# H1-B VISA PREDICTION

PRESENTED BY-
AAKARSH KUMAR SINGH-21052721
PRATIKSHYA BEHERA-2105050
MANSHA PATRA-21052769
SHREYA ROY-21051764


# INTRODUCTION

The H1-B visa program is a key component of the United States immigration system, allowing employers to hire foreign workers in specialty occupations. These occupations typically require a high level of specialized knowledge and at least a bachelor's degree or its equivalent. The H1-B visa datasets contain valuable information regarding visa applications, providing insights into various aspects of the program, including employer details, job positions, salaries, locations, prevailing wages, and visa outcomes. Analyzing these datasets can offer valuable insights into trends, patterns, and dynamics within the labor market, as well as shed light on the impact of immigration policies and regulatory changes.

# PROJECT OVERVIEW

Analysed US H1B dataset, which contains insights into various aspects of the H1B visa program like job titles, salaries, employer locations, etc.

Applied data analysis and machine learning techniques to extract meaningful trends and patterns from the dataset, contributing to informed decision-making and a deeper understanding of H1B visa landscape

Upon this analysis,we are making a model to predict the status of the applied H1-b visa.

# DATA PREPROCESSING

# MISSING VALUES

```
df['PREVAILING_WAGE'].fillna(df['PREVAILING_WAGE'].media
n(), inplace=True)
median_year = df['YEAR'].median()
df['YEAR'].fillna(median_year, inplace=True)
```

```
df['CASE_STATUS'].fillna(df['CASE_STATUS'].mode()[0],
inplace=True)
df['EMPLOYER_NAME'].fillna(df['EMPLOYER_NAME'].mode()
[0], inplace=True)
df['JOB_TITLE'].fillna(df['JOB_TITLE'].mode()[0], inplace=True)
df['FULL_TIME_POSITION'].fillna(df['FULL_TIME_POSITION'].m
ode()[0], inplace=True)
df['SOC_NAME'].fillna(df['SOC_NAME'].mode()
[0],inplace=True)
```

## EXPLANATION

****Median for Numeric Columns:****
**- Robust to outliers, providing a better representation of central tendency in skewed distributions.**
**- Less sensitive to extreme values, making it a suitable choice for data with outliers.**
**- Preserves the overall distribution of the data while handling missing values.**

****Mode for Categorical Columns:****
**- Represents the most frequently occurring value in the dataset.**
**- Ensures that imputed values align with the majority of observations in the categorical column.**
**- Preserves the distribution of categorical variables, aiding in maintaining data integrity and accuracy in subsequent analyses.**

