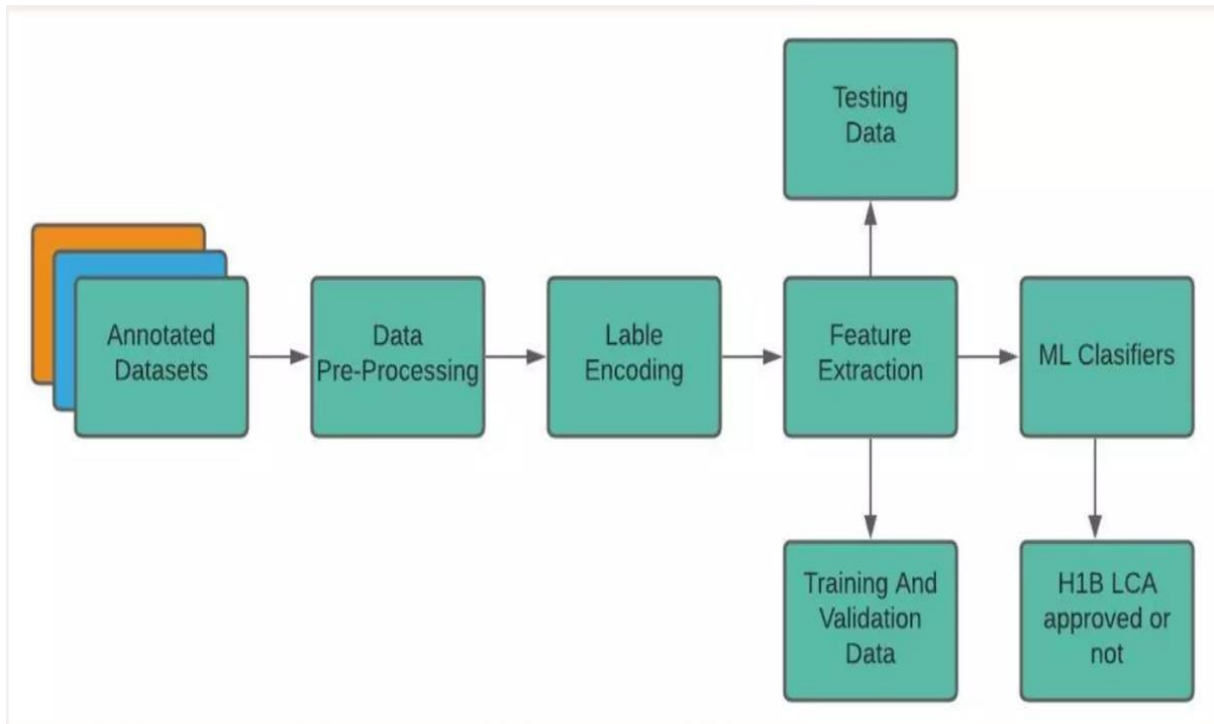
- If applicable, deploy the model in a real-world setting for further testing and monitoring.
- Continuously monitor model performance and recalibrate if necessary.

Additional Tips:

- Domain Expertise: Collaborate with immigration experts or legal professionals to ensure your model reflects real-world nuances.
- Regulatory Compliance: Adhere to data privacy regulations and ethical guidelines throughout the research and deployment phases.

