UNIVERSITY PARTNER

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

Big Data

(6CS030)

BIG DATA IN CRIME RATE ANALYSIS

Student Id : 2050453

Student Name : Kunga Nyima Gurung

Group : LGCG5

Module Leader : Mr. Jnaneshwar Bohara

Submitted on : 4/13/2022

BIG DATA IN CRIME ANALYSIS & PREDICTION

Kunga Nyima Gurung; Manjil Shrestha

Herald College Kathmandu

Abstract- **Criminality is one of the country's most pressing social issues, having an impact on public peace, child psychology, and grownup social class. Policymakers trying to reduce crime and improve citizens' quality of life must first understand what factors contribute to the rise in crime rates? as with the increasing crime rates it is difficult to determine the safe and danger zones. The goal of the project is to look at big data analytics for criminal data predictions and come up with machine learning-based solutions to help tackle some crime-related problems. We study the crime rate in the United States using several data mining clustering and classification methodologies. These promising discoveries will aid in the prevention of future crimes by utilizing analysis performed in comparison to the available data using machine learning algorithms.**

Keywords – Crime, Machine Learning, Hadoop, Spark, MongoDB.

I. Introduction

Criminality is one of the country's most pressing social issues, having an impact on public peace, child psychology, and grownup social class. Policymakers trying to reduce crime and improve citizens' quality of life must first understand what factors lead to rising crime rates. Understanding how to control crime is critical since, in the United States, exposure to violence and crime has been abnormally high for several decades and, while reducing, remains high. According to [1]'s research, there are several evidence that the decline in crime is due to a reduction in minor infractions. Homicide and intimate relationship violence, which are often not perpetrated with co-offenders, have either remained stable or grown.

II. Problem Statement:

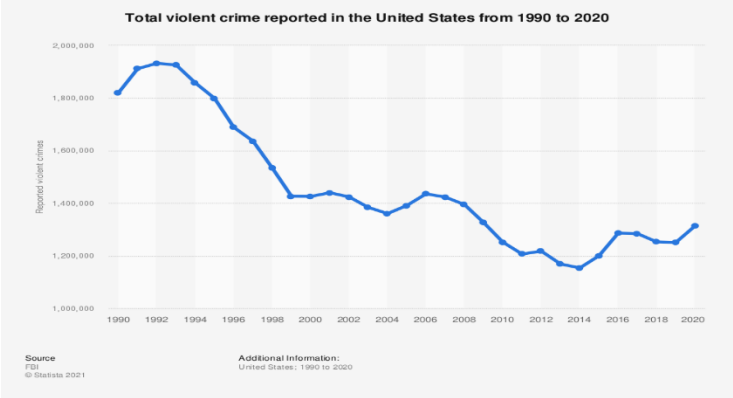

Figure 1: Graphical representation of crime rate in the US 2020

From 1990 to 2020, violent crime fell across the country, according to the FBI. The number of violent crimes reached in 1992 at 1.93 million, but has since declined to 1.15 million in 2014. The FBI's Uniform Crime Reporting Program classified offenses involving force or threat as violent crimes because they have an impact on people's lives. In 2020, there were 21,570 murder and nonnegligent manslaughter cases reported in the United States. On a list of US states, California came in first, followed by Texas, and then Florida. [2]. Crimes are rising daily, with varying degrees of violence and diversity. As a result, society suffers significant losses in terms of monitory loss, social loss, and increased threat to society's smooth livelihood.

III. Aims and Objective:

The primary goal of this report is to determine how much crime has increased in various US police departments throughout the years. On the basis of data analysis and visualization, certain trends would be recognized, examined, and discussed in order to make well-informed decisions so that law and order may be

properly maintained and individuals feel safe and secure. As a result, the project was picked with the goal of preventing or controlling future illegal activity. We must travel to a variety of locations on a regular basis for our daily needs, and We deal with a range of safety issues on a regular basis, including hijacking, kidnapping, and harassment. When we need to get someplace, we usually look to Google Maps first. Google Maps displays one, two, or more routes to the target, but because we don't understand the travel scenario accurately, we always select the shortcut route. Is it really safe, and if it is, why are there so many issues? The project's purpose is to look at big data analytics for criminal data predictions and come up with solutions based on machine learning. The crime rate prediction methodologies can be used on historical data from police records by looking at the data from many perspectives, such as the reason for the crime, the frequency of comparable crimes at a certain place, and other parameters to create a model for crime prediction. Understanding the diverse data accessible to us, then modeling it to forecast future incidence with acceptable accuracy and lowering the crime rate, is a huge task.

IV. Project Structure

The remainder of this report is organized in the following manner.Page-2 includes a review of related literature. The methodologies for trend analysis described in Page-3 are deep learning and machine learning. Page-5 explores into the data processing and visualization approaches used and presents the experimental results, which are followed by some concluding notes in Page-7.

V. Related Works

For a long time, big data analytics has been widely used and studied in the domains of data science and computer science. When working with massive amounts of large data, machine learning presents both potential and challenges [3].

The writers of a book released by [4] discuss the issues that the healthcare business has, such as gathering, storing, finding, sharing, and evaluating data. Furthermore, the book highlights the issues that data scientists, engineers, and physicians face while using Big Data and provides solutions, for data scientists, engineers, and doctors, with a focus on data repositories, issues, and concepts.

[5] are familiar with using Apache Pig and Hadoop to solve the socioeconomic problem of crime, which requires using incident-level crime data provided by past identical crime incidences to discover similar vicious crime episodes selectively.[6] addressed some of the most extensively used data mining algorithms as well as the many software frameworks available for BDA.

