Final Project

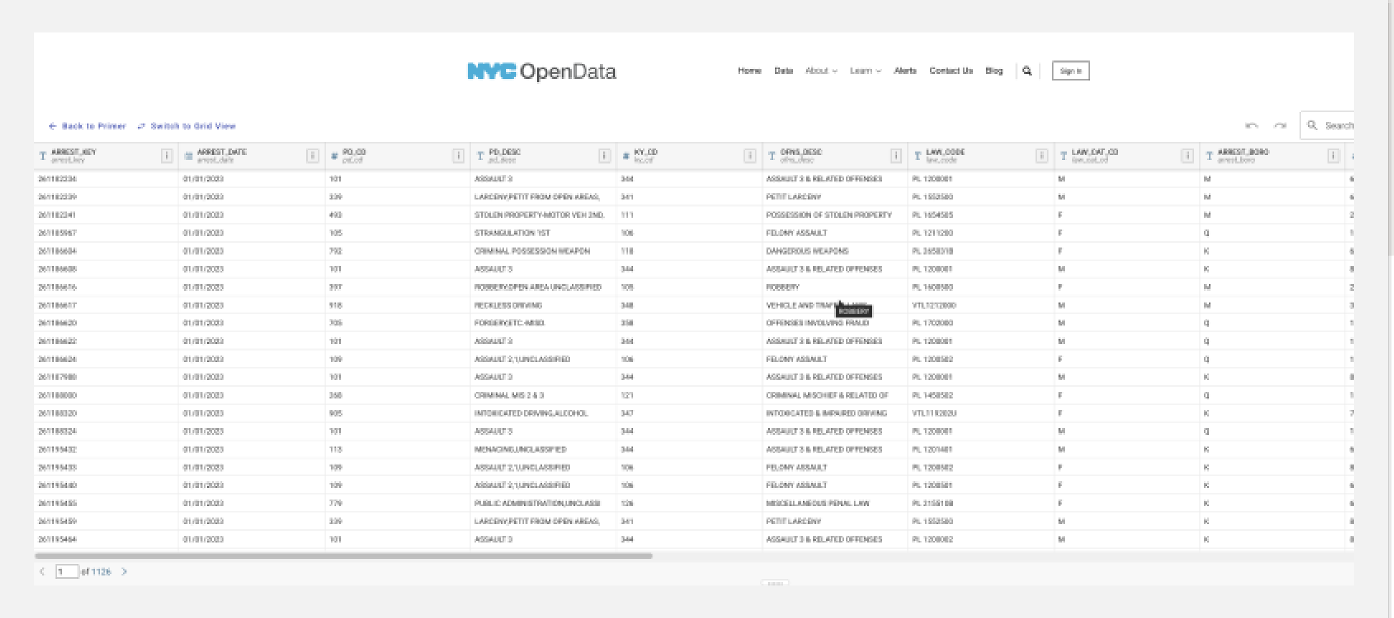
Instructor – Daya Rudhramoorthi
28 October 2023

Team Members: Devansh Hukmani , Yuning Chen

Real-World Problems Addressed by Big Data

- Healthcare Management
- Fraud Detection
- Customer Relationship Management
- Environmental Sustainability
- Cybersecurity
- Education Enhancement
- Supply Chain Management
- Energy Management