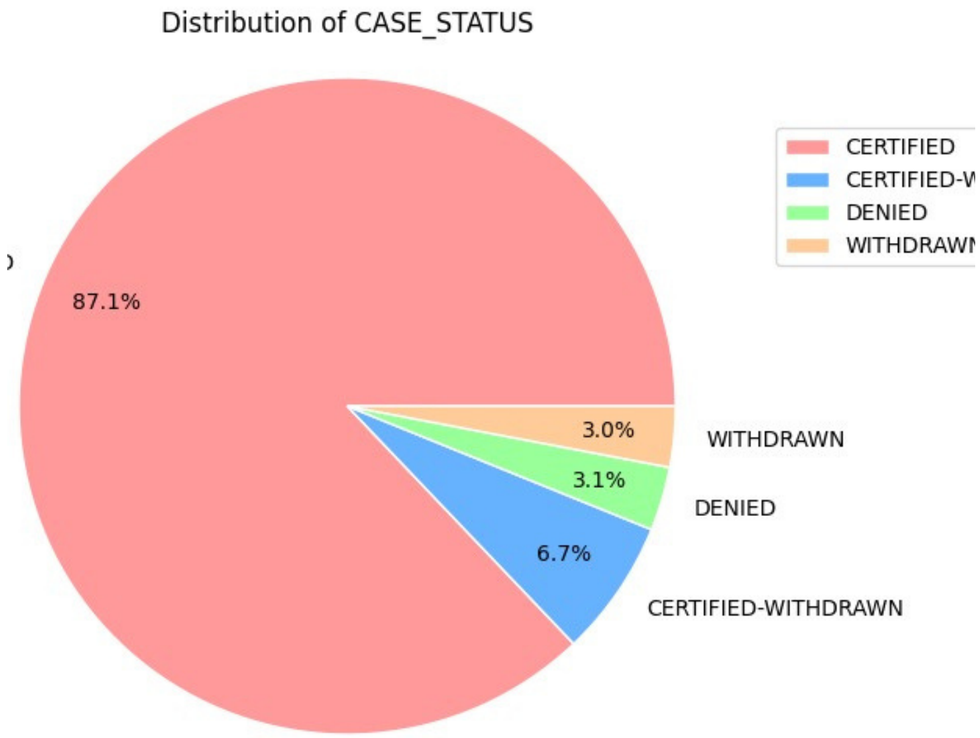
# OUTLIERS


```
total = df['PREVAILING_WAGE'].count()
total

numerical_columns = ['PREVAILING_WAGE']


Q1 = df[numerical_columns].quantile(0.25)
Q3 = df[numerical_columns].quantile(0.75)
Q2 =df[numerical_columns].median()
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

print("Q1:",Q1)
print("Q3:",Q3)
print("Q2:",Q2)
plt.figure(figsize=(10, 6))
sns.boxplot(x='PREVAILING_WAGE', data=df,showfliers=True, color='skyblue',flierprops=dict(marker='o',
markerfacecolor='red', markersize=5))
plt.title('Boxplot of Prevailing Wage')
plt.xlabel('Prevailing Wage')
plt.xlim(left=0, right=df['PREVAILING_WAGE'].quantile(0.99))
#plt.ylabel('Job Title')
plt.grid(True)
plt.show()
```

# DATA ANALYSIS

- Distribution of visa petitions by case status
- Top companies filing the most visa applications
- Top 20 states filing highest visa petitions
- Top employers granting the maximum prevailing wages
- Distribution of visa petitions by year
- Top job titles with the most visa petitions
- Fraction of full-time and part-time workers