Many researchers have used similar methods to predict future crimes based on particular input data. They used a variety of methods to test the prediction models, including [7]'s artificial neural network, [8]'s support-vector, and [9]'s time series analysis, among others. From historical data, previously undiscovered information can be better extracted. For crime analysis, [10] used clustering algorithms.

[11] suggested a machine learning system for detecting criminal trends and tendencies. During the analysis of experimental data, the ID3 algorithm provides more reasonable and effective categorization criteria. This work also proposes using the Bayes algorithm to train a model with crime data. After testing, it was discovered that utilizing the Bayes theorem, classification approaches provided greater than 90% accuracy.

To aid local police, [12] proposed a study in which a regression model was constructed using data from Indian statistics, which provides data on various crimes over the past 14 years (2001-2014), and the crime rate for the next years in various states may be forecasted. They employed supervised, semi-supervised, and unsupervised learning algorithms on crime records to reveal new information and improve crime prediction accuracy.

[13] offered two parameters to quantify such patterns for crime prediction: an auto-regressed temporal correlation and a

feature-based inter-area spatial correlation. They also introduced a tree-structured clustering approach for discovering high comparable areas based on geographical attributes, which can help our proposed model perform better.

V. Methodologies

Machine learning and data mining, which is a process of acquiring data, extracting information from it, and making predictions based on that knowledge, have become extremely important when working with data. Any present pattern in that data can be examined, which is quite important for planning ahead.

A. Data Collections:

During the data collecting process, we often collect data from a variety of websites, including news sites, blogs, social media, RSS feeds, and so on. The dataset of violent crimes in the United States in 2016 was created using data from data. world . The raw dataset contains 8114 rows of data and 13 columns that indicate the names of major US cities, population, Total Violent Crimes, and information about various violent crimes such as assault, rape, burglary, theft, and so on. Another Kaggle dataset was used, which contains crime-related data for the city of Chicago in the United States from 2012 to 2017. There are 1456714 rows of data and 23 columns in the raw dataset. The dataset comprises data on various case numbers, types of crimes, FBI codes, and so forth.

B. Data Preprocessing:

A series of preprocessing processes for data conditioning are performed before executing any algorithms on our datasets, as shown below:

- • We opted to remove unknown values from the data set because they are an indicative of incomplete papers rather than a value to be considered.
- Because the Chicago dataset is too vast, we would only subsample a dataset for modeling as a proof of concept.
- The dataset was stripped of any extraneous or non-meaningful attributes that did not add to the model's accuracy measurement or prediction of crime.

C. Variable Selection

Rather of projecting crime levels, we focused on Units or major cities in our article. As a result, we chose Crime Type as our target variable and 'IUCR', 'Description', 'FBI Code' as our predictor factors.

D. Data Visualization

Data visualization can take numerous forms. Pie chart, bar chart, heat map, histograms, and more visualizations are included in the report. These visualizations help us understand varied information, such as the most common crime type, the association between crime kinds, and so on.

E. Model Selection

We might use a variety of machine learning classifiers to forecast the specific unit. The following are some of the machine learning algorithms used:

1) Linear Regression:

A linear model is one in which the independent variables and the single dependent variable are assumed to have a linear relationship. The dependent variable is continuous and their relationship is linear [14]. The model's performance was calculated, and the mean squared error result was way too high, approximately 130. since the result was too high, and the dataset's model was underperforming because as we know, the lower the MSE, the better the model.

2) K-Nearest Neighbor

The KNN algorithm is a classification method that determines which category something belongs to. K-nearest neighbors is used when the target variable must be classified into more than two classes [15]. In this dataset, there are three types of predictor variables that can be used to predict the target variable crime type: IUCR, Description, and FBI Code. After hyper parameter bosting and label encoding, KNN accuracy was 99.9%.

3) Random Forest

Random Forest is a type of ensembled classifier that use a randomized decision tree algorithm. Everything is chosen at random, from samples to features, and a decision tree is created [16]. Because Random Forest is a versatile, easy-to-use machine learning method that produces outstanding results in the vast majority of scenarios even without hyper-parameter optimization. Random Forest accuracy was 99.6 percent after hyper parameter bosting after label encoding.

4) Neural Network

Deep learning is based on neural networks, a branch of machine learning in which the algorithms are inspired by human brain anatomy. A neural network takes in data, trains to recognize patterns in it, and then predicts the outcome for a new batch of data that is similar. When applied to human problems, neural-network types such as feedforward and feedback propagation artificial neural networks perform better, according to the researchers. [17]. In their application to pattern recognition tasks, modern ANN models perform admirably [18]. Neural Network accuracy was 98.2 percent after hyper parameter bosting after label encoding.

5) Logistic Regression

When the dependent variable is nominally or ordinally scaled, a particular kind of regression analysis called logistic regression is used. The classification accuracy will be improved by selecting a better mix of characteristics. The logistic regression classification model, according to the results of the trials, can forecast the index range and save time during the training stage [19]. Logistic Regression accuracy was around 47 percent in our model.

F. MongoDB

MongoDB's flexible design makes it easier to work with data in flat formats like CSV and hierarchical data, where individual fields are made up of several items [20]. MongoDB was used to perform some basic operations like importing the data into MongoDB database, check number of documents in collections, checking unique values etc.

G. Hadoop and Apache Spark

Hadoop makes it easy to use all of a cluster server's storage and processing capabilities, as well as to perform distributed operations on massive data sets. It acts as a platform for the development of new services and applications [21]. . For big data workloads, Apache Spark is an open-source distributed processing solution. For speedy analytic queries against any scale of data, it combines in-memory caching and rapid query execution [22].Hadoop and Apache Spark were used in this project to perform basic operations like storing the data in the dfs, reading the csv file using SQL queries etc.

