DATA ANALYSIS ON DATASETS INVOLVING CRIMES RECORDED BY THE UK POLICE DEPARTMENT

BY

CHIBUIKE DONALD IBE

TABLE OF CONTENTS

TABLE OF CONTENTS ..... **ii**

LIST OF FIGURES ..... **iv**

INTRODUCTION **1**

PART A **1**

1.1 PERFORMING ANALYSIS USING PANDAS AND MATPLOTLIB 2

1.2 MACHINE LEARNING ALGORITHMS FOR CLASSIFICATION 20

1.3 ANALYSIS USING SPARK 23

1.4 ANALYSIS USING HIVE 28

1.5 ANALYSIS USING MAPREDUCE 29

1.6 VISUALIZATION USING POWERBI 29

PART B **31**

2.1 A DEEP DIVE INTO THE SCRIPT FOR ANALYZING CRIME DATA 31

2.2 IMPORTING LIBRARIES 31

2.3 LOADING DATASETS 31

2.4 MERGING DATASETS 31

2.5 BASIC DATA EXPLORATION 32

2.6 DATA CLEANING 33

2.7 ADVANCED ANALYSIS 33

2.7.1 CRIME CATEGORIES INVESTIGATION 33

2.7.2 GEOGRAPHICAL ANALYSIS 34

2.7.3 VISUALIZATION 34

2.8 IDENTIFYING AND CORRECTING DATA ANOMALIES 34

2.8 DESCRIPTIVE STATISTICS USING df.describe() 35

2.9 IDENTIFYING AND RESOLVING OUTLIERS IN "NUMBER_OF_OFFENCES" 35

2.10 RESEARCH QUESTION 1 36

2.10.1 INSIGHTS FROM RESEARCH QUESTION 1 37

2.10.2 KEY OBSERVATIONS 39

2.11 RESEARCH QUESTION 2 39

2.11.1 FINDINGS 39

Data Quality Check 40
2.11.2 KEY INSIGHTS 40
2.12 RESEARCH QUESTION 3 41
2.12.1 KEY INSIGHTS 42
2.12.2 KEY INSIGHTS 42
2.13 UNDERSTANDING MACHINE LEARNING FOR CRIME CLASSIFICATION 43
2.13.1 DROPPING UNNECESSARY COLUMNS 43
2.13.2 ENCODING CATEGORICAL VALUES 44
2.13.3 SPILITTING INTO TRAINING AND TESTING SET 44
2.13.4 IMPORTING MACHINE LEARNING ALGORITHMS 44
2.13.5 BUILDING AND EVALUATING DECISION TREE CLASSIFIER 45
2.13.6 BUILDING AND EVALUATING RANDOM FOREST CLASSIFIER 45
2.13.6 BUILDING AND EVALUATING SUPPORT VECTOR MACHINES (SVM) 45
2.13.7 CONCLUSION 46
REFERENCES 47

LIST OF FIGURES

Figure 1: Picture showing description..... 12
Figure 2: Picture showing the Negative Number of Offences..... 13
Figure 3:Research Question 1..... 15
Figure 4: Research Question 2..... 17
Figure 5: Research Question 3..... 20
Figure 6: New Data Frame..... 21
Figure 7: After Encoding..... 21
Figure 8: Loading of Datasets 24
Figure 9: Merging Datasets 24
Figure 10: Dropping rows with a Negative number of offenses 25
Figure 11: Research Question 1..... 25
Figure 12: Research Question 2..... 26
Figure 13: Research Question 3(A) 26
Figure 14: Research Question 3(B) 27
Figure 15: Research Question 3(C)..... 27
Figure 16: Code using HIVE..... 28
Figure 17: Output using HIVE 28
Figure 18: Using MapReduce..... 29
Figure 19: Research Question 1..... 29
Figure 20: Research Question 3..... 30

INTRODUCTION

PART A

The dataset is a comprehensive compilation of information concerning criminal acts from 2012–2013 to 2023–2024 (statista, 2012-2020). Every column in the data contains significant details about each incidence that has been documented.

I. **Year of Finance**: To pinpoint the precise financial year in which the offenses occurred; this column employs a range of 2012–13 to 2023–24.

