# Project Topic

Crime Analytics using NYPD Historical Complaint Data.

# Team Number and Team Members

Team Number 22

Members:

* Kshitij Prabhu (NEU ID 002769231)
* Manav Hirey (NEU ID 002786706)

# Introduction

A group of police officers

Description automatically generated

The most populated city in the US is New York City (NYC). New York Metropolis is also the most densely populated large city in the United States, with an estimated population of 8,804,190 in 2020 spread across 300.46 square miles. The city, which is situated at the southernmost point of New York State, serves as the physical and demographic hub for both the Northeast megalopolis and the New York metropolitan area, which is the largest metropolitan area in the world in terms of the total area covered by urban territory. New York is one of the most populated megacities in the world, with over 20.1 million residents in its metropolitan statistical area and 23.5 million in its combined statistical area as of 2020, and over 58 million people reside within 250 miles of the city.

# Methodology

With the help of this data set, we will be able to determine the following Two things: -

• Classification of Crime

* This will be carried out using K-Means Clustering and Naïve Bayes to help us classify crime in 3 categories; Felony, Misdemeanors and Violations, that we can then employ in the objectives.

• Crime rate

* We will first create a graph of New York's crime rates by year, after which we will train the data to make predictions for two to three years. Here, we'll employ SVM and Linear Regression.

# Dataset Description

The dataset contains data points of all legitimate felonies, misdemeanors, and violations reports received by The New York City Police Department (NYPD) from 2006 to the end of 2021.

# Data Sources

<https://www.kaggle.com/datasets/supreeth888/nypd-data>

# Result and Analysis



First, let us better understand the dataset and obtain more specific information about it.

The primary goal is to discover mistakes, anomalies, and distinct patterns in data. One of the tools, Exploratory Data Analysis, enables analysts to have a better understanding of the data before making assumptions. EDA data may help businesses understand their consumers, build their business, and make appropriate decisions.

We first explore the “LAW\_CAT\_CD” which is the column that categorizes complaints received by the New-York City Police Department into one of the three categories ,i.e., Misdemeanors, Violations and Felony as mentioned in the diagram below.

A misdemeanor is a type of offense punishable under criminal law. A misdemeanor is typically a crime punishable by less than 12 months in jail. Community service, probation, fines, and imprisonment for less than a year are commonly issued punishments for misdemeanors.

The term “violation” includes any action (alone or with another or others) for or toward causing, bringing about, participating in, counseling, or aiding or abetting a violation.

According to the definition mentioned in the 192nd General Court of the commonwealth of Massachusetts a felony is described as A crime punishable by death or imprisonment in the state prison is a felony. All other crimes are misdemeanors.

We were able to find these three unique values by using the unique function that is available in the inbuilt python library. As we can see that majority of the crimes present in this dataset are categorized as misdemeanor (56%), followed by felony (31%) and violation that make up 13% of the data present in the dataset.

Chart, pie chart

Description automatically generated

*Data distribution within the column “LAW\_CAT\_CD”*

We go further into the column "OFNS DESC," which gives a description of the offense committed. Further investigation reveals that there are 64 different string values. This information includes a description of what occurred during the crime or the complaint that the citizen is dealing with.

The three-digit offence classification code "KY CD" is used to describe the description of the offence. The combination of "KY CD" and "OFNS DESC" gives significance to the three-digit number. The photo below shows the maximum number of complaints classified as "Harassment 2," which is signified by the three-digit integer 578.

A picture containing table

Description automatically generated

*Types of complaints received by NYPD*

Chart, bar chart

Description automatically generated

*Boroughs from where complains are coming from*

In the dataset, “BORO\_NM” contains the name of the borough from where the complaint was received by the New York police Department. As we can see from this bar plot the greatest number of complaints that the department has received from is from the Brooklyn Borough which is around 30% of the data present in this dataset.

Map

Description automatically generated

With the help of jurisdiction code and jurisdiction description we are able to figure out what department of the city will be able to solve the issue. The column names “JURISDICTION\_CODE” and “JURIS\_DESC” is able to label each complain that is conveyed by residing citizen. “CRM\_ATPT\_CPTD\_CD” provides us with a label if the crime was attempted or completed.

Chart, pie chart

Description automatically generated

*Percentage of crimes completed to attempted*

There are several columns present inside of the dataset that denote the exact location of the crime where it was committed. We can plot all the crimes that are mentioned in this dataset on a map, but since there are too many data points on the map this will look something like the diagram mentioned below. The data is too concentrated for a person to distinguish between where exactly the crime was committed. Hence, we further delve deep into learning more about how we can precisely classify crime into three broad categories; Misdemeanors, Violation and Felony.

Map

Description automatically generated

*Map of New York City*

Map

Description automatically generated

*Crimes plotted on a map*

Moving on to cleaning of the dataset, data cleaning is the process of locating and rectifying any faulty or incorrect data in a dataset. One such method is to do something about the missing values in the dataset. Numerous datasets will include many missing values in real life, therefore dealing with them is a critical step.