H. Work Flow Diagram

The data for this paper was first gathered through Kaggle and Data World. Following that, we reduced the data to 100000 and applied Machine Learning algorithms to forecast the outcomes of various crimes. The process began with gathering the data, filtering it, implementing algorithms, and producing an analysis result and conclusion. K-Nearest Neighbor produced the best results, with a 99.9% accuracy rate.

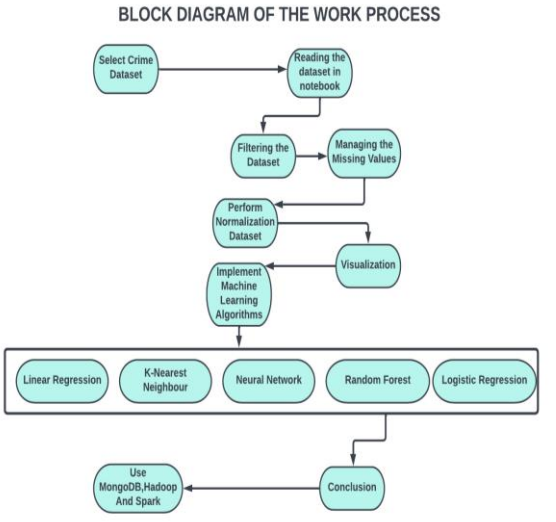

Figure 2: Work Flow Diagram.

VI. Results and Discussion

A. Reading and exploring 2016 dataset

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        %matplotlib inline
        import seaborn as sns
        import warnings
        warnings.filterwarnings('ignore')

In [2]: df = pd.read_csv("./US_Crime_Rate_2016.csv")
        df.head()
```

Out[2]:

	City	Population	Violent crime	Murder and nonnegligent manslaughter	Rape1	Robbery	Aggravated assault	Property crime	Burglary	Larceny-theft	Motor vehicle theft	Arson2
0	New York	8379043.0	47821.0	319.0	2770.0	13396.0	31336.0	122299.0	9846.0	106931.0	5522.0	NaN
1	Los Angeles	8379043.0	29400.0	258.0	2274.0	9652.0	17216.0	90704.0	13809.0	66253.0	10642.0	1672.0
2	Chicago	8379043.0	25932.0	492.0	1761.0	7983.0	15296.0	80742.0	9678.0	62083.0	9081.0	416.0
3	Houston	8379043.0	25257.0	275.0	1249.0	9147.0	14586.0	101750.0	17038.0	71614.0	13098.0	485.0
4	Detroit	8379043.0	13040.0	275.0	552.0	2345.0	9467.0	28550.0	6820.0	14844.0	6886.0	789.0

```
In [3]: df.columns.tolist()
```

Out[3]:
```
['City',
 'Population',
 'Violent crime',
 'Murder and nonnegligent manslaughter',
 'Rape1',
 'Robbery',
 'Aggravated assault',
 'Property crime',
 'Burglary',
 'Larceny-theft',
 'Motor vehicle theft',
 'Arson2']
```

Figure 3: Reading the Violent crime 2016 dataset

B. Cleaning

The pandas library was used to read the dataset of the United States' 2016 violent crime rate. Information on various violent crimes in various cities is acquired. Data exploration was carried out to determine the dataset's shape, as well as unique column values.

```
In [8]: df.dropna(inplace=True)

In [9]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 7937 entries, 1 to 8883
Data columns (total 12 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   City                                  7937 non-null   object
 1   Population                            7937 non-null   float64
 2   Violent crime                         7937 non-null   float64
 3   Murder and nonnegligent manslaughter  7937 non-null   float64
 4   Rape1                                 7937 non-null   float64
 5   Robbery                               7937 non-null   float64
 6   Aggravated assault                    7937 non-null   float64
 7   Property crime                        7937 non-null   float64
 8   Burglary                              7937 non-null   float64
 9   Larceny-theft                         7937 non-null   float64
 10  Motor vehicle theft                   7937 non-null   float64
 11  Arson2                                7937 non-null   float64
dtypes: float64(11), object(1)
memory usage: 806.1+ KB
```

