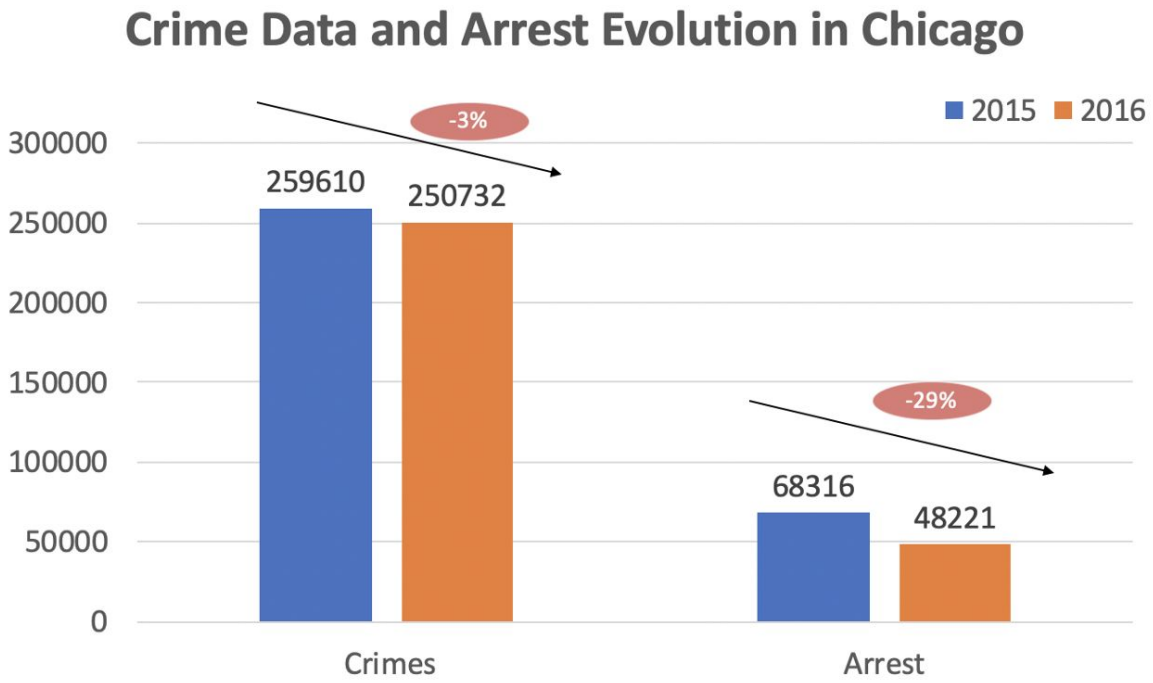
Chicago Crime Classification

Team Members:-

1. ***Parshv Patel***
2. ***Priyansh Patel***
3. ***Het Brahmbhatt***
4. ***Monil Sakhidas***

Introduction

- Explore the crime data in Chicago.
- Performing preprocessing, exploratory data analysis and classification using multiple algorithms.
- Implementation of a predictive model for arrests in Chicago.


Crime Data and Arrest Evolution in Chicago
2015
2016
300000
250000
200000
150000
100000
50000
0
-3%
259610
250732
-29%
68316
48221
Crimes
Arrest

Dataset Description

- The dataset was extracted from the Chicago Police Department's CLEAR (Citizen Law Enforcement Analysis and Reporting) system.

- The dataset is from the year 2001 till present i.e. 2021.

- This dataset contains over 7,312,666 observations and 22 features.

- Column Names

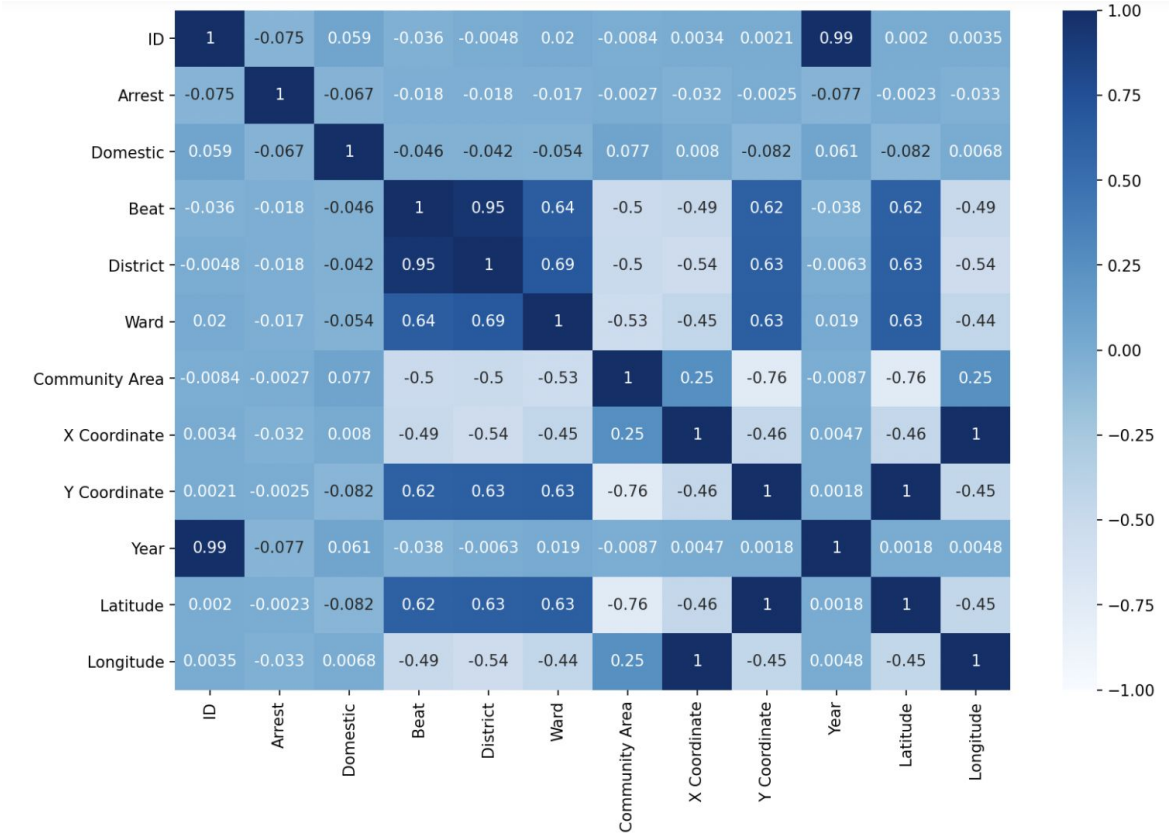
```
['ID',
 'Case Number',
 'Date',
 'Block',
 'IUCR',
 'Primary Type',
 'Description',
 'Location Description',
 'Arrest',
 'Domestic',
 'Beat',
 'District',
 'Ward',
 'Community Area',
 'FBI Code',
 'X Coordinate',
 'Y Coordinate',
 'Year',
 'Updated On',
 'Latitude',
 'Longitude',
 'Location']
```