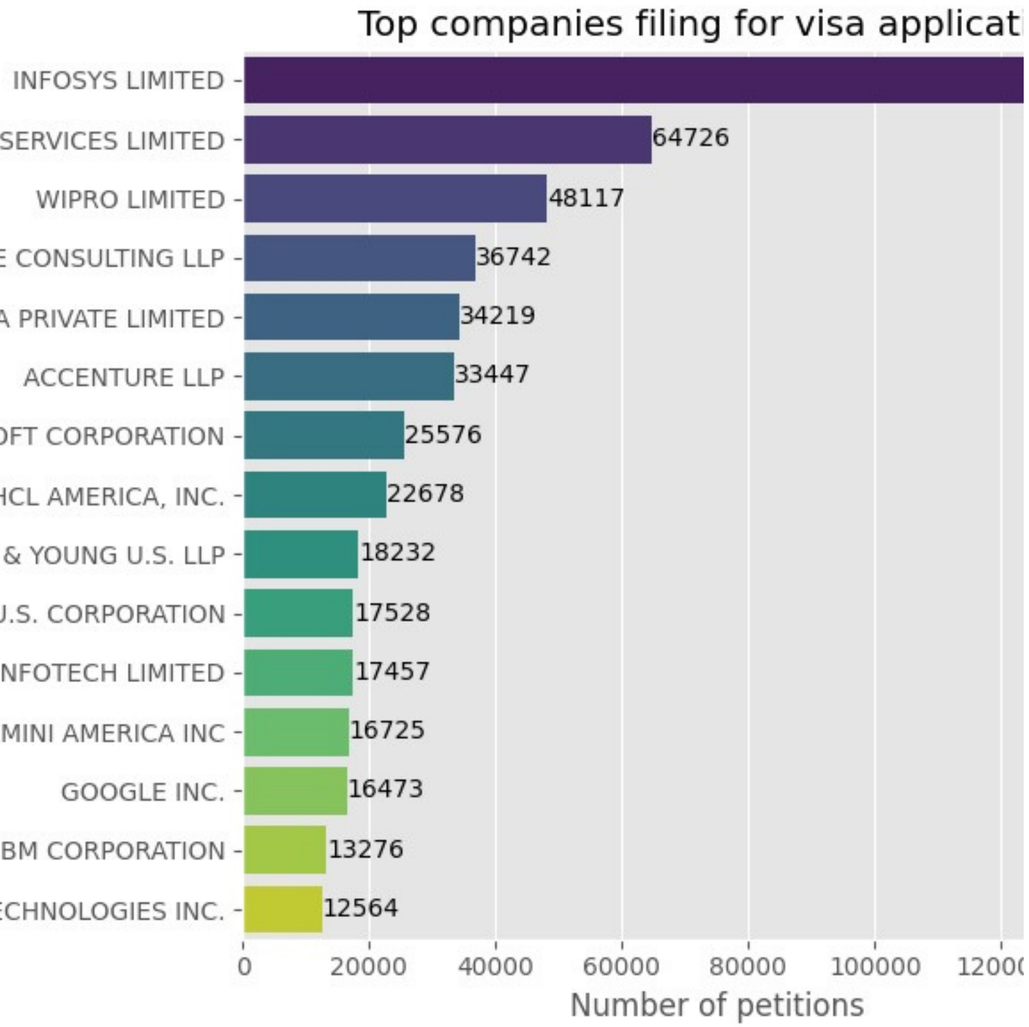
# DISTRIBUTION OF CASE STATUS




## CODE

```
colors = ['#FF9999', '#66B2FF', '#99FF99', '#FFCC99']

plt.figure(figsize=(8, 6), dpi=100)
c, _, _ = plt.pie(x=cs[:4], labels=cs.index[:4], autopct='%.1f%%', colors=colors,
pctdistance=0.8)
plt.title('Distribution of CASE_STATUS')

plt.axis('equal')

for wedge in c:
    wedge.set_edgecolor('white')  # Add white border to pie slices

plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.legend(c, cs.index, loc='upper right', bbox_to_anchor=(1.3, 0.9))

plt.show()
```

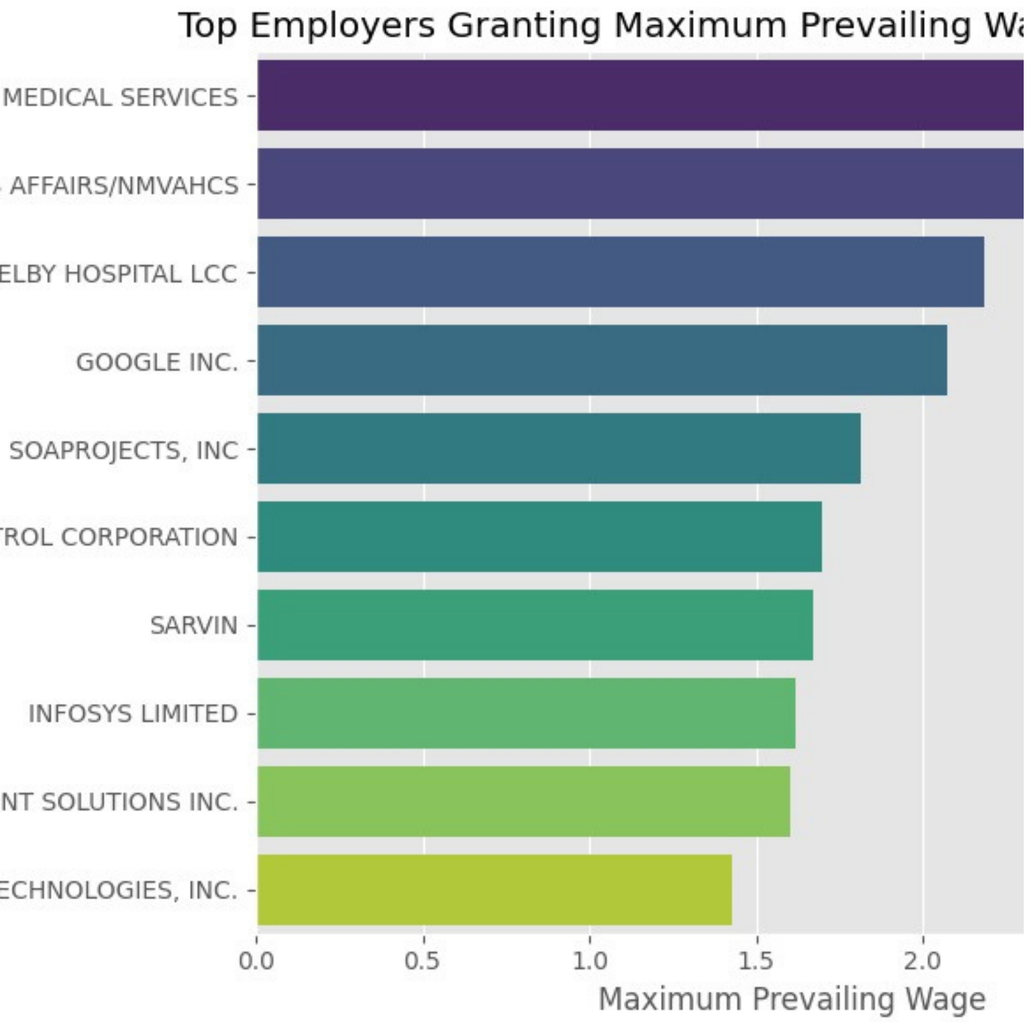
# TOP COMPANIES FILING FOR VISA APPLICATION