```
In [10]: df.describe()
```

Out[10]:

	Population	Violent crime	Murder and nonnegligent manslaughter	Rape1	Robbery	Aggravated assault	Property crime	Burglary	Larceny-theft	Motor vehicle theft	Arson2
count	7.937000e+03	7937.000000	7937.000000	7937.000000	7937.000000	7937.000000	7937.000000	7937.000000	7937.000000	7937.000000	7937
mean	7.162187e+06	96.092226	1.237369	10.417286	23.057830	61.379740	554.639158	83.772564	410.972650	59.893915	2
std	2.931687e+06	710.505605	10.766028	56.536709	220.564414	436.874052	2898.714693	469.964810	2067.730681	410.355006	25
min	5.000000e+00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0
25%	8.379043e+06	2.000000	0.000000	0.000000	0.000000	1.000000	22.000000	5.000000	16.000000	1.000000	0
50%	8.379043e+06	10.000000	0.000000	2.000000	1.000000	6.000000	83.000000	12.000000	63.000000	5.000000	0
75%	8.379043e+06	37.000000	0.000000	6.000000	5.000000	24.000000	326.000000	45.000000	254.000000	20.000000	1
max	8.379043e+06	29400.000000	492.000000	2274.000000	9652.000000	17216.000000	101750.000000	17038.000000	71614.000000	13042.000000	1672

Figure 4: data filter

C. Analyzing and Visualization

Data filtration was performed for null and missing values, and certain fundamental statistical features of a data frame were discovered, such as percentile, mean, and standard deviation.

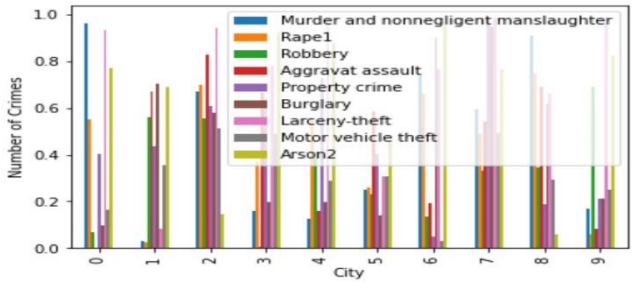

Figure 5: Bar graph Violent Crime 2016

Following the filtration of the dataset, a bar plot was created to determine which violent crime occurred the most in 2016. Larceny theft is the most common crime in the top ten countries, as shown in the bar graph.

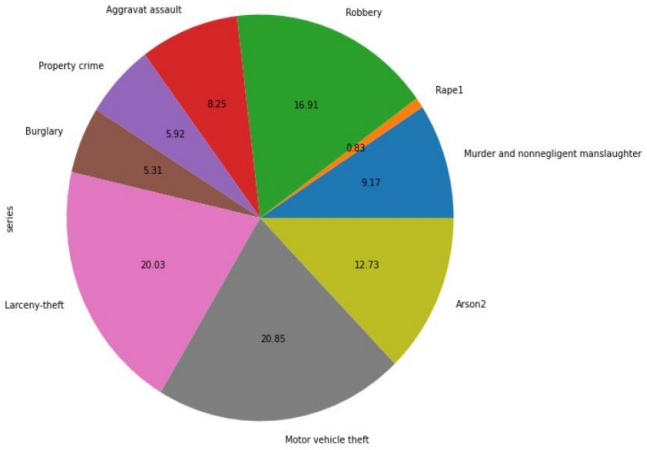

Figure 6: Pie Chart Violent crime 2016

When a random 10 data were taken and a pie chart plotted on the basis of violent crimes. We can see that the most occurred violent crime is Motor vehicle theft with 20.85 % followed by Larceny theft with 20.03 %.

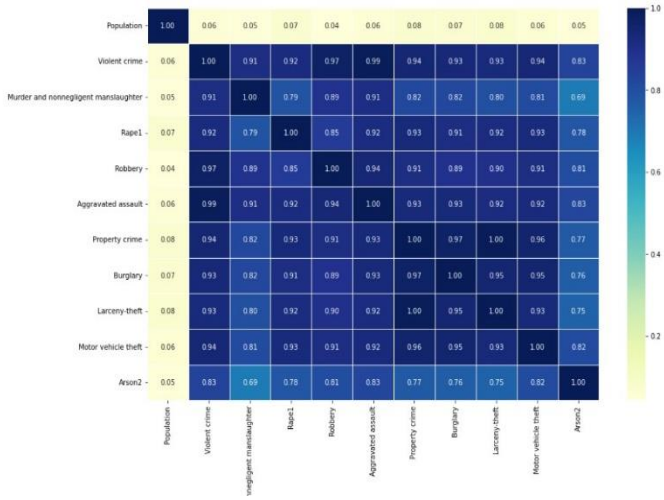

Figure 7: Heat Map Violent crime 2016

The features were chosen by generating a heat map between them and using correlation. The correlation coefficient has a range of values from -1 to 1. A larger positive correlation is shown by a number closer to 1, whereas a strong negative correlation is indicated by a value closer to -1.

D. Reading and exploring 2012-2017 dataset

```
In [1]: # Visualization Libraries
        import matplotlib
        import matplotlib.pyplot as plt
        import seaborn as sns

        #Preprocessing Libraries
        import pandas as pd
        from sklearn.model_selection import train_test_split
        from sklearn.metrics import precision_score, recall_score, confusion_matrix, classification_report, accuracy_score, f1_score

        # ML Libraries
        from sklearn.ensemble import RandomForestClassifier,VotingClassifier
        from sklearn.neighbors import KNeighborsClassifier
        from sklearn.neural_network import MLPClassifier

        # Evaluation Metrics
        from yellowbrick.classifier import ClassificationReport
        from sklearn import metrics

In [2]: df = pd.read_csv("./Chicago_Crimes_2012_to_2017.csv")

In [3]: df.head()
Out[3]:
```

	Unnamed: 0	ID	Case Number	Date	Block	IUCR	Primary Type	Description	Location Description	Arrest	...	Ward	Community Area	FBI Code	X Coordinate	...
0	3	10508693	HZ250496	05/03/2016 11:40:00 PM	013XX S SAWYER AVE	0486	BATTERY	DOMESTIC BATTERY SIMPLE	APARTMENT	True	...	24.0	29.0	08B	1154907.0	
1	09	10508695	HZ250409	05/03/2016 09:40:00 PM	061XX S DREXEL AVE	0486	BATTERY	DOMESTIC BATTERY SIMPLE	RESIDENCE	False	...	20.0	42.0	08B	1183068.0	
2	197	10508697	HZ250503	05/03/2016 11:31:00 PM	053XX W CHICAGO AVE	0470	PUBLIC PEACE VIOLATION	RECKLESS CONDUCT	STREET	False	...	37.0	25.0	24	1140789.0	
3	673	10508698	HZ250424	05/03/2016 10:10:00 PM	049XX W FULTON ST	0460	BATTERY	SIMPLE	SIDEWALK	False	...	28.0	25.0	08B	1143223.0	
4	911	10508699	HZ250455	05/03/2016 10:00:00 PM	003XX N LOTUS AVE	0820	THEFT	$500 AND UNDER	RESIDENCE	False	...	28.0	25.0	06	1139890.0	

5 rows × 23 columns

E. Figure 8: Chicago Crime data 2012-2017

Various visualization, preprocessing, and machine learning libraries were imported. The Chicago crime dataset from 2012 to 2017 was displayed. The FBI code, location, crime kinds, case numbers, and other details are discovered.

F. Preprocessing