Dataset selected



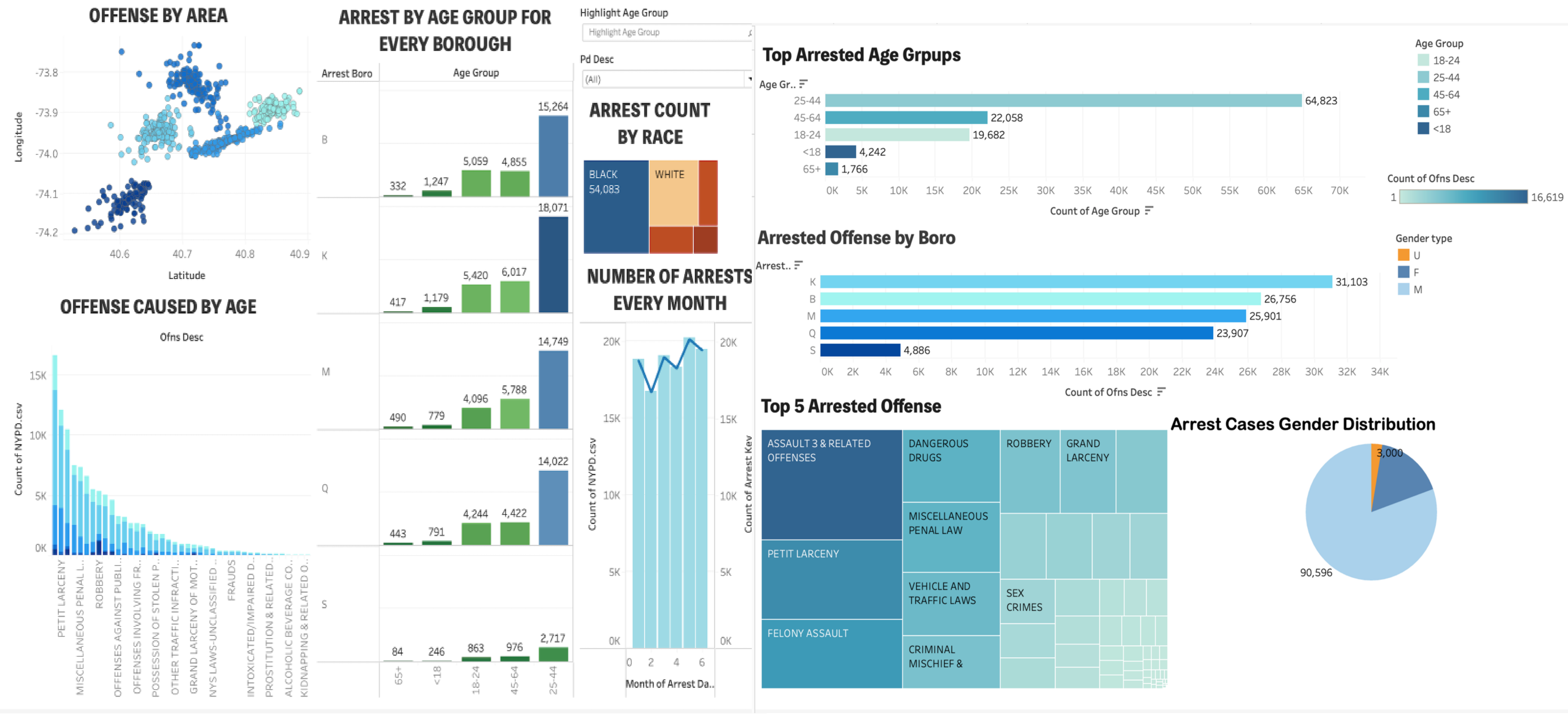
- This dataset comprising over 113,000 rows and 19 columns. It represents a comprehensive breakdown of every arrest conducted by the New York City Police Department (NYPD) during this year.
- Data source: https://data.cityofnewyork.us/Public-Safety/NYPD-Arrest-Data-Year-to-Date-/uip8-fykc

Purpose of analyzing

- In the ever-evolving landscape of law enforcement, the analysis of arrest data plays a pivotal role in understanding the dynamics of policing within a major metropolis like New York City.

- This data can be used by the public to explore the nature of police enforcement activity.