## CODE

```
plt.style.use('ggplot')
plt.figure(figsize=(10,6),dpi=100)
barplot = sns.barplot(x=top_companies.values,y=top_companies.index,orient='h',hue
=top_companies.index,palette='viridis')
plt.title("Top companies filing for visa application")
plt.xlabel("Number of petitions")
plt.ylabel("Companies")


for i, v in enumerate(top_companies.values):
    barplot.annotate(str(v), xy=(v, i), xytext=(v + 10, i), color='black', va='center',ha='left')
plt.xlim(right=max(top_companies.values) + 19349)
plt.tight_layout()
plt.show()
```

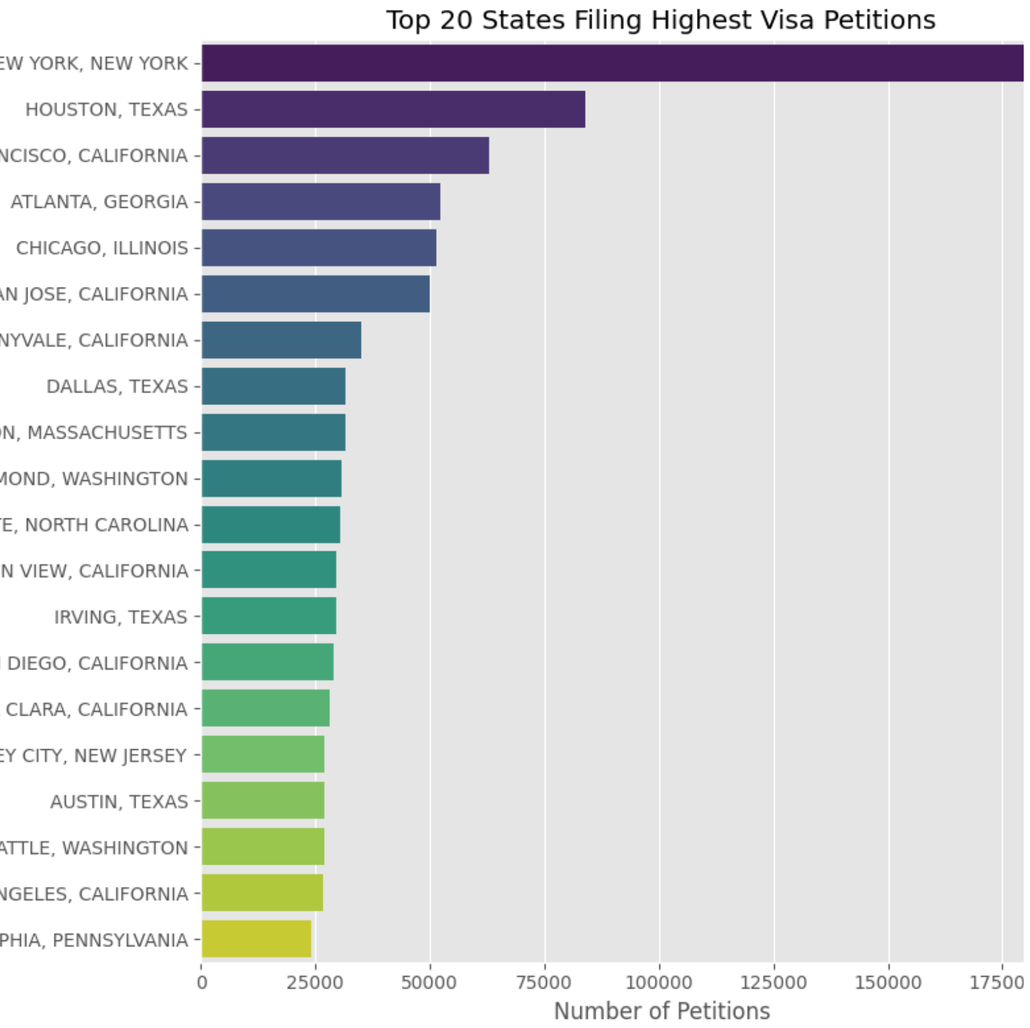
# TOP EMPLOYERS GRANTING MAXIMUM PREVAILING WAGES