Preprocessing on the dataset

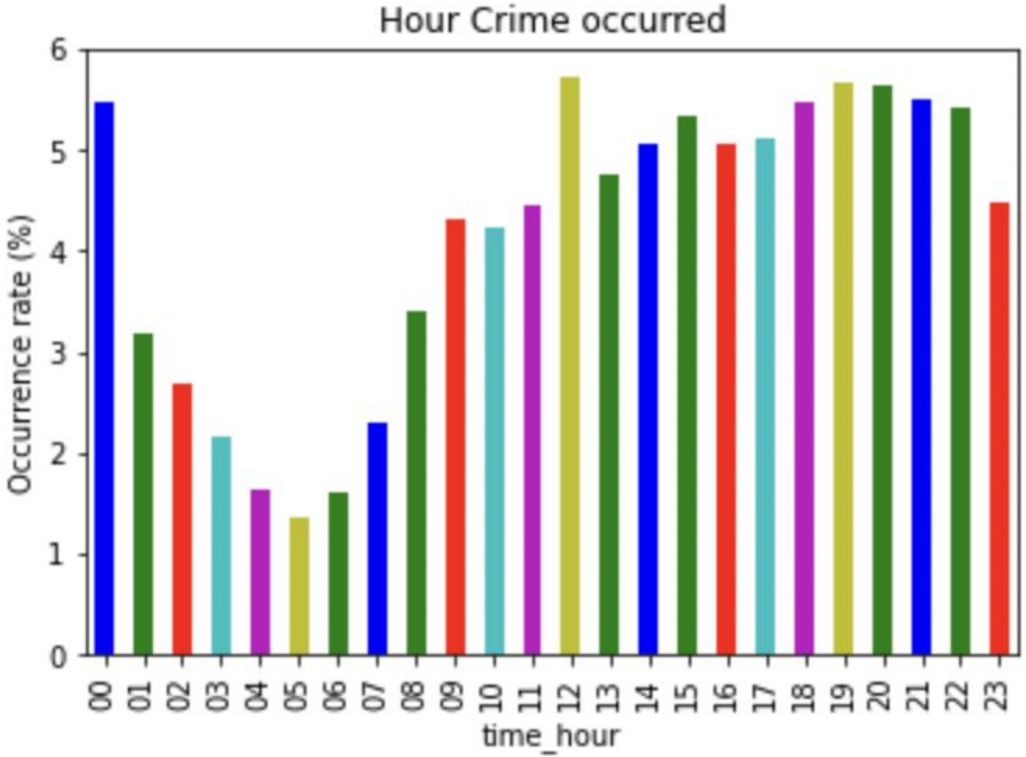
- Checked the number of missing values in the dataset.
- Dropped the missing values since the dataset is very large and also the missing values are less than 10% of the entire dataset.
- Also checked the feature type and corrected it if necessary.
- Checked if there are any duplicate rows in the dataset but couldn’t find any.
- Created new features by extracting the month, day and hour from the ‘Date’ column.