```
In [6]: # Preprocessing
        # Remove NaN Value (As Dataset is huge, the NaN row could be neglectable)
        df = df.dropna()

In [7]: # As the dataset is too huge is size, we would just subsampled a dataset for modelling as proof of concept
        df = df.sample(n=100000)

In [8]: # Remove irrelevant/not meaningfull attributes
        df = df.drop(['Unnamed: 0'], axis=1)
        df = df.drop(['ID'], axis=1)
        df = df.drop(['Case Number'], axis=1)

        df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 100000 entries, 612597 to 209248
Data columns (total 20 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   Date                  100000 non-null  object
 1   Block                 100000 non-null  object
 2   IUCR                  100000 non-null  object
```

G. Figure 9: Data preprocessing

Data preparation was completed. The null value was deleted from the dataset, as were any undesirable or extraneous columns. Furthermore, the dataset was far too vast to be analyzed hence was reduced from 1.4 million to 100000.

H. Visualizing

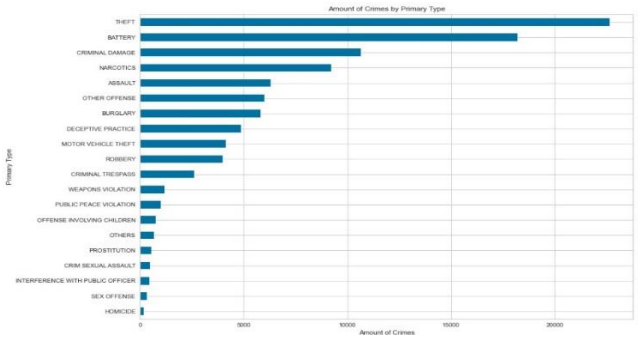

Figure 10: Chicago crime 2012-2017 bar plot

In the years 2012-2017, a bar plot of crimes in Chicago was plotted, and it was discovered that theft was the most common crime and homicide was the least common.

I. Choosing Models

```
In [19]: # At Current Point, the attributes is select manually based on Feature Selection Part.
         Features = ["IUCR", "Description", "FBI Code"]
         print('Full Features: ', Features)

Full Features:  ['IUCR', 'Description', 'FBI Code']

In [20]: #Split dataset to Training Set & Test Set
         x, y = train_test_split(df,
                                 test_size = 0.2,
                                 train_size = 0.8,
                                 random_state= 3)

         x1 = x[Features]     #Features to train
         x2 = x[Target]       #Target Class to train
         y1 = y[Features]     #Features to test
         y2 = y[Target]       #Target Class to test

         print('Feature Set Used    : ', Features)
         print('Target Class        : ', Target)
         print('Training Set Size   : ', x.shape)
         print('Test Set Size       : ', y.shape)

Feature Set Used    :  ['IUCR', 'Description', 'FBI Code']
Target Class        :  Primary Type
Training Set Size   :  (80000, 23)
Test Set Size       :  (20000, 23)
```

Figure 11: Splitting the dataset

Feature selection was used to establish the dependent and target variables for various classification and regression tasks. Following that, the dataset was partitioned into training and testing datasets, each having 80% and 20% of the total.

1. Random Forest

```
In [21]: # Random Forest
         # Create Model with configuration
         rf_model = RandomForestClassifier(n_estimators=70, # Number of trees
                                           min_samples_split = 30,
                                           bootstrap = True,
                                           max_depth = 50,
                                           min_samples_leaf = 25)
         # Model Training
         rf_model.fit(X=x1,
                      y=x2)
         # Prediction
         result = rf_model.predict(y[Features])

In [22]: # Model Evaluation
         ac_sc = accuracy_score(y2, result)
         rc_sc = recall_score(y2, result, average="weighted")
         pr_sc = precision_score(y2, result, average="weighted")
         f1_sc = f1_score(y2, result, average='micro')
         confusion_m = confusion_matrix(y2, result)

         print("========== Random Forest Results ==========")
         print("Accuracy    : ", ac_sc)
         print("Recall      : ", rc_sc)
         print("Precision   : ", pr_sc)
         print("F1 Score    : ", f1_sc)
         print(confusion_m)

========== Random Forest Results ==========
Accuracy    :  0.99535
Recall      :  0.99535
Precision   :  0.9953187840443487
F1 Score    :  0.99535
```

Figure 12: Random Forest

The features and target data were fitted into the model after building a random forest model setup. Following the prediction, the model was evaluated. The model's accuracy was determined to be 99.5 percent. After that a classification report of Random Forest was generated.

Figure 13: Random Forest classification report

2. Neural Network

```
In [24]:
# Neural Network
# Create Model with configuration
nn_model = MLPClassifier(solver='adam',
                         alpha=1e-5,
                         hidden_layer_sizes=(40,),
                         random_state=1,
                         max_iter=1000
                        )
# Model Training
nn_model.fit(X=x1,
             y=x2)
# Prediction
result = nn_model.predict(y[Features])
```