II. **Quarterly Finance:** Offenses are categorized based on the fiscal quarter in which they occurred, using a value range of 1 to 4.

III. **Name of Force:** A variety of law enforcement agencies or forces, such as Bedfordshire, Avon and Somerset, Action Fraud, Cambridgeshire, British Transport Police, and Cheshire, have reported offenses to the dataset.

IV. **Description of Offense:** This section clarifies the specific illegal activities that are involved in the offense and provides a comprehensive explanation of its features (criminal justice hub, 2021).

V. **Offense Group:** Offenses include fraud, assaults on persons, theft, arson and criminal damage, drug offenses, gun possession, sexual offenses, public order offenses, and robbery are all categorized into different groups.

VI. **Offence Subgroup:** This section further refines the classification by offering more

specific details about the type of offense within each group.

VII. **Offence Code:** Each offense is assigned a unique code, which offers a consistent and practical means of identification.

VIII. **Number of Offences:** This numerical column quantifies the frequency or total number of crimes, allowing for statistical analysis.

1.1 PERFORMING ANALYSIS USING PANDAS AND MATPLOTLIB

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

 # to import the dataset

```
df1 = pd.read_csv("/content/20122013.csv")
df2 = pd.read_csv("/content/20132014.csv")
df3 = pd.read_csv("/content/20142015.csv")
df4 = pd.read_csv("/content/20152016.csv")
df5 = pd.read_csv("/content/20162017.csv")
df6 = pd.read_csv("/content/20172018.csv")
df7 = pd.read_csv("/content/20182019.csv")
df8 = pd.read_csv("/content/20192020.csv")
df9 = pd.read_csv("/content/20202021.csv")
```

```
df10 = pd.read_csv("/content/20212022.csv")
df11 = pd.read_csv("/content/20222023.csv")
df12 = pd.read_csv("/content/20232024.csv", encoding = "latin-1")
```

DATA PREPROCESSING

#merging all datasets together

```
df = pd.concat([df1,df2,df3,df4,df5,df6,df7,df8,df9,df10,df11,df12])
```

```
df.shape
```

```
(258905, 8)
```

```
df.columns
```

```
Index(['Financial_Year', 'Financial_Quarter', 'Force_Name',
 'Offence_Description', 'Offence_Group', 'Offence_Subgroup',
 'Offence_Code', 'Number_of_Offences'],
 dtype='object')
```

df.info()

#provides information

<class 'pandas.core.frame.DataFrame'>

Int64Index: 258905 entries, 0 to 6382

Data columns (total 8 columns):

 # Column Non-Null Count Dtype

-- ------- ---------- ------

0 **Financial_Year** 258905 non-null object

1 **Financial_Quarter** 258905 non-null int64

2 **Force_Name** 258905 non-null object

3 **Offence_Description** 258905 non-null object

4 **Offence_Group** 258905 non-null object

5 **Offence_Subgroup** 258905 non-null object

6 **Offence_Code** 258905 non-null object

7 **Number_of_Offences** 258905 non-null int64

dtypes: int64(2), object(6)

memory usage: 17.8+ MB

#checking for missing values

df.isnull().sum()

Financial_Year 0

Financial_Quarter 0

Force_Name 0

Offence_Description 0

Offence_Group 0

Offence_Subgroup 0

Offence_Code 0

Number_of_Offences 0

dtype: int64

From the analyses, I was able to discover that there was no missing data using Pandas.

#Checking the data type for Number of Offences

df["Number_of_Offences"].dtype

dtype('int64')

DATA CLEANING

df["Financial_Year"].unique()

array(['2012/13', '2013/14', '2014/15', '2015/16', '2016/17', '2017/18',
'2018/19', '2019/20', '2020/21', '2021/22', '2022/23', '2023/24'],
dtype=object)

df["Financial_Quarter"].unique()

array([1, 2, 3, 4])

df["Force_Name"].unique()