- leverage this rich dataset to explore and analyze the patterns and trends within NYPD's arrest activities. Answering our business question: **How do the distribution of arrests across New York City boroughs correlate with the types of crimes committed in different age groups and genders, can we identify patterns and variations that can inform regulated suggestions for the NYPD?**



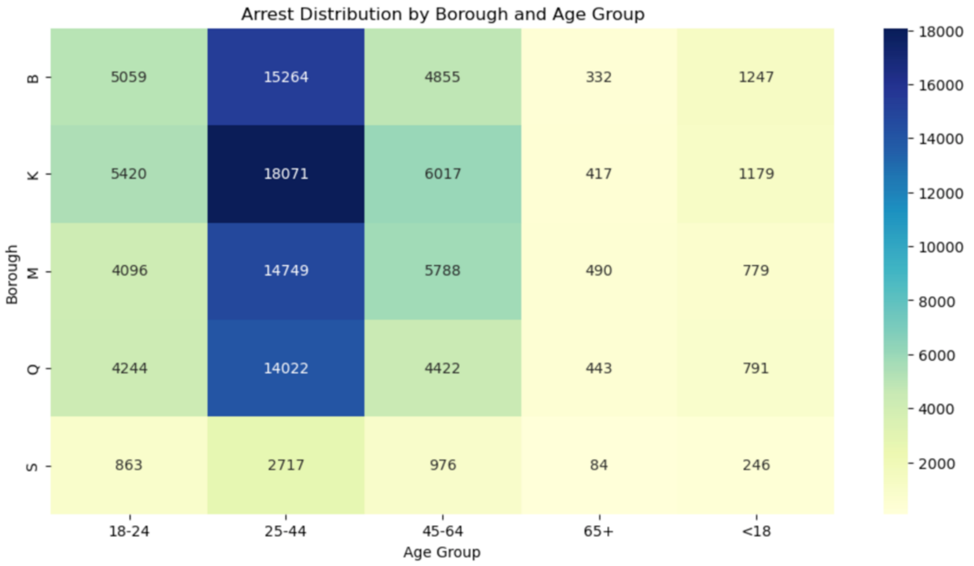
The methodology used to analyze the data: Tableau

EDA

By creating the graph of Crime Distribution by Borough and Age Group. We can specifically derive the highest and lowest age groups for crime in each Borough to assist the NYPD in providing the appropriate prevention techniques for individual Boroughs. And most of the crimes committed in Kings are Arson, in Manhattan are Dangerous Drugs

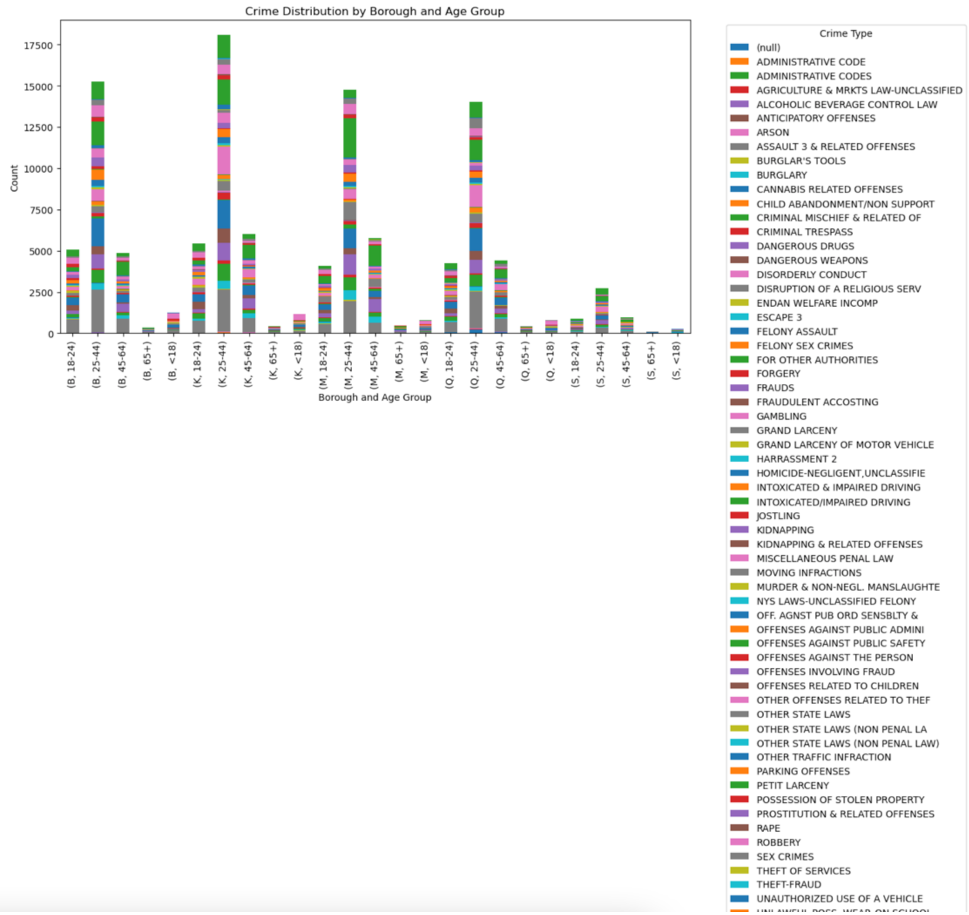
In [7]:
```
# Group data by "ARREST_BORO" and "AGE_GROUP" and calculate counts
arrests_by_borough_age = nyp.groupby(['ARREST_BORO', 'AGE_GROUP']).size().unstack()

# Create a bar chart
plt.figure(figsize=(12, 6))
sns.heatmap(arrests_by_borough_age, cmap="YlGnBu", annot=True, fmt="d")
plt.title("Arrest Distribution by Borough and Age Group")
plt.xlabel("Age Group")
plt.ylabel("Borough")
plt.show()
```