Why do you need to fill in the blanks? Because most machine learning models will generate an error if NaN numbers are passed into them. The simplest solution is to just fill them with 0, however this will drastically lower the accuracy of your model.

There are several techniques for filling in missing data. To fully grasp how to manage missing values in Python, you must first understand the type of missing value and its relevance before you begin filling/deleting the data.

*Chart, bar chart

Description automatically generated*

*Plotting the sum of null values present inside of the dataset.*

You can deal with missing values using one of the following methods mentioned below:

1. removing the columns with missing data
2. Delete the rows that have missing data
3. Adding a value to the missing data - Imputation
4. Imputation with an extra column
5. Using a Regression Model to Fill

In this project we first used isnull() function inorder to find all the null values in this dataset. We first dropped all the irrelevant columns such as “PARK\_NM”, “X\_COORD\_CD”,”Y\_COORD\_CD”, etc. so that we can move on to using the dropna() python function in order to drop all null values that are present inside of the NYPD Complain Data dataset.

After carefully studying and exploring the NYPD complaint dataset. We have finally decided the objective of our project and that will be the classification of crimes based on its description that is provided in the columns of “OFNS\_DESC” and “PD\_DESC”. We also try to recreate a machine learning model that trains on past year’s data and predicts crime rate of the next 25 days into the future.

Table

Description automatically generated

*Total number of Null Values found in all columns in the dataset.*

# Using **Support Vector Machine** and **Linear Regression**, we predicted the last **25 days** of the year **2019**

The support vector machine algorithm seeks a hyperplane in an N-dimensional space (N — the number of features) that distinguishes between data points. There are several hyperplanes that might be used to split the two groups of data points. Our goal is to discover a plane with the greatest margin, or the greatest distance between data points from both classes. Maximizing the margin distance gives some reinforcement, allowing subsequent data points to be categorized with more certainty.

Hyperplanes are decision boundaries that aid in the classification of data items. Data points on either side of the hyperplane might belong to distinct classes. Furthermore, the size of the hyperplane is determined by the number of features. When the number of input features is two, the hyperplane is simply a line. When the number of input features reaches three, the hyperplane transforms into a two-dimensional plane. When the number of features exceeds three, it becomes difficult to imagine.

Support vectors are data points that are closer to the hyperplane and have an effect on its position and orientation. Using these support vectors, we maximize the classifier's margin. The position of the hyperplane will change if the support vectors are removed. These are the points that will assist us in developing our SVM.

Line chart

Description automatically generated with low confidence Chart

Description automatically generated

Using Support-Vector Machine we can predict the crime rate for the next 25 days in the future by training the model on past data. Now in order to judge a the SVM model that we just trained we use Root Mean Square (RMSE). We also use R-squared value for judging the model. (model-explained total variance) / total variance." If it is 100%, the two variables are fully linked, i.e., there is no variation. Its formula is as follows:

Text

Description automatically generated

*Formula for Root-Mean Squared Value used to judge a Machine-Learning Model*

We have also predicated the crime rate for the next 25 days using Linear Regression. Linear regression analysis predicts the value of one variable depending on the value of another. The variable you wish to forecast is referred to as the dependent variable. The variable you are using to forecast the value of the other variable is known as the independent variable.

This type of analysis calculates the coefficients of a linear equation that includes one or more independent variables that best predict the value of the dependent variable. Linear regression is used to fit a straight line or surface that minimizes the differences between predicted and actual output values. Simple linear regression calculators that use the "least squares" method to find the best-fit line for a set of paired data are available. The value of X (dependent variable) is then estimated from Y (an Independent Variable).

# Result and Findings for the implementation of SVM and Linear Regression.

Graphical user interface, chart, line chart

Description automatically generated

*SVM prediction for the next 25 days in the future*

Chart, scatter chart

Description automatically generated

*Linear Regression Scatter plot for prediction*

Graphical user interface, chart, line chart

Description automatically generated

*Linear Regression plot on the entire plot*

Graphical user interface, text, application

Description automatically generated

*R2 score comparison between SVM and Linear Regression*

*Graphical user interface, text, application, email

Description automatically generated*

*RMSE Score comparison for SVM and Linear Regression*

So, as we can see from the above results, we have observed that the RMSE score for SVM is more than the RMSE value of Linear Regression. Similarly, we can see that the R-Squared value for SVM is again way smaller than the R-Squared value of Linear Regression. To conclude our findings, we have observed that Support Vector Machine Learning model performs way better than the Linear Regression Classification model.

# Using **K-Nearest Neighbor** and **Decision Tree**, we classified which type of crime complaint was under what Jurisdiction

The k-nearest neighbors’ algorithm, often known as KNN or k-NN, is a non-parametric, supervised learning classifier that employs proximity to classify or predict the grouping of a single data point. While it may be used for either regression or classification issues, it is most commonly utilized as a classification technique, based on the idea that comparable points can be discovered nearby.