## CODE

```
plt.figure(figsize=(10, 6), dpi=100)
barplot = sns.barplot(x=top_emp_max_wages['PREVAILING_WAGE'],
y=top_emp_max_wages.index, orient='h',hue=top_emp_max_wages.index, palette='viridis')
plt.title("Top Employers Granting Maximum Prevailing Wages (Certified Cases)")
plt.xlabel("Maximum Prevailing Wage")
plt.ylabel("Employer Name")
plt.tight_layout()
plt.show()
```

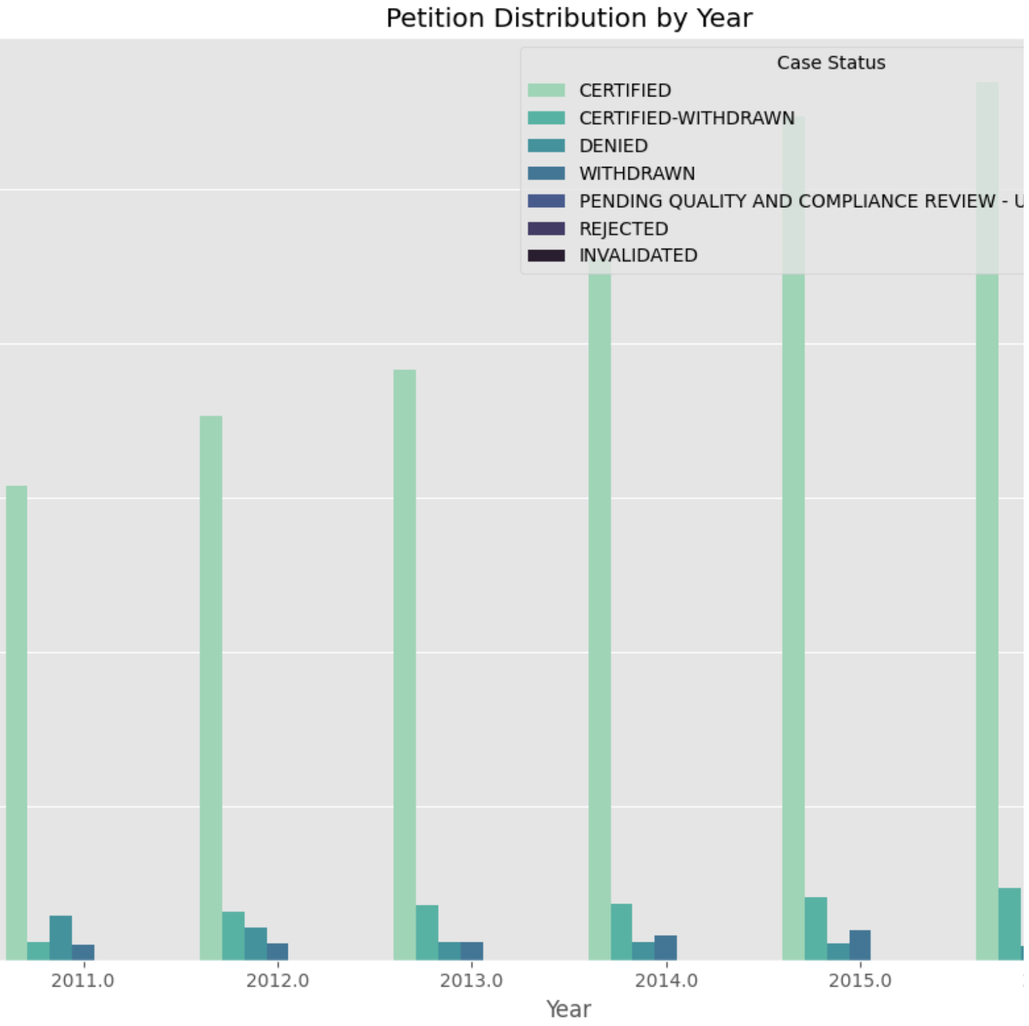
# TOP 20 STATES FILING HIGHEST VISA PETITIONS