By creating the graph of Arrest Distribution by Borough and age Group , we can see highest crimes happened in Kings(18071 records) followed by Bronx(15264 records), and criminal suspects age are concentrated in 25-to 44.

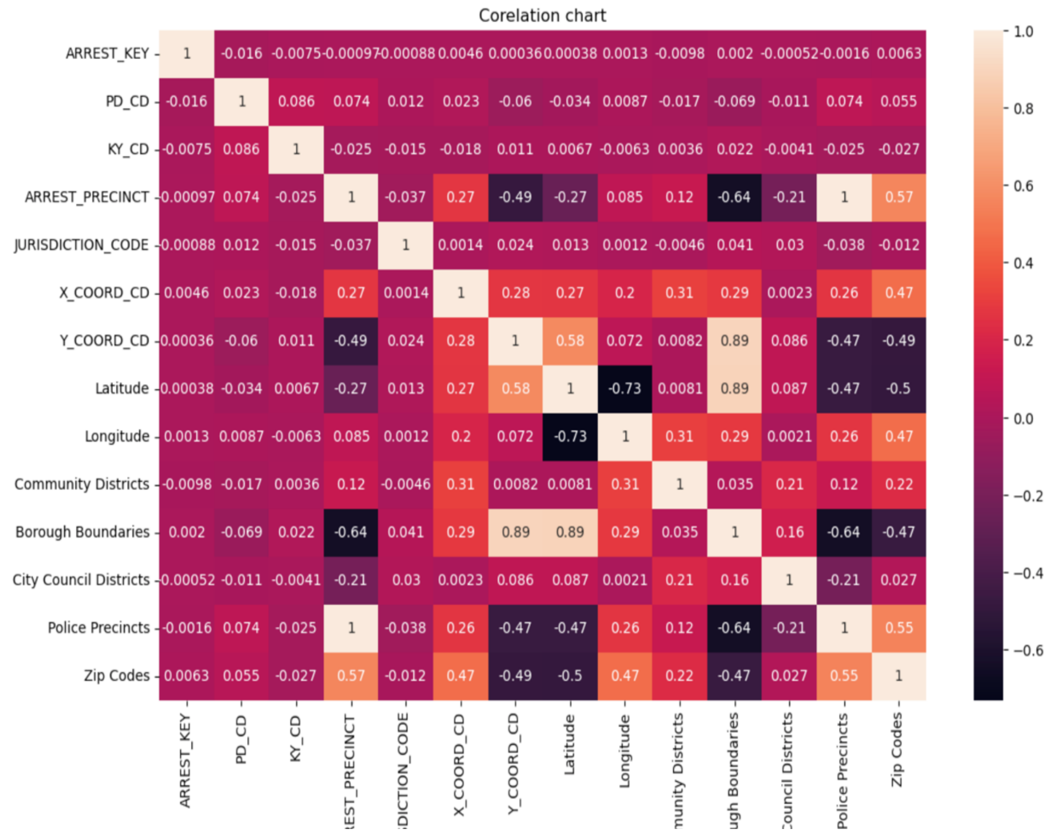
In [8]:
```
crime_by_borough_age = nyp.groupby(['ARREST_BORO', 'AGE_GROUP', 'OFNS_DESC']).size().unstack()
crime_by_borough_age.plot(kind='bar', stacked=True, figsize=(12, 6))