Exploratory Data Analysis

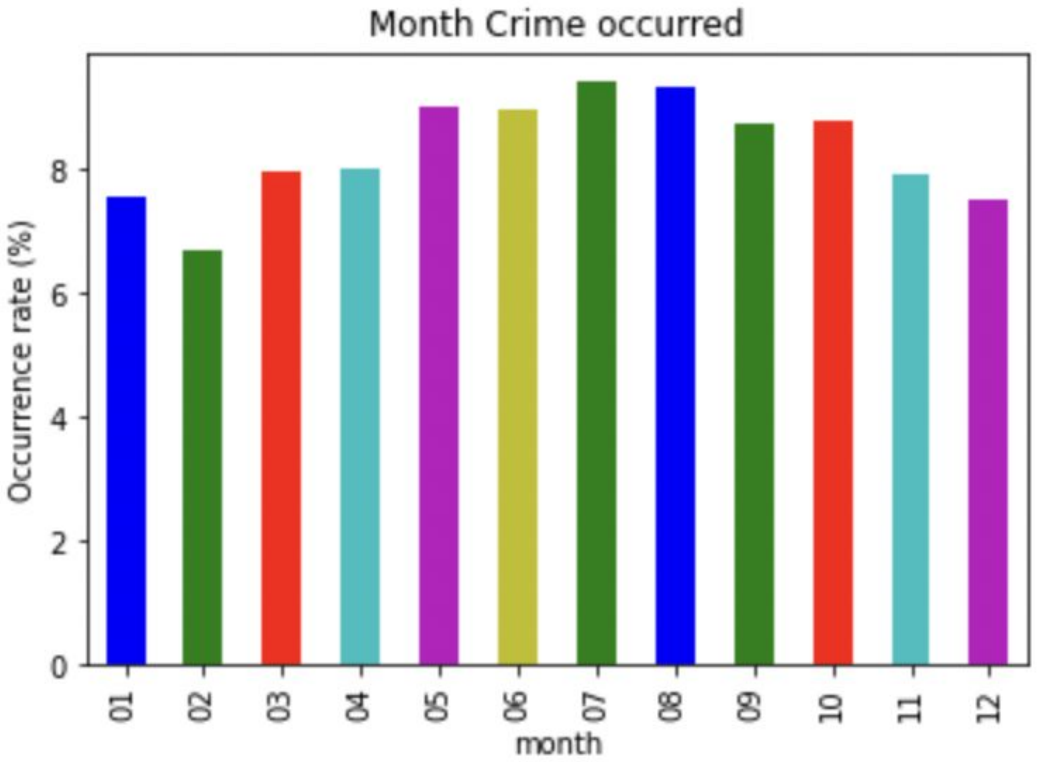
- Heatmap of the dataset



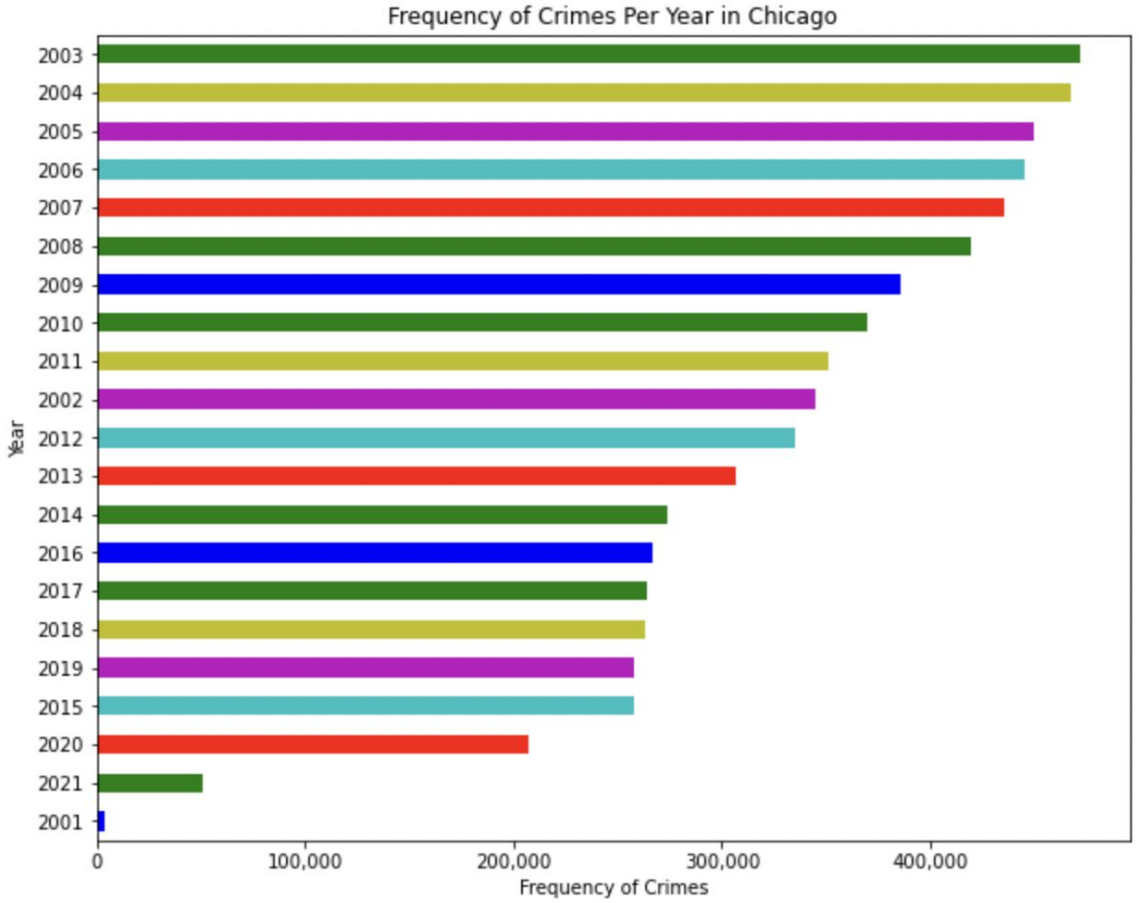
- The graph shows the percentage of occurrence of crime at a particular hour.



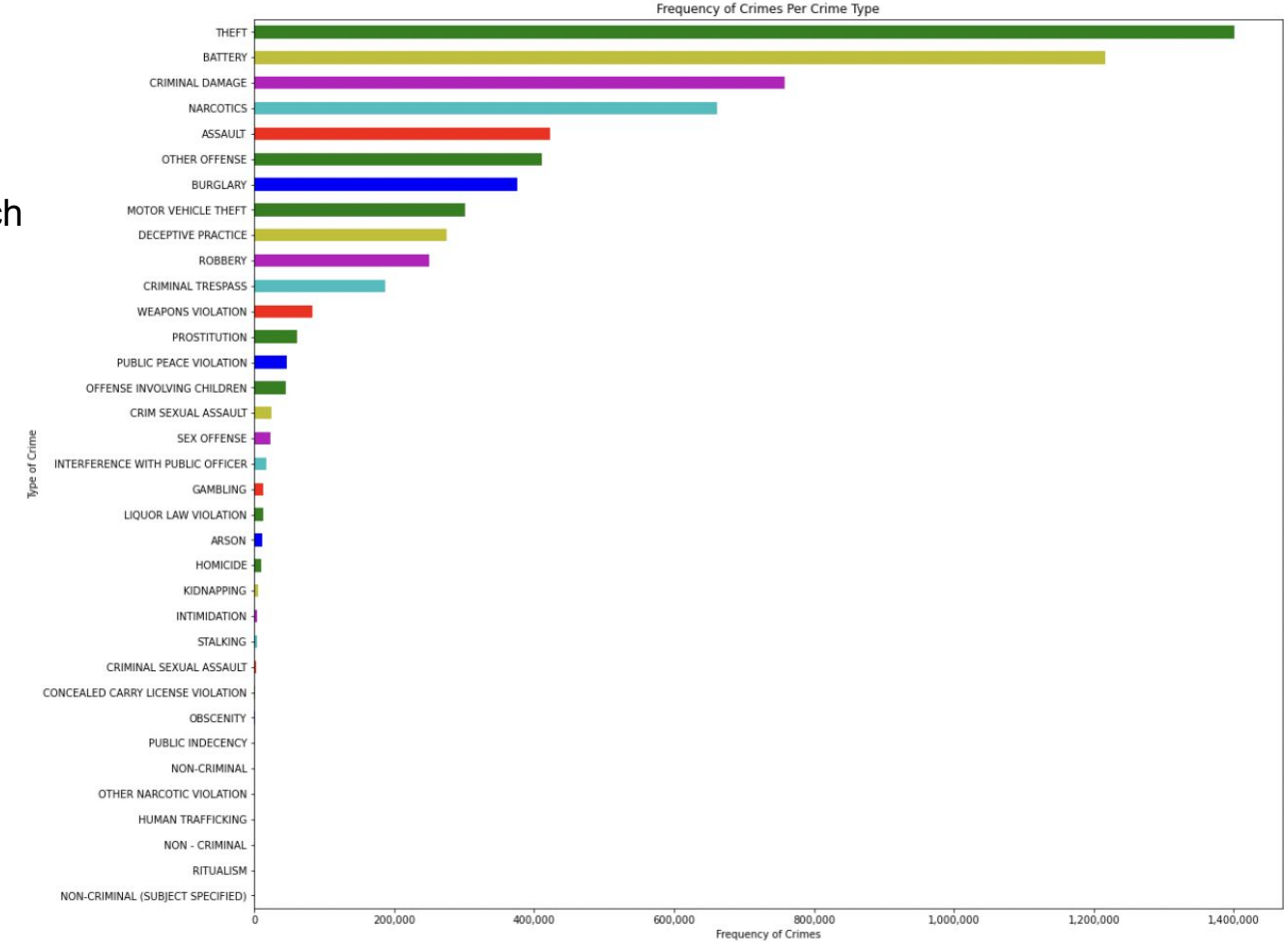
- The graph shows the percentage of occurrence of crime at a particular month.