```
In [25]:
# Model Evaluation
ac_sc = accuracy_score(y2, result)
rc_sc = recall_score(y2, result, average="weighted")
pr_sc = precision_score(y2, result, average="weighted")
f1_sc = f1_score(y2, result, average='micro')
confusion_m = confusion_matrix(y2, result)

print("========== Neural Network Results ==========")
print("Accuracy     : ", ac_sc)
print("Recall       : ", rc_sc)
print("Precision    : ", pr_sc)
print("F1 Score     : ", f1_sc)
print("Confusion Matrix: ")
print(confusion_m)
```

```
========== Neural Network Results ==========
Accuracy     :  0.9805
Recall       :  0.9805
Precision    :  0.9815380906667347
F1 Score     :  0.9805
```

Figure 14: Neural Network

The features and target data were fitted into the model after building a neural network model setup. Following the prediction, the model was evaluated. The model's accuracy was determined to be 98 percent. After that a classification report of Neural Network was generated.

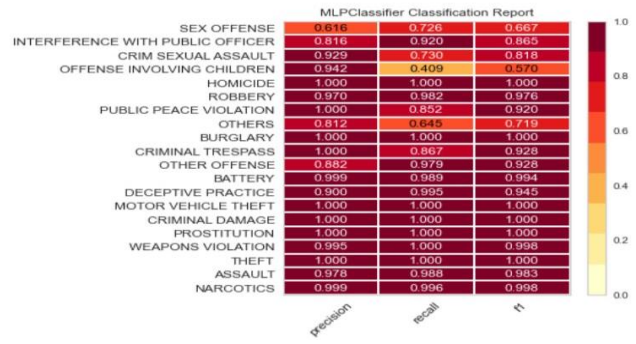

Figure 15: Neural Network Classification report

3. Logistic Regression

```
In [40]:
import warnings
warnings.filterwarnings('ignore')
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(x1,x2,train_size=0.5)
```

```
In [41]:
from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
```

```
In [42]:
model.fit(X_train,Y_train)
```

```
Out[42]:
LogisticRegression()
```

```
In [43]:
model.predict(X_test)
```

```
Out[43]:
array([5, 2, 0, ..., 8, 2, 7], dtype=int64)
```

```
In [44]:
Y_predict=model.predict(X_test)
print('Accuracy of logistic regression classifier on test set is:',((model.score(X_test,Y_test)*100)))
```

```
Accuracy of logistic regression classifier on test set is: 56.832499999999996
```

Figure 16: Logistic Regression

The training and testing data were fitted into the model after building a logistic regression model setup. Following the prediction, the model was evaluated. The model's accuracy was determined to be 56.8 percent.

4. K- Nearest Neighbor

```
In [27]:
# K-Nearest Neighbors
# Create Model with configuration
knn_model = KNeighborsClassifier(n_neighbors=3)
# Model Training
knn_model.fit(X=x1,
              y=x2)
# Prediction
result = knn_model.predict(y[Features])
```

```
In [28]:
# Model Evaluation
ac_sc = accuracy_score(y2, result)
rc_sc = recall_score(y2, result, average="weighted")
pr_sc = precision_score(y2, result, average="weighted")
f1_sc = f1_score(y2, result, average='micro')
confusion_m = confusion_matrix(y2, result)

print("========== K-Nearest Neighbors Results ==========")
print("Accuracy     : ", ac_sc)
print("Recall       : ", rc_sc)
print("Precision    : ", pr_sc)
print("F1 Score     : ", f1_sc)
print("Confusion Matrix: ")
print(confusion_m)
```