## CODE

```
top_states_petitions = df['WORKSITE'].value_counts().nlargest(20)


plt.figure(figsize=(10, 8), dpi=100)
barplot = sns.barplot(x=top_states_petitions.values, y=top_states_petitions.index, orient='h',
                      hue= top_states_petitions.index,palette='viridis')
plt.title("Top 20 States Filing Highest Visa Petitions")
plt.xlabel("Number of Petitions")
plt.ylabel("State")


plt.tight_layout()
plt.show()
```

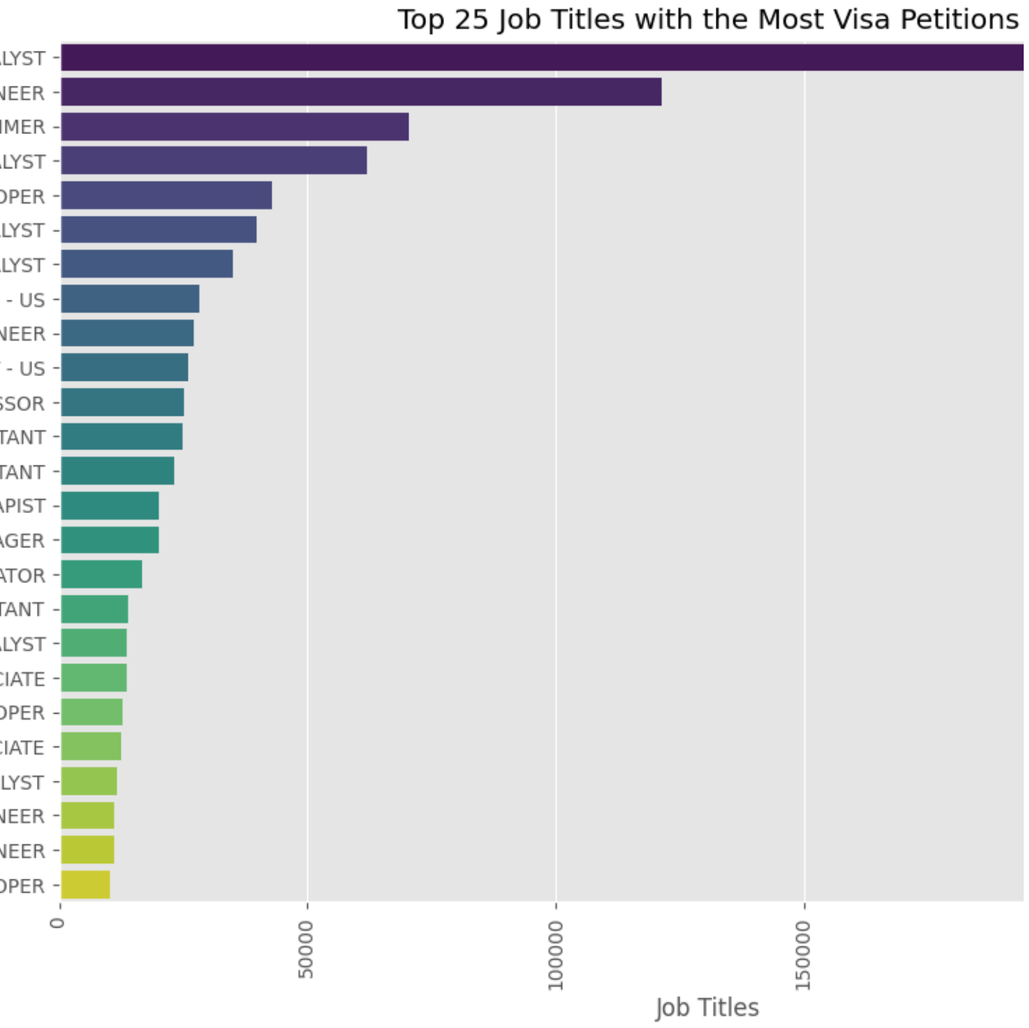