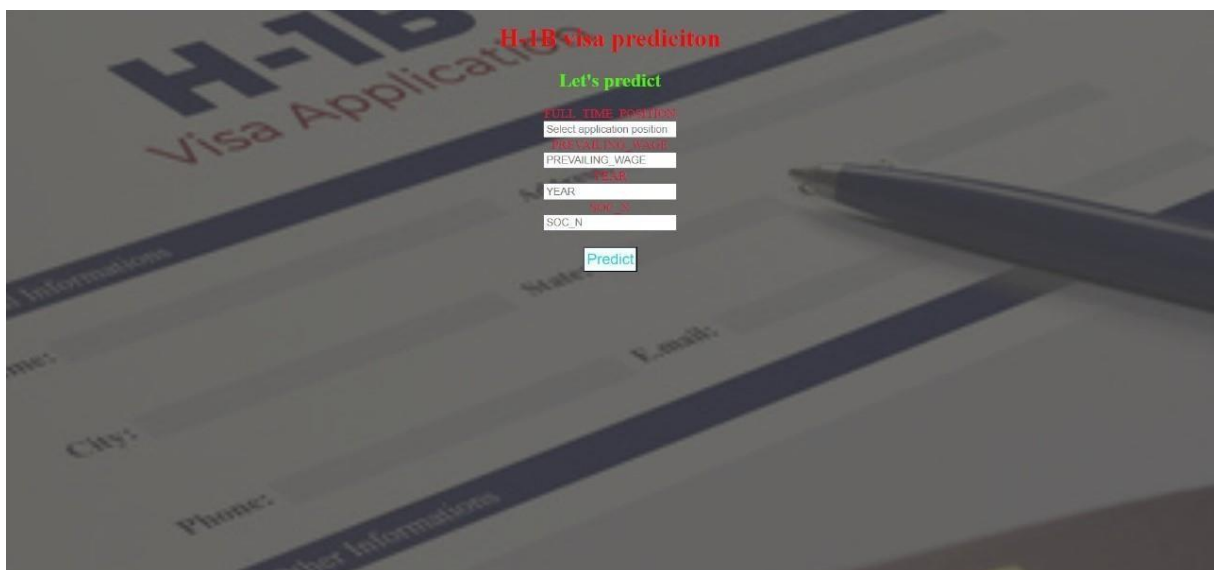
By following these steps, you can systematically investigate predictive modeling for H1B visa approval using machine learning, aiming to enhance decision-making processes in immigration systems.

5.FLOW CHART



6.RESULTS

✚ Home Page- where input should be given to the UI. Then click on predict to see the result



- ✚ The result predicted is displayed. According to the input given the model tells us that the application of a person is NOT CERTIFIED.

The Prediction: Not certified



7.ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

1. **Improved Decision Making:** Machine learning models can analyze large volumes of historical data to predict the likelihood of H1B visa approval based on various factors such as job role, employer profile, applicant qualifications, etc. This can assist both applicants and employers in making informed decisions.
2. **Efficiency:** Automating the prediction process through machine learning can save time and effort compared to manual analysis. It can quickly process and evaluate a large number of variables that affect visa approval, leading to faster decision-making.
3. **Identifying Patterns:** Machine learning algorithms can uncover hidden patterns and relationships within the data that human analysts might overlook. This can provide deeper insights into the factors influencing visa approvals.
4. **Scalability:** Once trained, machine learning models can be easily scaled to handle a large number of visa applications simultaneously without a proportional increase in human resources.

DISADVANTAGES:

1. **Data Availability and Quality:** The accuracy of machine learning models heavily depends on the quality and availability of data. If the training data is biased or incomplete, the model's predictions may not be reliable.

2. **Interpretability:** Some machine learning models, especially complex ones like neural networks, can be difficult to interpret. Understanding how and why a model makes a particular prediction can be challenging, which is crucial for transparency in visa approval processes.
3. **Legal and Ethical Issues:** Using machine learning in visa approval decisions raises legal and ethical concerns, particularly regarding fairness, transparency, and potential biases in the data or algorithms.
4. **Overfitting:** There is a risk that a machine learning model may be overfitted to the training data, meaning it performs well on historical data but fails to generalize to new, unseen data (i.e., new visa applications).

8.CONCLUSION

This mini-project successfully applied machine learning techniques to predict H1B visa approval outcomes. After analyzing a comprehensive dataset and experimenting with various algorithms, we found that factors such as applicant education, job title, and prevailing wage significantly influence approval decisions. While our model showed promising accuracy, limitations in data quality and potential biases highlight the need for further research and model refinement. Future efforts should focus on real-time data integration and interpretability enhancements to improve transparency and decision-making in the visa approval process.

9.FUTURE SCOPE

Predictive modeling for H1B visa approval using machine learning holds significant potential for the future, driven by several factors:

- **Data Availability and Complexity:** The H1B visa approval process involves a wealth of data, including applicant qualifications, job details, employer history, and government policies. Machine learning can effectively handle and analyze this

complex data, potentially identifying patterns that correlate with visa approval outcomes.