array(['Action Fraud', 'Avon and Somerset', 'Bedfordshire',

'British Transport Police', 'Cambridgeshire', 'Cheshire', 'CIFAS',
'Cleveland', 'Cumbria', 'Derbyshire', 'Devon and Cornwall',
'Dorset', 'Durham', 'Dyfed-Powys', 'Essex',
'Financial Fraud Action UK', 'Gloucestershire',
'Greater Manchester', 'Gwent', 'Hampshire', 'Hertfordshire',
'Humberside', 'Kent', 'Lancashire', 'Leicestershire',
'Lincolnshire', 'London, City of', 'Merseyside',
'Metropolitan Police', 'Norfolk', 'North Wales', 'North Yorkshire',
'Northamptonshire', 'Northumbria', 'Nottinghamshire',
'South Wales', 'South Yorkshire', 'Staffordshire', 'Suffolk',
'Surrey', 'Sussex', 'Thames Valley', 'Warwickshire', 'West Mercia',
'West Midlands', 'West Yorkshire', 'Wiltshire', 'Cifas',
'UK Finance'], dtype=object)

```
df["Offence_Description"].unique()
```

```
array(['Fraud offences recorded by Action Fraud',
 'Abandoning child under two years (outcomes only)',
 'Absconding from lawful custody',
 'Abuse of children through prostitution and pornography',
 'Abuse of position of trust of a sexual nature',
 'Actually bodily harm and other injury(outcomes only)',
 'Adulteration of food (outcomes only)',
 'Aggravated burglary in a building other than a dwelling',
 'Aggravated burglary in a dwelling', 'Aggravated vehicle taking',
 'Aiding suicide', 'Arson endangering life',
 'Arson not endangering life', 'Assault with injury',
 'Assault with intent to cause serious harm',
```

'Assault without injury', 'Assault without injury on a constable',
'Attempted burglary in a building other than a dwelling',
'Attempted burglary in a dwelling',
'Attempted distraction burglary in a dwelling', 'Attempted murder',
'Bail offences', 'Bankruptcy and insolvency (outcomes only)',
'Betting, gaming and lotteries (outcomes only)', 'Bigamy',
'Blackmail', 'Burglary in a building other than a dwelling',
'Burglary in a dwelling', 'Causing death by aggravated vehicle taking',
'Causing death by careless driving under influence of drink or drugs',
'Causing death by careless or inconsiderate driving',
'Causing death by dangerous driving',
'Causing death by driving: unlicensed drivers etc.',
'Causing or allowing death of child or vulnerable person',
'Causing sexual activity without consent', 'Child abduction',
'Concealing an infant death close to birth',
'Conspiracy to murder',
'Criminal damage to a building other than a dwelling',
'Criminal damage to a dwelling', 'Criminal damage to a vehicle',
'Cruelty to and neglect of children (outcomes only)',
'Cruelty to children/young persons',
'Customs and Revenue offences (outcomes only)',
'Dangerous driving',
'Disclosure, obstruction, false or misleading statements etc',
'Dishonest use of electricity',
'Distraction burglary in a dwelling', 'Endangering life',
'Endangering life at sea (outcomes only)',
'Endangering railway passengers (outcomes only)',
'Exploitation of prostitution', 'Exposure and voyeurism',

'False accounting (outcomes only)',

'Forgery or use of false drug prescription',

'Fraud by abuse of position (outcomes only)',

'Fraud by company director (outcomes only)',

'Fraud by failing to disclose information (outcomes only)',

'Fraud by false representation: cheque, plastic card and online bank accounts (not PSP) (outcomes

only)',

'Fraud by false representation: other frauds (outcomes only)',

'Fraud, forgery etc associated with vehicle or driver records',

'Going equipped for stealing, etc', 'Handling stolen goods',

'Harassment', 'Health and Safety offences (outcomes only)',

'Homicide', 'Immigration Acts (outcomes only)',

'Incest or familial sexual offences',

'Inflicting grievous bodily harm without intent (outcomes only)',

'Intentional destruction of a viable unborn child',

'Interfering with a motor vehicle', 'Kidnapping',. ..]