```
========== K-Nearest Neighbors Results ==========
Accuracy     :  0.9994
Recall       :  0.9994
Precision    :  0.9994008401571886
F1 Score     :  0.9994
```

Figure 17: K- Nearest Neighbor

The features and target data were fitted into the model after building a KNN model setup. Following the prediction, the model was evaluated. The model's accuracy was determined to be 99.9 percent. After that a classification report of KNN was generated.

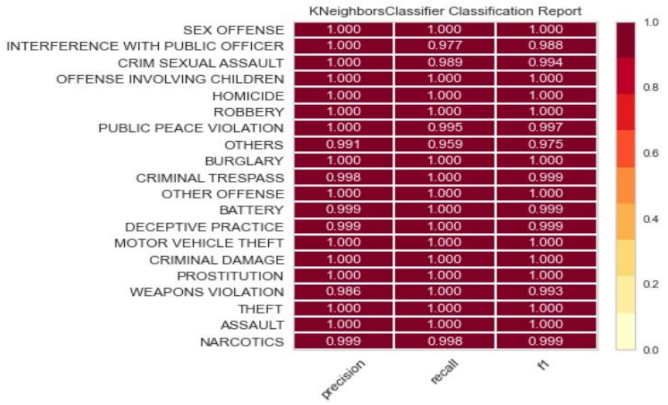

Figure 18: KNN classification report

J. Analysis Result & Conclusion

Previous research has demonstrated that crime forecasting and prediction accuracy can be obtained utilizing learning models. As a result, this study applied numerous machine learning algorithms to better suit the crime data. To forecast region, we utilized Random Forest, KNN, Neural Network, Linear Regression, and Linear Regression, and the accuracy dropped as expected. Random Forest, Neural Network, Logistic Regression, and K-Nearest Neighbor all had accuracy of 99.5, 98, 56, and 99.9%, respectively, with KNN performing the best in predicting crime. The performance of Linear Regression was calculated, and the mean squared error result was considerably too high, at 130. Because the lower the MSE, the better the model, we found that the result was too high and that the model for the dataset was not performing well. Based on the plotting, it can be seen that Theft is the most occurred crime between 2012-2016.

K. Implementing MongoDB

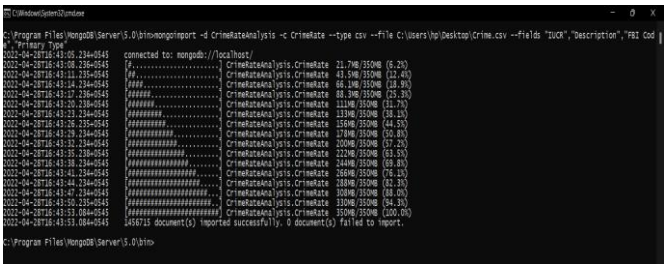

Figure 19: Importing the dataset into MongoDB

The crime dataset of Chicago was imported into the MongoDB database by creating a collection.

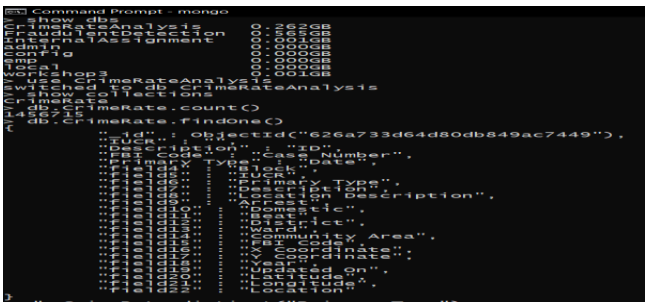

Figure 20: Analysis done in MongoDB terminal

The analysis was carried out in both the MongoDB terminal and Python using the "pymongo" packages. Basic analyses were carried out, such as counting the number of documents, displaying information from a single document, and so on.

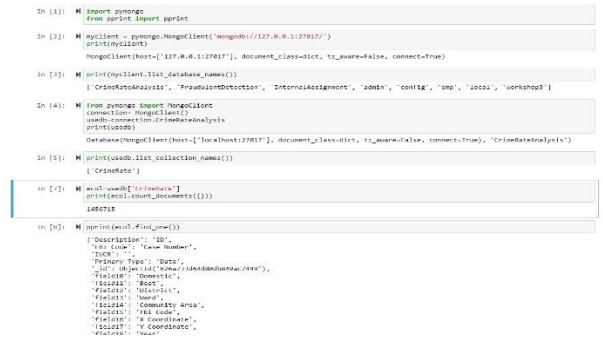

Figure 21: Analysis done in python

L. Implementing Apache Spark

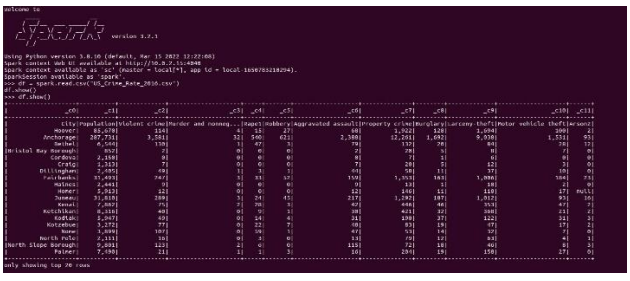

Figure 22: Reading 2016 crime data in Apache Spark

The csv dataset of 2016 was read into the Apache spark shell and some basic SQL queries were performed such as, groupBy, displaying data on some conditions, changing values etc.

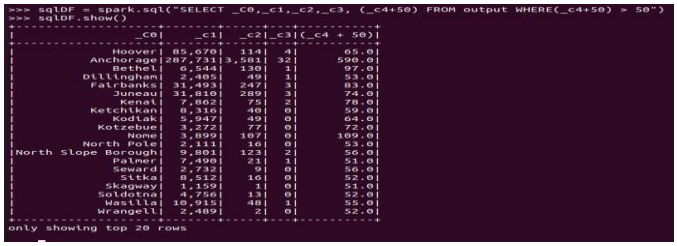

Figure 23: SQL queries in Apache spark

M. Implementing Hadoop

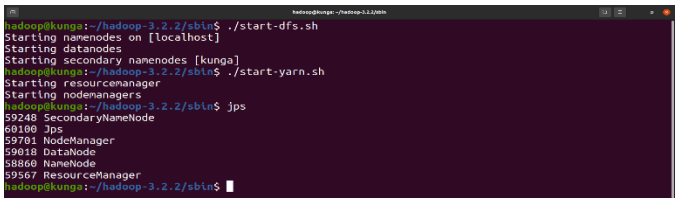

Figure 24: Starting the hadoop nodes

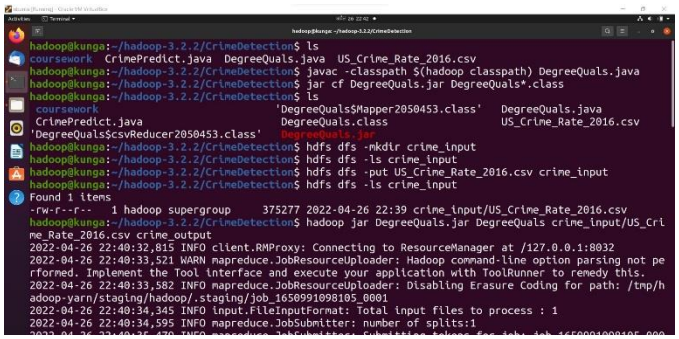

Figure 25: storing the data in dfs, making input and output directory, performing wordcount program

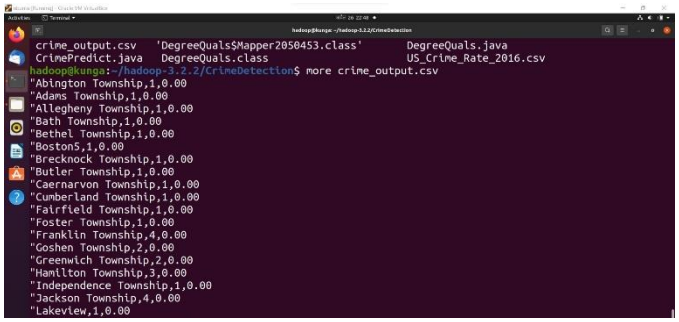

Figure 26: Displaying output of a wordcount program

Some queries were performed in hadoop like creating and starting the hadoop nodes, creating jar files, storing the data into dfs, creating input and output directory and displaying the result in the output directory after applying a simple word count program.

REFRENCES

[1] J. H. B. IV and O. Gallupe, "Has COVID-19 Changed Crime? Crime Rates in the United States during the Pandemic," *American Journal of Criminal Justice*, vol. 45, p. 537–545, 2020.

[2] Statista Research Department, "statista.com," statista, 2021. [Online]. Available: https://www.statista.com/statistics/191129/reported-violent-crime-in-the-us-since-1990/. [Accessed 15 March 2022].

[3] R. Lin, Z. Ye, H. Wang and B. Wu, "Chronic Diseases and Health Monitoring Big Data: A Survey," *IEEE Reviews in Biomedical Engineering*, vol. 11, no. DOI: 10.1109/RBME.2018.2829704, pp. 275 - 288, 2018.

[4] A. Khanna, D. Gupta and N. Dey, Applications of Big Data in Healthcare, 1st ed., Delhi, India: Elsevier Wordmark, 2021.