A majority vote is used to apply a class label to a classification problem—that is, the label that is more commonly expressed around a specific data point is utilized. While officially this is referred to as "plurality voting," the term "majority vote" is more generally used in literature. The difference between these terms is that "majority voting" technically requires a majority of more than 50%, which only works when there are only two options. When there are multiple classes—for example, four categories—you don't always need 50% of the vote to decide about a class; you could assign a class label with a vote of more than 25%.

Decision Trees (DTs) are a type of non-parametric supervised learning approach that may be used for classification and regression. The objective is to build a model that predicts the value of a target variable using basic decision rules derived from data attributes. A tree is an example of a piecewise constant approximation.

Graphical user interface, text, application, email

Description automatically generated

*Comparison of RMSE values for Decision Tree and KNNs*

# Result and Findings for the implementation of K-Nearest Neighbors and Decision Tree

As we can see from the above results, the RMSE score for KNN is greater than the RMSE value for Decision Tree. Similarly, the R-Squared value for SVM is significantly lower than the R-Squared value for Linear Regression. To summarize our findings, we discovered that the K-Nearest Neighbors model is at par with the Decision Tree Classification model.

# Using **K-Means**, Gaussian Naïve Bayes and Multinomial Naïve Bayes, we classified Crime using Criminal Description

The K-means method in data mining begins with a first set of randomly picked centroids, which serve as the starting points for each cluster, and then performs iterative (repetitive) computations to optimize the centroids' locations.

It halts constructing and optimizing clusters when either:

* Because the clustering was effective, the centroids have stabilized — there has been no change in their values.
* The maximum number of iterations has been reached.

The Bayes Theorem is used to influence Naive Bayes, a simple but effective probabilistic classification model in machine learning.

The Bayes theorem is a formula that provides a conditional probability of an event A occurring if another event B has previously occurred. It has the following mathematical formula: -

Text, letter

Description automatically generated

Types of Naïve Bayes:

1. Naive Gaussian Bayes: When the predictor values are continuous and are predicted to follow a Gaussian distribution, this classifier is used.
2. Multinomial Naïve Bayes: This classifier employs a multinomial distribution and is frequently used to tackle document or text categorization problems.

# Result and Findings for the implementation of Naïve Bayes and K-Means

Chart, line chart

Description automatically generated

*Elbow Method to find the optimal value of K, which in this case is 3*

Chart, scatter chart

Description automatically generated

*K-Means Clustering scatter plot*

*Graphical user interface, text, application

Description automatically generated*

*Comparison of accuracy score between Gaussian Naïve Bayes and Multinomial Naïve Bayes*

*Text

Description automatically generated with medium confidence*

*Accuracy score of K-Means*

*Chart, bar chart, waterfall chart

Description automatically generated*

*Algorithm Accuracy Comparison between GaussianNB, MultinomialNB and K-Means*

From the above images we can clearly see that Multinomial Naïve Bayes Classifier is working the best with it being accurate about 90% of the time followed by Gaussian Naïve Bayes which is accurate about 75% of the time and the least accurate is K-Means with it being accurate about 30% of the time. This outcome is expected as K-Means is a unsupervised machine learning model whereas Naïve Bayes is a supervised machine learning model.

# References:

* <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>
* <https://scikit-learn.org/stable/modules/svm.html>
* <https://www.ibm.com/topics/linear-regression>
* <https://medium.com/swlh/predictive-modelling-using-linear-regression-e0e399dc4745>
* <https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1>
* <https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm>
* <https://neptune.ai/blog/k-means-clustering>
* <https://www.analyticsvidhya.com/blog/2020/10/a-simple-explanation-of-k-means-clustering/>
* <https://stackoverflow.com/questions/57626286/how-to-plot-text-clusters>
* <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761>
* <https://learn.g2.com/k-nearest-neighbor>
* <https://www.codingninjas.com/codestudio/library/knn-vs-k-means>
* <https://www.google.com/search?client=safari&rls=en&q=gaussian+naive+bayes&ie=UTF-8&oe=UTF-8>
* <https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html>
* <https://jakevdp.github.io/PythonDataScienceHandbook/05.05-naive-bayes.html>
* <https://www.geeksforgeeks.org/decision-tree-implementation-python/>
* <https://machinelearningmastery.com/implement-decision-tree-algorithm-scratch-python/>
* <https://www.analyticsvidhya.com/blog/2020/10/all-about-decision-tree-from-scratch-with-python-implementation/>
* <https://scikit-learn.org/stable/modules/tree.html>
* <https://www.geeksforgeeks.org/linear-regression-python-implementation/>
* <https://realpython.com/linear-regression-in-python/>
* <https://www.analyticsvidhya.com/blog/2022/02/linear-regression-with-python-implementation/>