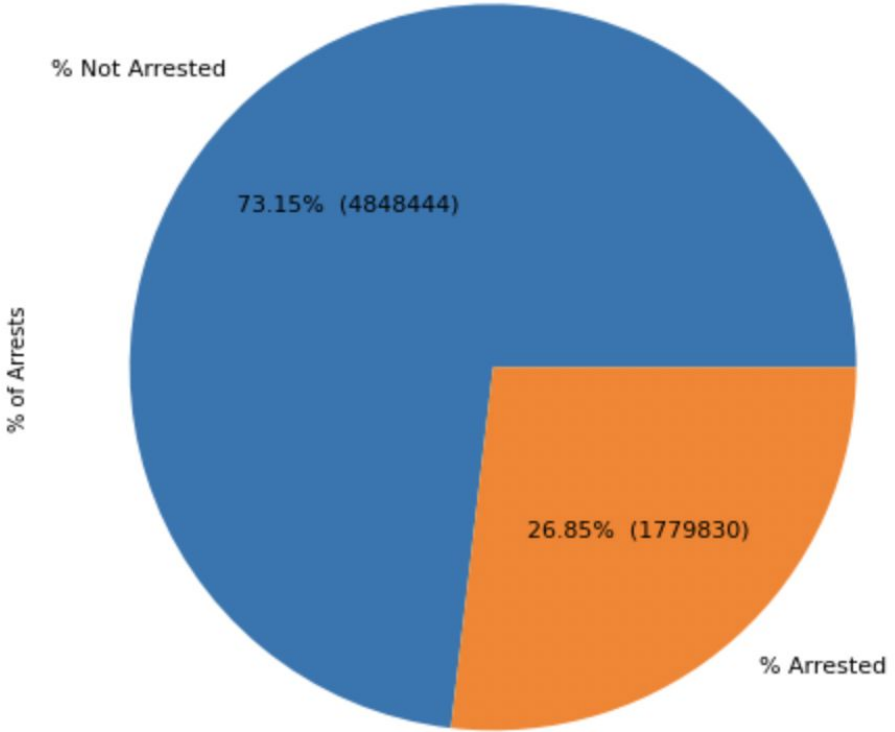
- The graph shows the number of crimes that took place at a particular year.



- The graph shows the Frequency of Crimes for each particular crime type.