# PETITION DISTRIBUTION BY YEAR




## CODE

```
plt.figure(figsize=(10, 8), dpi=100)
barplot = sns.barplot(data=petition_by_year,x='YEAR',y='COUNT', hu
'CASE_STATUS',palette='mako_r')
plt.title("Petition Distribution by Year")
plt.ylabel("Number of Petitions")
plt.xlabel("Year")

plt.legend(title='Case Status',loc='upper right')
plt.tight_layout()
plt.show()
```

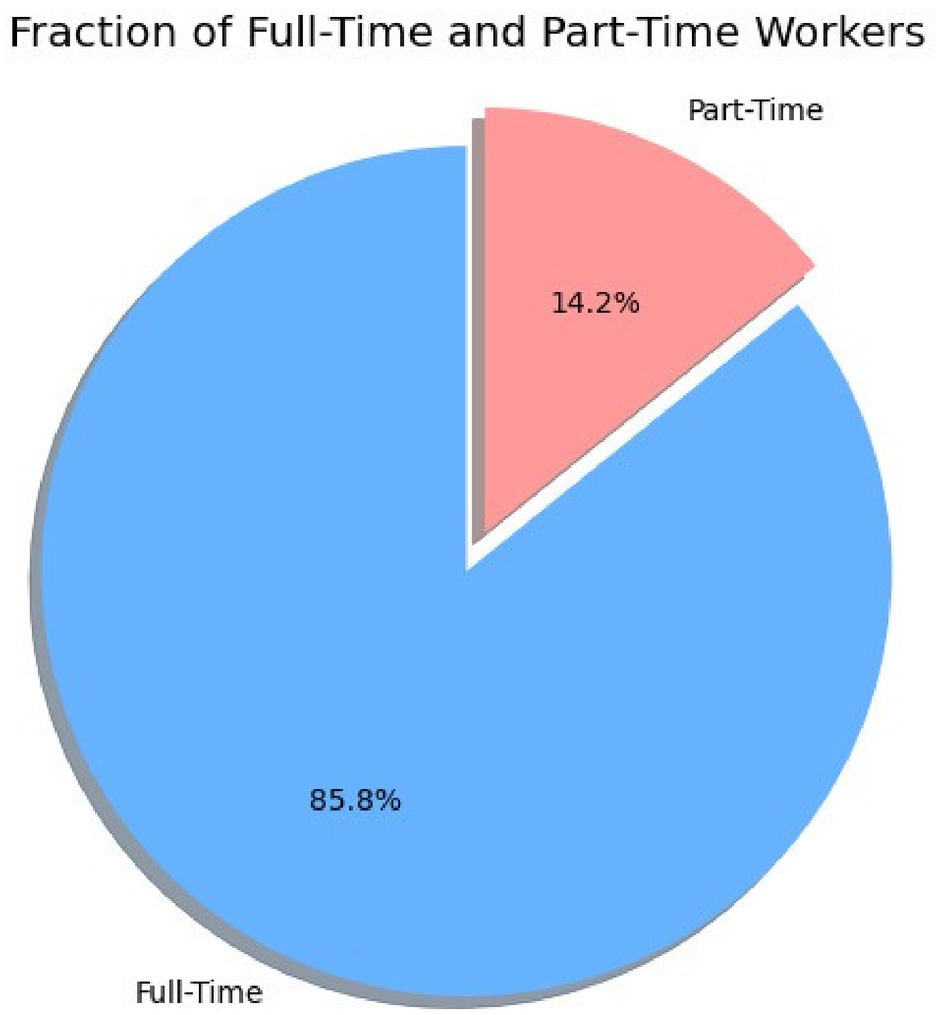
# TOP 25 JOB TITLES WITH THE MOST VISA PETITIONS