[5] Monika and A. Bhat, "An analysis of Crime data under Apache Pig on Big Data," in *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, India, 2019.

[6] A. Londhe and P. P. Rao, "Platforms for big data analytics: Trend towards hybrid era," in *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, Chennai, India, 2017.

[7] S. Sandagiri, B. Kumara and B. Kuhaneswaran, "Detecting Crime Related Twitter Posts using Artificial Neural Networks based Approach," in *2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer)*, Colombo, Sri Lanka, 2020.

[8] S. Sandagiri, B. Kumara and B. Kuhaneswaran, "Detecting Crimes Related Twitter Posts using SVM based Two Stages Filtering," in *2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS)*, RUPNAGAR, India, 2020.

[9] R. Yadav and S. K. Sheoran, "Crime Prediction Using Auto Regression Techniques for Time Series Data," in *2018 3rd International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE)*, Jaipur, India, 2018.

[10] P. Das and A. K. Das, "Behavioural analysis of crime against women using a graph based clustering approach," in *2017 International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, India, 2017.

[11] C. Chauhan and S. Sehgal, "A review: Crime analysis using data mining techniques and algorithms," in *2017 International Conference on Computing, Communication and Automation (ICCCA)*, Greater Noida, India, 2017.

[12] S. Yadav, M. Timbadia, A. Yadav, R. Vishwakarma and N. Yadav, "Crime pattern detection, analysis & prediction," in *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India, 2017.

[13] F. Yi, Z. Yu, F. Zhuang, X. Zhang and H. Xiong, "An Integrated Model for Crime Prediction Using Temporal and Spatial Factors," in *2018 IEEE International Conference on Data Mining (ICDM)*, Singapore, 2018.

[14] N. Nafi'iyah and K. F. Mauladi, "Linear Regression Analysis and SVR in Predicting Motor Vehicle Theft," in *2021 International Seminar on Application for Technology of Information and Communication (iSemantic)*, Semarangin, Indonesia, 2021.

[15] S. Zhang, X. Li, M. Zong, X. Zhu and R. Wan, "Efficient kNN Classification With Different Numbers of Nearest Neighbors," *IEEE Transactions on Neural Networks and Learning Systems* , vol. 29, no. 5, pp. 1774-1785, 2018.

[16] D. M. Raza and D. B. Victor, "Data mining and Region Prediction Based on Crime Using Random Forest," in *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, Coimbatore, India, 2021.

[17] O. I. Abiodun, A. Jantana, A. E. Omolara, K. V. Dada, N. A. Mohamed and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. IV, no. 11, 2018.

[18] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada and A. M. Umar, "Comprehensive Review of Artificial Neural Network Applications to Pattern Recognition," in *IEEE Access*, 2019.

[19] L. Lei, "Prediction of Score of Diabetes Progression Index Based on Logistic Regression Algorithm," in *2020 International Conference on Virtual Reality and Intelligent Systems (ICVRIS)*, Zhangjiajie, China, 2020.

[20] M. M. Patil, A. Hanni, C. H. Tejeshwar and P. Patil, "A qualitative analysis of the performance of MongoDB vs MySQL database based on insertion and retriewal operations using a web/android application to explore load balancing — Sharding in MongoDB and its advantages," in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, India, 2017.

[21] P. Merla and Y. Liang, "Data analysis using hadoop MapReduce environment," in *2017 IEEE International Conference on Big Data (Big Data)*, Boston, MA, USA, 2017.

[22] G. Gousios, "Big Data Software Analytics with Apache Spark," in *2018 IEEE/ACM 40th International Conference on Software Engineering: Companion (ICSE-Companion)*, Gothenburg, Sweden, 2018.