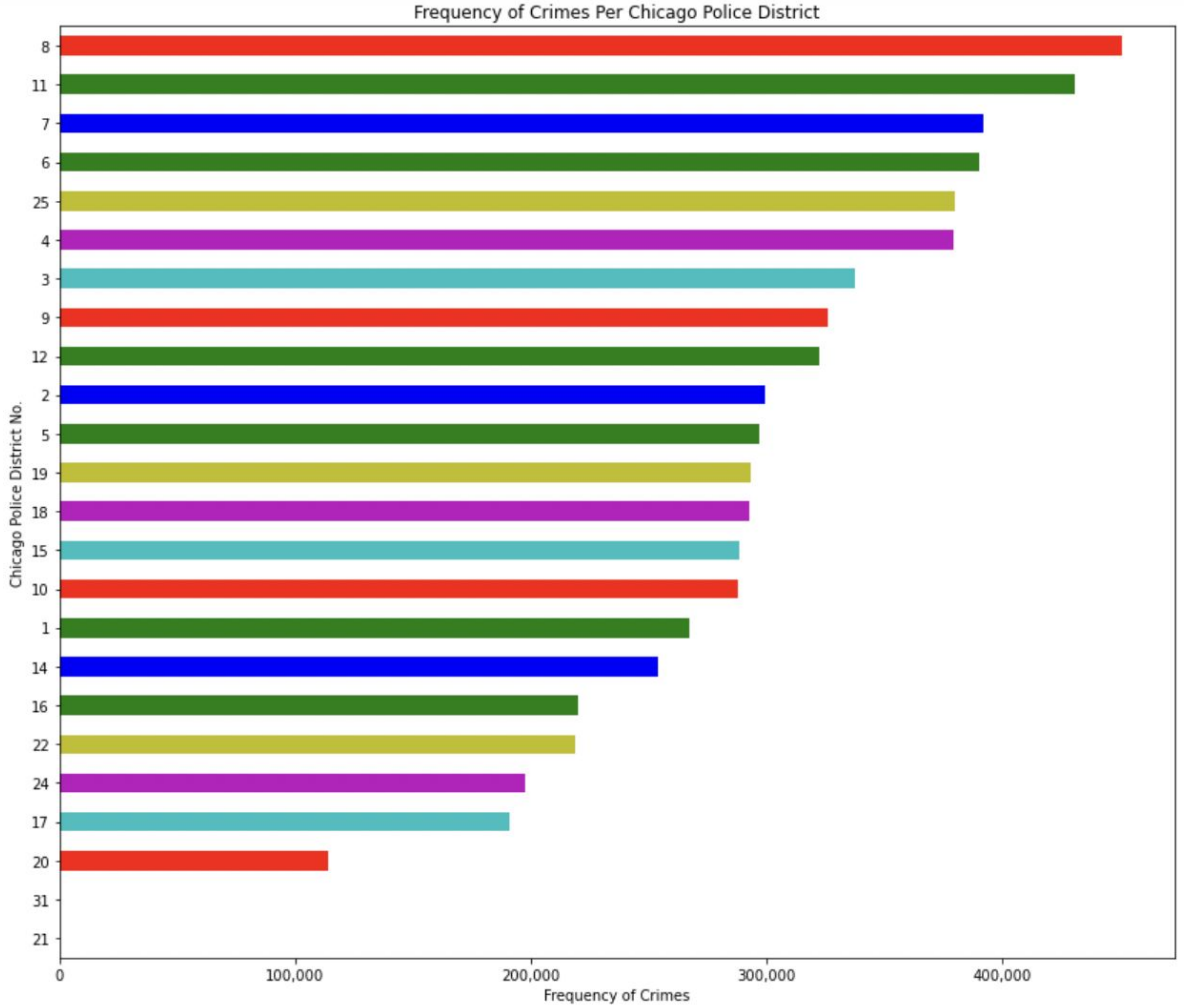


- Plotted a pie chart to visualize the percentage of arrests

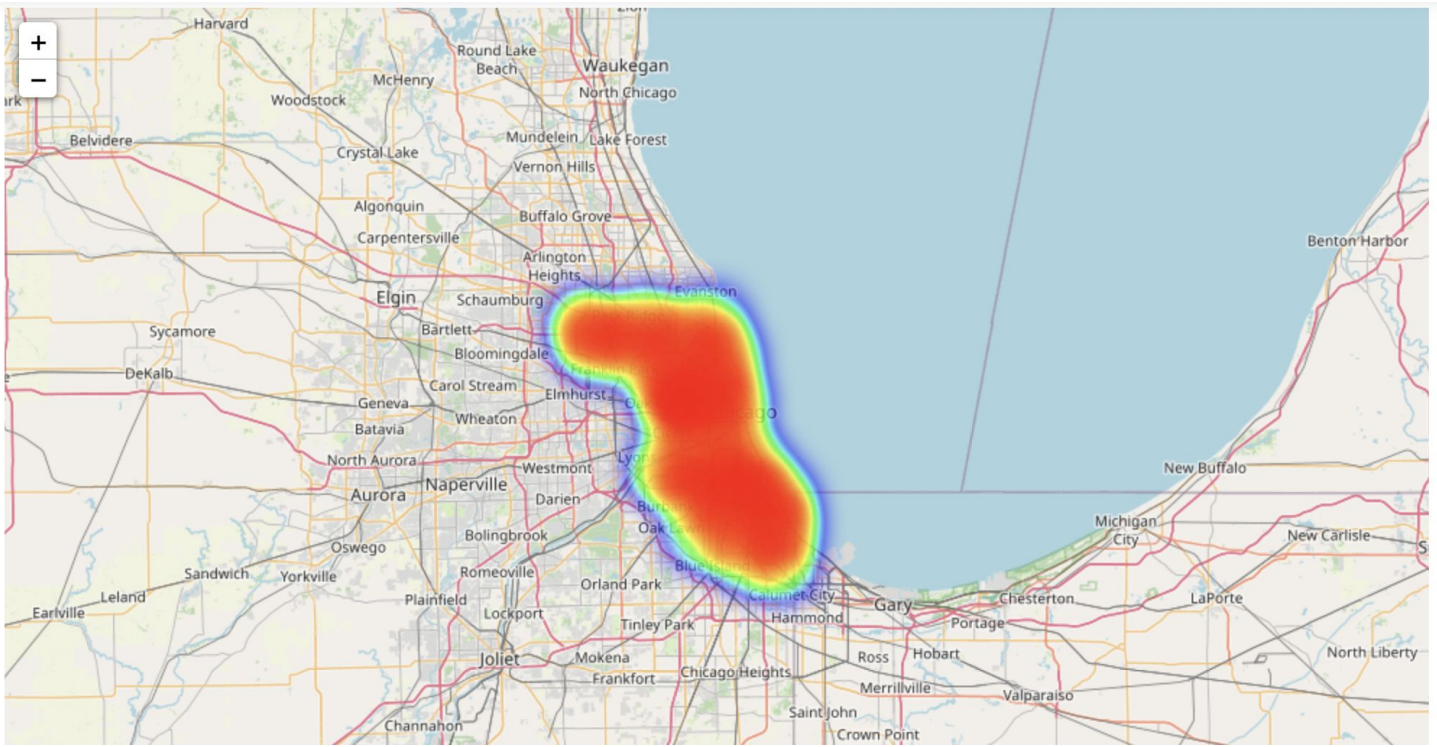


- A line plot of Percentage of successful arrests from 2001 to 2021 which shows the successful arrest percentage for each year from 2001 to 2021

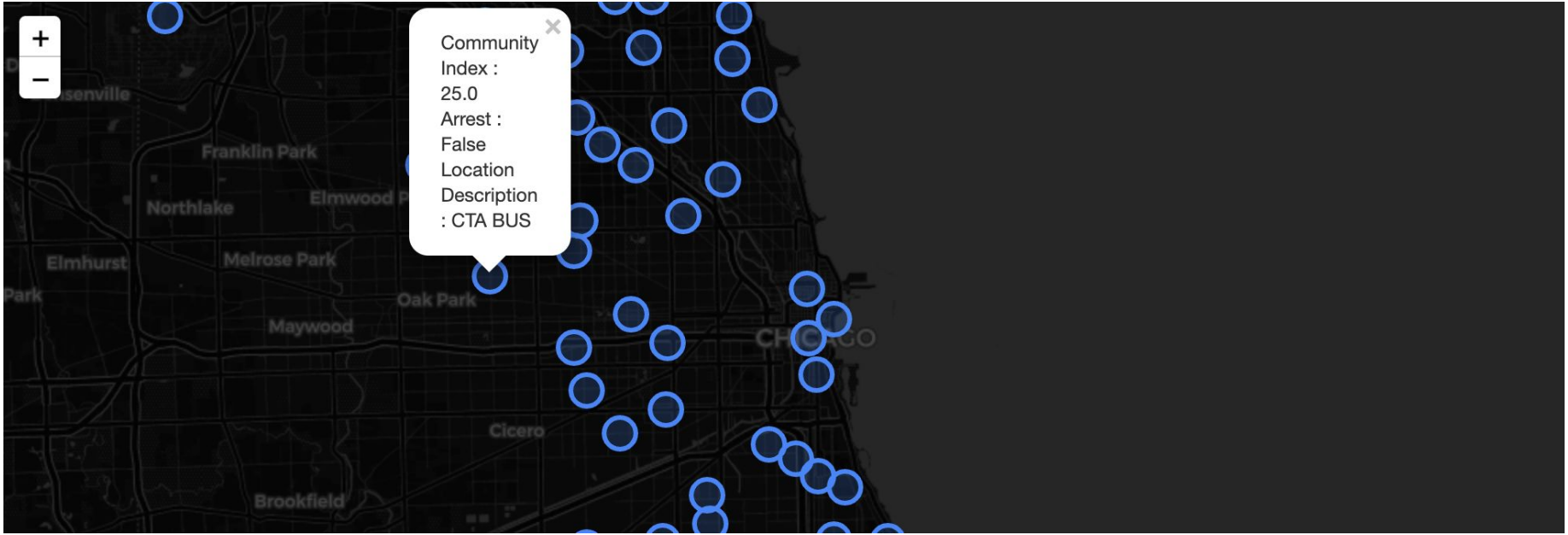
- The graph shows the Frequency of Crimes per Chicago Police District which shows the number of crimes registered at a particular Chicago Police District No.