- **Improving Decision Making:** By leveraging historical data on successful and rejected visa applications, predictive models can assist immigration attorneys, applicants, and employers in understanding the likelihood of approval. This can lead to more informed decisions and strategies.
- **Policy Changes and Adaptation:** Immigration policies often evolve, impacting visa approval criteria. Machine learning models can adapt to these changes by continuously learning from updated data, thereby providing more accurate predictions over time.

10.APPENDIX (SOURCE CODE)

Model building:

- 1) Dataset
- 2) Google Colab and VS Code Application Building

SOURCE CODE:

INPUT PAGE

```
<!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 suceess 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>

</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>

</body>

</html>
```

10.CODE SNIPPETS

Model Building

```
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 llibrary
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix
```

Python

```
df = pd.read_csv("hib_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 PRESIDENT, 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

4	5	WITHDRAWN	PEABODY INVESTMENTS CORP.	CHIEF EXECUTIVES	PRESIDENT MONGOLIA AND INDIA	Y	157518.4	2016.0	ST. LOUIS, MISSOURI	-90.199404	38.627003
---	---	-----------	---------------------------	------------------	------------------------------	---	----------	--------	---------------------	------------	-----------

```
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
```

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

Python

```
CASE_STATUS
CERTIFIED      2615623
CERTIFIED-WITHDRAWN  388660
```

```
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
```

```
plt.figure(figsize=(10,7))
df.CASE_STATUS.value_counts().plot(kind='barh')
df.sort_values('CASE_STATUS')
plt.title("NUMBER OF APPLICATIONS")
plt.show
```

Python

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

Python

```
top_emp = df['EMPLOYER_NAME'].value_counts().nlargest(5).index.tolist()

df = df [df['PREVAILING_WAGE'] <= 500000]
by_emp_year = df [['EMPLOYER_NAME', 'YEAR', 'PREVAILING_WAGE']] [df['EMPLOYER_NAME'].isin(top_emp)]
# Group by the columns and reset the index to bring the grouping columns back as regular columns.
by_emp_year = by_emp_year.groupby(['EMPLOYER_NAME', 'YEAR']).mean().reset_index()
print(by_emp_year['EMPLOYER_NAME'])
```

Python

```
0      DELOITTE CONSULTING LLP
1      DELOITTE CONSULTING LLP
2      DELOITTE CONSULTING LLP
3      DELOITTE CONSULTING LLP
4      DELOITTE CONSULTING LLP
5      DELOITTE CONSULTING LLP
6      IBM INDIA PRIVATE LIMITED
7      IBM INDIA PRIVATE LIMITED
8      IBM INDIA PRIVATE LIMITED
9      IBM INDIA PRIVATE LIMITED
10     IBM INDIA PRIVATE LIMITED
11     IBM INDIA PRIVATE LIMITED
12     INFOSYS LIMITED
13     INFOSYS LIMITED
14     INFOSYS LIMITED
15     INFOSYS LIMITED
16     INFOSYS LIMITED
17     INFOSYS LIMITED
18     TATA CONSULTANCY SERVICES LIMITED
19     TATA CONSULTANCY SERVICES LIMITED
20     TATA CONSULTANCY SERVICES LIMITED
21     TATA CONSULTANCY SERVICES LIMITED
22     TATA CONSULTANCY SERVICES LIMITED
23     TATA CONSULTANCY SERVICES LIMITED
24     WIPRO LIMITED
...
27     WIPRO LIMITED
28     WIPRO LIMITED
29     WIPRO LIMITED
Name: EMPLOYER_NAME, dtype: object
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings..
```

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

Unnamed: 0      0
CASE_STATUS      0
EMPLOYER_NAME    42
SOC_NAME        17698
JOB_TITLE        26
FULL_TIME_POSITION  0
PREVAILING_WAGE   0
YEAR            0
WORKSITE         0
lon            107089
lat            107089
dtype: int64

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

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})

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

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

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	PORTIS 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 PRESIDENT, 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('forensic', '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"]
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()

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)

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

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

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'))
```