plt.title("Crime Distribution by Borough and Age Group")
plt.xlabel("Borough and Age Group")
plt.ylabel("Count")
plt.legend(title="Crime Type", bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```



EDA

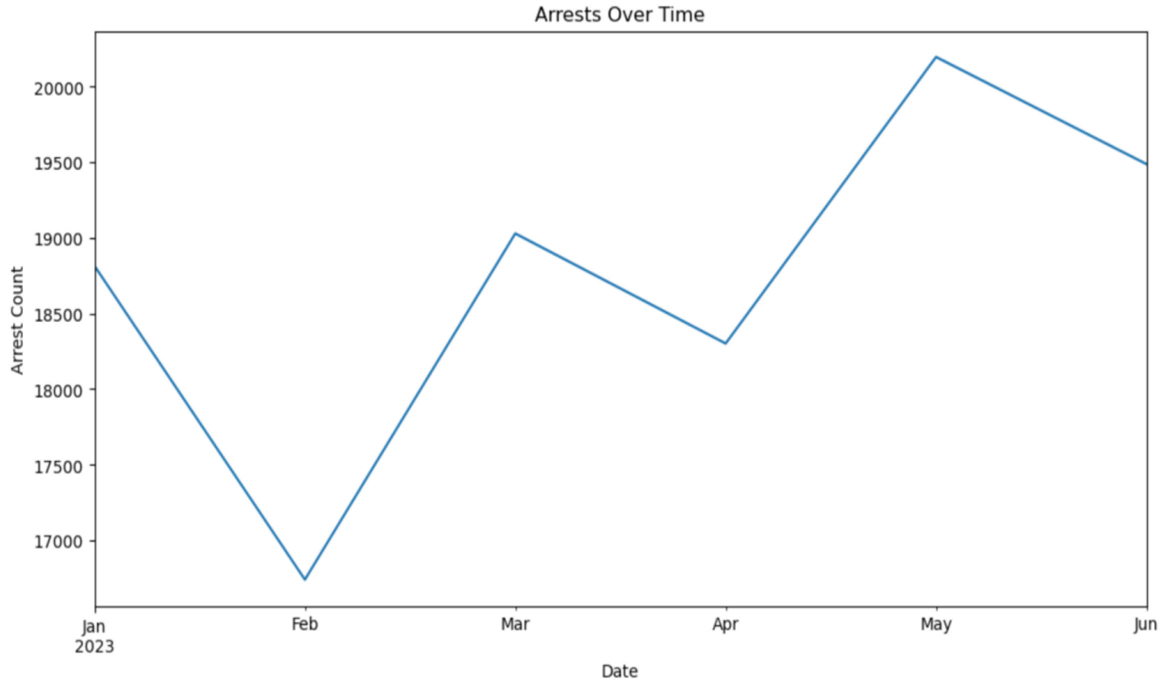
With the below heatmap, we can analyze whether there is a correlation between the variables to help us choose the subsequent algorithmic models.