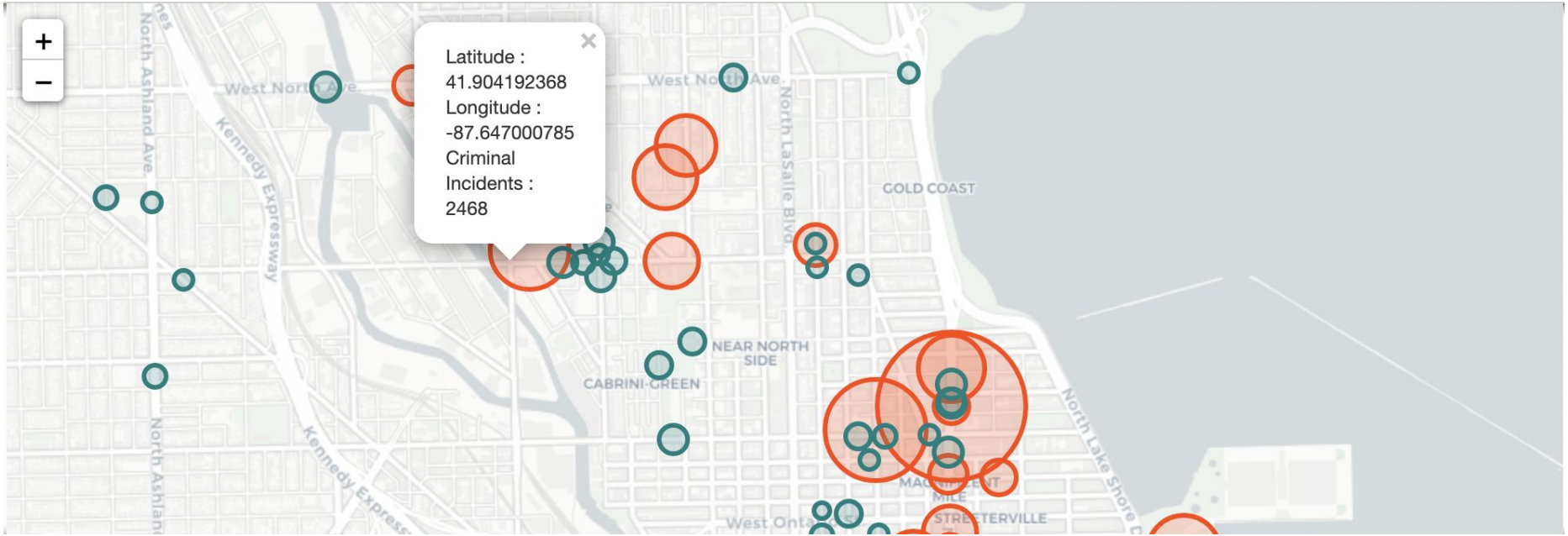
- Created a heatmap of crimes in Chicago March 2018



- Plotted a Map of Chicago city which shows the Community Index based upon Community Area, Arrest and Location Description for each crime that took place in the city.



- Plotted a Map of Chicago city which shows the Latitude and Longitude at which the crime took place and Number of criminal incidents that took place at that location.



Target Selection

- We defined three things that would be interesting to predict with this dataset:-

  1. The ward where a crime will happen.

  2. The type of crime (column “Primary Type”)

  3. If a crime will end up in an arrest.

- Due to high cardinality of Ward and Primary Type, we decided to use “Arrest” feature as the target.

	Unique Values
Ward	50
Primary Type	33
Arrest	2

Splitting of Dataset

- We extracted the X_features and the y_target from the dataframe.

- We then splitted the X and y into X_trainval, X_test, y_trainval and y_test keeping test size as 0.2.

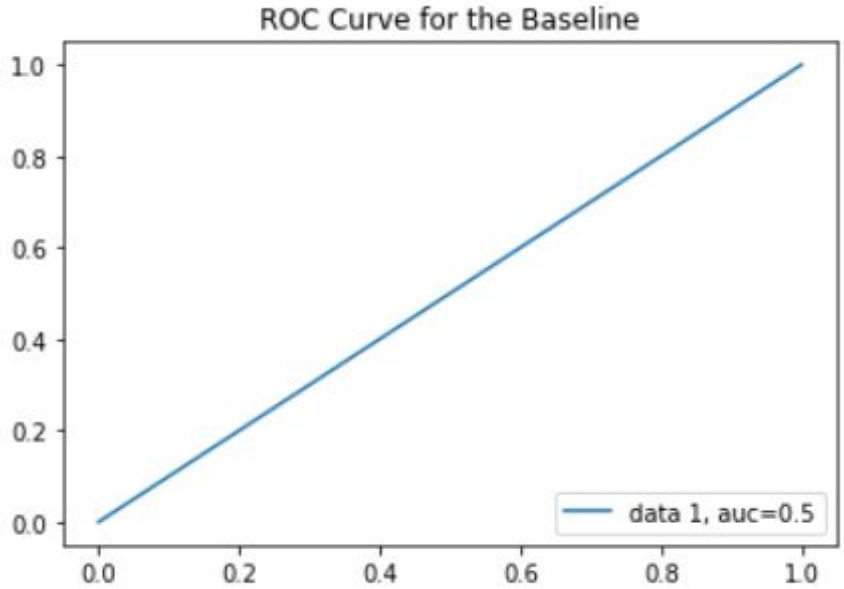
```
X_trainval, X_test, y_trainval, y_test = train_test_split(
    X, y, train_size=0.80, test_size=0.20, random_state=42)

X_train, X_val, y_train, y_val = train_test_split(
    X_trainval, y_trainval, test_size=0.2, random_state=42)
```

- We then splitted the X_trainval and y_trainval into X_train, X_val, y_train and y_val keeping the validation size as 0.2.

- Hence we splitted the entire dataset into train, validation and test dataset.

Getting accuracy score for majority class baseline

- The objective of baseline is to create an initial prediction.
- To calculate an accuracy percentage.
- This will be the standard to beat with the future predictive model.
- Here we have used mode as the prediction because our target is categorical.