## CODE

```
plt.figure(figsize=(12, 8), dpi=100)
barplot = sns.barplot(y=job_titles.index, x=job_titles.valu
orient='h',hue=job_titles.index,palette='viridis')
plt.title("Top 25 Job Titles with the Most Visa Petitions")
plt.xlabel("Job Titles")
plt.ylabel("Number of Petitions")
plt.xticks(rotation=90)
plt.show()
```

# FRACTION OF FULL TIME AND PART TIME WORKERS




## CODE

```
fractions = [fraction_full_time, fraction_part_time]
labels = ['Full-Time', 'Part-Time']
colors = ['#66B2FF', '#FF9999']
explode = [0.1, 0]

plt.figure(figsize=(6, 6))
plt.pie(fractions, labels=labels, colors=colors, explode=explode, autopct='%1.1f%%',
startangle=90, shadow=True)
plt.title('Fraction of Full-Time and Part-Time Workers')
plt.axis('equal')  # Equal aspect ratio ensures a circular pie

plt.show()
```

# PREDICTION H1- B STATUS USING ML CLASSIFIER

**MACHINE LEARNING (ML) is category of algorithm that allows software applications to become more accurate in predicting outcomes without being explicitly programmed .The basic premise of machine learning is to build algorithms that can receive input data and use statistical analysis to predict output while updating outputs as new data becomes available.**

# LOGISTIC REGRESSION AS ML CLASSIFIER

Logistic regression is a machine learning algorithm used for binary classification tasks, making it suitable for predicting visa outcomes like approval or denial. In brief:

- Data Preparation: Select relevant features and preprocess the dataset.
- Model Training: Train the logistic regression model to estimate the probability of visa outcomes based on input features.
- Model Evaluation: Assess the model's performance using metrics like accuracy, precision, recall, F1-score, or AUC.
- Feature Importance: Analyze coefficients to understand the importance of features in predicting visa outcomes.
- Hyperparameter Tuning: Fine-tune model parameters to improve performance and prevent overfitting.
- Model Interpretability: Logistic regression offers inherent interpretability, allowing stakeholders to understand the factors driving visa approval decisions.
- Deployment: Deploy the logistic regression classifier in a production environment for real-time prediction of visa outcomes.

Overall, logistic regression provides a transparent and interpretable solution for visa prediction based on applicant and employer characteristics.

# MODEL CODE

```
Logistic Regression Model Accuracy: 0.8712089420008926
                                                precision    recall  f1-score   support

                                     CERTIFIED       0.87      1.00      0.93    522884
                           CERTIFIED-WITHDRAWN       1.00      0.00      0.00     40735
                                        DENIED       0.89      0.01      0.03     18734
PENDING QUALITY AND COMPLIANCE REVIEW - UNASSIGNED       1.00      0.00      0.00         1
                                      REJECTED       1.00      0.00      0.00         2
                                     WITHDRAWN       1.00      0.00      0.00     18136

                                      accuracy                           0.87    600492
                                     macro avg       0.96      0.17      0.16    600492
                                  weighted avg       0.88      0.87      0.81    600492

Predicted Visa Acceptance Rate (Logistic Regression): ['CERTIFIED']
```

# THANK YOU