```
olumns or specify the value of numeric_only to silence this warning.
  sns.heatmap(nyp.corr(), annot = True)
Out[12]: Text(0.5, 1.0, 'Corelation chart')
```



Arrest overtime graph indicate the trends in crime rates in New York arrest rates in 2023

```
[10]: nyp['ARREST_DATE'] = pd.to_datetime(nyp['ARREST_DATE'])
      arrests_over_time = nyp.resample('M', on='ARREST_DATE').size()
      arrests_over_time.plot(figsize=(12, 6))
      plt.title("Arrests Over Time")
      plt.xlabel("Date")
      plt.ylabel("Arrest Count")
      plt.show()
```



Predictive Analysis: Logistic Regression

Model:

```
In [22]:
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Select the target variable 'LAW_CAT_CD' and relevant features
target_column = 'LAW_CAT_CD'
features = ['ARREST_BORO', 'AGE_GROUP', ]

# Filter the DataFrame to include only the selected columns
nyp = nyp[[target_column] + features]

# Encode categorical features
label_encoders = {}
for column in features:
    if nyp[column].dtype == 'object':
        label_encoders[column] = LabelEncoder()
        nyp[column] = label_encoders[column].fit_transform(nyp[column])

# Split the dataset into training and testing sets
X = nyp[features]
y = nyp[target_column]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Standardize features (optional but can help with model performance)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Create and train the logistic regression model
model = LogisticRegression(max_iter=1000)  # You may need to adjust the max_iter based on your dataset
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
confusion = confusion_matrix(y_test, y_pred)
classification_report_str = classification_report(y_test, y_pred)

print(f"Accuracy: {accuracy}")
print("Confusion Matrix:")
print(confusion)
print("Classification Report:")
print(classification_report_str)
```

Results:

```
Accuracy: 0.5777500223035061
Confusion Matrix:
[[  422     0  9146     0]
 [    0     0    27     0]
 [  170     0 12530     0]
 [    1     0   122     0]]
Classification Report:
              precision    recall  f1-score   support

           F       0.71      0.04      0.08      9568
           I       0.00      0.00      0.00        27
           M       0.57      0.99      0.73     12700
           V       0.00      0.00      0.00       123

    accuracy                           0.58     22418
   macro avg       0.32      0.26      0.20     22418
weighted avg       0.63      0.58      0.45     22418
```

Predictive Analysis: Random Forest Classifier Model

Model:

In [23]:

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Assuming 'nyp' is your dataset

# Select relevant columns for the analysis
features = ['ARREST_BORO', 'AGE_GROUP']
target = 'LAW_CAT_CD'

# Split the data into training and testing sets
X = nyp[features]
y = nyp[target]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create and train the Random Forest Classifier model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
confusion = confusion_matrix(y_test, y_pred)
classification_report_str = classification_report(y_test, y_pred)

print(f"Accuracy: {accuracy}")
print("Confusion Matrix:")
print(confusion)
print("Classification Report:")
print(classification_report_str)
```

Results:

```
Accuracy: 0.5837273619412972
Confusion Matrix:
[[  623     0  8945     0]
 [    0     0    27     0]
 [  237     0 12463     0]
 [    1     0   122     0]]
Classification Report:
              precision    recall  f1-score   support

           F       0.72      0.07      0.12      9568
           I       0.00      0.00      0.00        27
           M       0.58      0.98      0.73     12700
           V       0.00      0.00      0.00       123

    accuracy                           0.58     22418
   macro avg       0.33      0.26      0.21     22418
weighted avg       0.64      0.58      0.46     22418
```

Conclusion/Finding

With the support of visualizations in Tableau and EDA, we‘ve uncovered disparities in arrests across racial groups, identified age-related patterns in law enforcement interactions. The borough with the highest crime rate is kings; The age group with the highest number of arrests in every borough is individuals aged 25-44, male Africa American and White Hispanic individuals are more involved in this age group. Assault, petit larceny and dangerous drugs are top 3 crime type appear in this age group and each borough.

To to explore the interplay of these factors to inform evidence-based, regulated suggestions for the NYPD. We need to picking out which models is best fitting to predict. By comparing the above two Prediction models, we can see that the logistic regression model is more suitable for our predictive analysis, and more accurate for predictions because our analysis is more designed to find out the potential correlation between the crime rate and the age and region, and with the help of this relationship classification model, we can help the NYPD to predict the future crime events and the behavioral dynamics of the suspects, which will reduce the overall crime rate.

Thank you