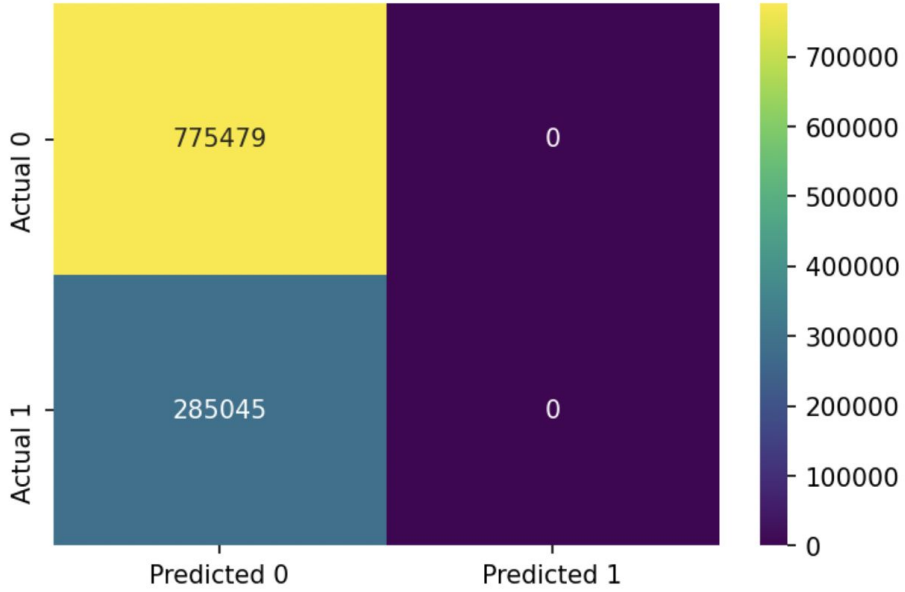
Model Selection

- We have implemented many different classification algorithms. They are :-

    - Logistic Regression
    - XGBoost
    - Random Forest Classifier
    - Decision Tree Classifier
    - Naive Bayes Classifier
    - Support Vector Machine

Logistic Regression

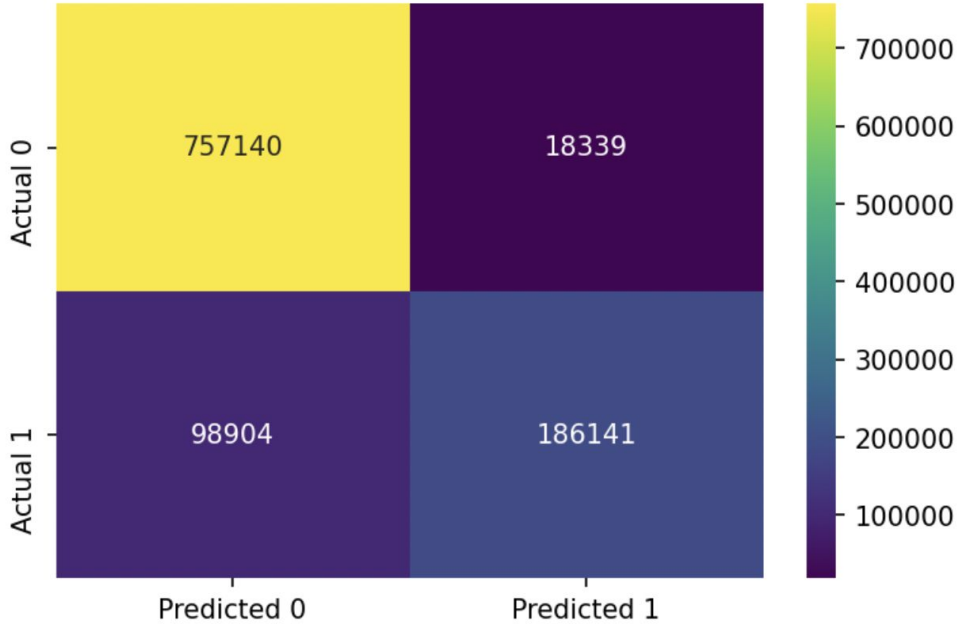
- Made a pipeline which contains category encoders, simple imputer and estimator logistic regression.
- Faster to train
- Gives lesser accuracy
- Accuracy :- 73%



	precision	recall	f1-score	support
0	0.73	1.00	0.84	775479
1	0.00	0.00	0.00	285045
accuracy			0.73	1060524
macro avg	0.37	0.50	0.42	1060524
weighted avg	0.53	0.73	0.62	1060524

XGBoost

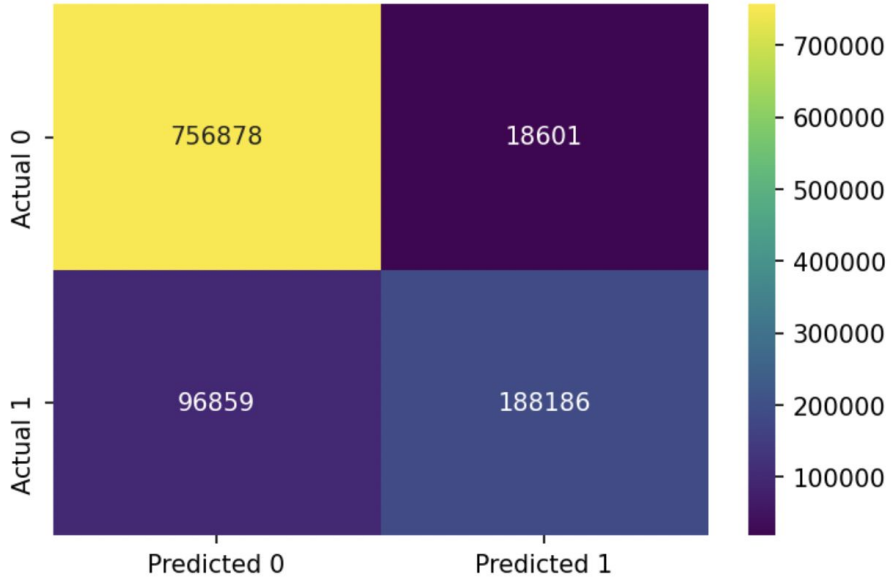
- Made a pipeline which contains category encoders, simple imputer and estimator XGBClassifier.
- Takes longer to train than logistic regression.
- Gives higher accuracy than logistic regression.
- Accuracy :- 89%



	precision	recall	f1-score	support
0	0.88	0.98	0.93	775479
1	0.91	0.65	0.76	285045
accuracy			0.89	1060524
macro avg	0.90	0.81	0.84	1060524
weighted avg	0.89	0.89	0.88	1060524

Random Forest Classifier

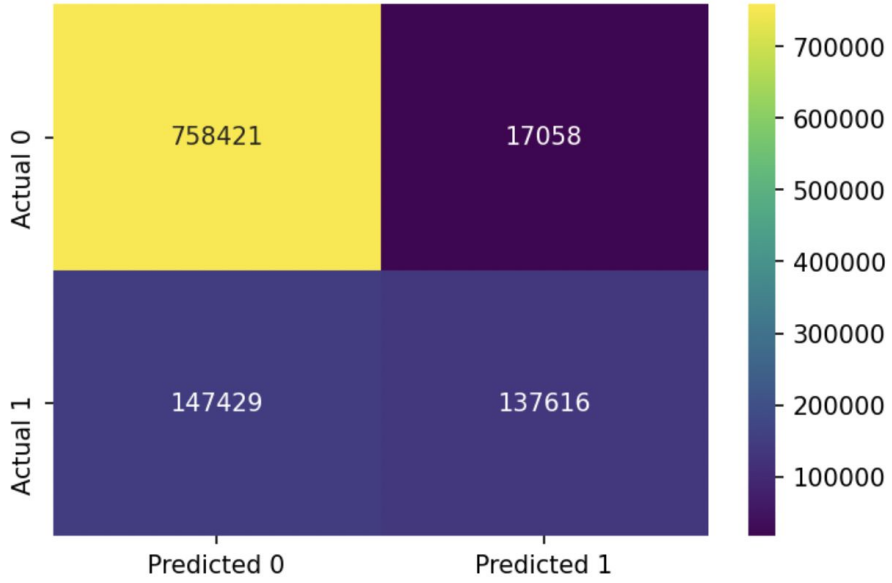
- Made a pipeline which contains category encoders, simple imputer and estimator Random Forest Classifier.
- Takes almost similar time to train as XGBoost.
- Gives almost same accuracy as XGBoost.
- Accuracy :- 89%



	precision	recall	f1-score	support
0	0.89	0.98	0.93	775479
1	0.91	0.66	0.77	285045
accuracy			0.89	1060524
macro avg	0.90	0.82	0.85	1060524
weighted avg	0.89	0.89	0.89	1060524

Decision Tree Classifier

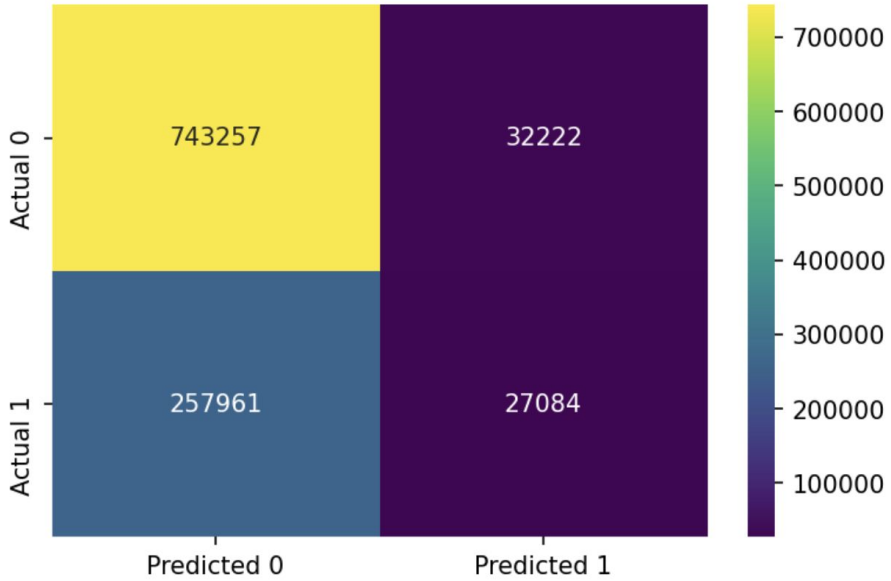
- Made a pipeline which contains category encoders, simple imputer and estimator Decision Tree Classifier.
- Takes lesser time to train than XGBoost and Random Forest Classifier.
- Gives lesser accuracy than XGBoost and Random Forest Classifier.
- Accuracy :- 84%



	precision	recall	f1-score	support
0	0.84	0.98	0.90	775479
1	0.89	0.48	0.63	285045
accuracy			0.84	1060524
macro avg	0.86	0.73	0.76	1060524
weighted avg	0.85	0.84	0.83	1060524

Naive Bayes Classifier

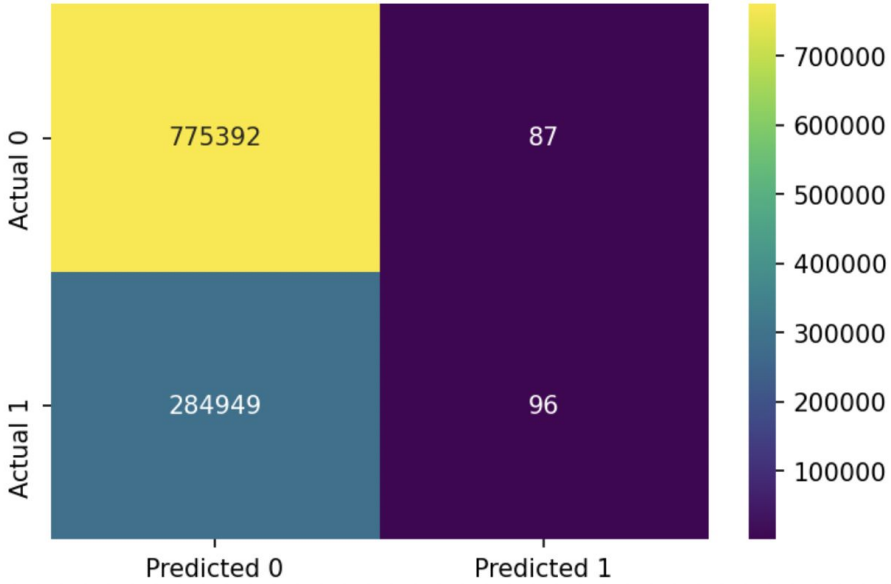
- Made a pipeline which contains category encoders, simple imputer and estimator Naive Bayes Classifier.
- Takes almost similar time to train as Decision Tree Classifier and Logistic Regression.
- Gives lesser accuracy almost similar to Logistic Regression.
- Accuracy :- 73%



	precision	recall	f1-score	support
0	0.74	0.96	0.84	775479
1	0.46	0.10	0.16	285045
accuracy			0.73	1060524
macro avg	0.60	0.53	0.50	1060524
weighted avg	0.67	0.73	0.65	1060524

Support Vector Machine

- Made a pipeline which contains category encoders, simple imputer and estimator Support Vector Machine.
- Takes very longer to train.
- Gives lesser accuracy almost similar to Logistic Regression and Naive Bayes Classifier.
- Accuracy :- 73%



	precision	recall	f1-score	support
0	0.73	1.00	0.84	775479
1	0.52	0.00	0.00	285045
accuracy			0.73	1060524
macro avg	0.63	0.50	0.42	1060524
weighted avg	0.68	0.73	0.62	1060524

Conclusion

- We performed different preprocessing techniques on the dataset such as removing null values, removing duplicates and many more.

- We also performed Exploratory Data Analysis on the dataset and plotted some graphs to visualize the dataset better.

- Then we tried out different classification algorithms for training and evaluation of the dataset.

Thank you