It was observed that two Offence description were wrongly written

1. Attempted residential burglary of\xa0unconnected building

2. Residential burglary of\xa0unconnected building

Applying corrections

replace_dict = {"Attempted residential burglary of\xa0unconnected building": "Attempted residential burglary of unconnected building", "Residential burglary of\xa0unconnected building" : "Residential burglary of unconnected building"}

df["Offence_Description"].replace(replace_dict, inplace=True)

```
df["Offence_Description"].unique()
```

```
array(['Fraud offences recorded by Action Fraud',
 'Abandoning child under two years (outcomes only)',
 'Absconding from lawful custody',
 'Abuse of children through prostitution and pornography',
 'Abuse of position of trust of a sexual nature',
 'Actually bodily harm and other injury(outcomes only)',
 'Adulteration of food (outcomes only)',
 'Aggravated burglary in a building other than a dwelling',
 'Aggravated burglary in a dwelling', 'Aggravated vehicle taking',
 'Aiding suicide', 'Arson endangering life',
 'Arson not endangering life', 'Assault with injury',
 'Assault with intent to cause serious harm',
 'Assault without injury', 'Assault without injury on a constable',
 'Attempted burglary in a building other than a dwelling',
 'Attempted burglary in a dwelling',
 'Attempted distraction burglary in a dwelling', 'Attempted murder',
 'Bail offences', 'Bankruptcy and insolvency (outcomes only)', 'Betting, gaming and
lotteries (outcomes only)', 'Bigamy',
 'Blackmail', 'Burglary in a building other than a dwelling',
 'Burglary in a dwelling',
 'Causing death by aggravated vehicle taking',
 'Causing death by careless driving under influence of drink or drugs',
 'Causing death by careless or inconsiderate driving',
 'Causing death by dangerous driving',
 'Causing death by driving: unlicensed drivers etc.',
```

'Causing or allowing death of child or vulnerable person',
'Causing sexual activity without consent', 'Child abduction',
'Concealing an infant death close to birth',
'Conspiracy to murder',
'Criminal damage to a building other than a dwelling',
'Criminal damage to a dwelling', 'Criminal damage to a vehicle',
'Cruelty to and neglect of children (outcomes only)',
'Cruelty to children/young persons',
'Customs and Revenue offences (outcomes only)',
'Dangerous driving',
'Disclosure, obstruction, false or misleading statements etc',
'Dishonest use of electricity',
'Distraction burglary in a dwelling', 'Endangering life',
'Endangering life at sea (outcomes only)',
'Endangering railway passengers (outcomes only)',
'Exploitation of prostitution', 'Exposure and voyeurism',
'False accounting (outcomes only)',
'Forgery or use of false drug prescription',
'Fraud by abuse of position (outcomes only)',
'Fraud by company director (outcomes only)',
'Fraud by failing to disclose information (outcomes only)',
'Fraud by false representation: cheque, plastic card and online bank accounts (not PSP) (outcomes
only)',
'Fraud by false representation: other frauds (outcomes only)',
'Fraud, forgery etc associated with vehicle or driver records',
'Going equipped for stealing, etc', 'Handling stolen goods',
'Harassment', 'Health and Safety offences (outcomes only)',

```
 'Homicide', 'Immigration Acts (outcomes only)',
 'Incest or familial sexual offences',
 'Inflicting grievous bodily harm without intent (outcomes only)',
 'Intentional destruction of a viable unborn child',
 'Interfering with a motor vehicle', 'Kidnapping',
 'Libel (outcomes only)', 'Making off without payment',.....]

df["Offence_Group"].unique()

array(['Fraud offences', 'Violence against the person',
 'Miscellaneous crimes against society', 'Sexual offences',
 'Theft offences', 'Criminal damage and arson', 'Drug offences',
 'Possession of weapons offences', 'Public order offences',
 'Robbery'], dtype=object)

df["Offence_Subgroup"].unique()

array(['Fraud: Action Fraud', 'Violence without injury',
 'Miscellaneous crimes against society', 'Other sexual offences',
 'Violence with injury', 'Non-domestic burglary',
 'Domestic burglary', 'Theft of a motor vehicle', 'Arson',
 'Fraud offences to 2012/13', 'Other theft offences',
 'Criminal damage', 'Homicide', 'Vehicle interference',
 'Possession of drugs', 'Possession of weapons offences',
 'Public order offences', 'Rape offences',
 'Robbery of business property', 'Robbery of personal property',
 'Shoplifting', 'Theft from the person', 'Theft from a vehicle',
 'Bicycle theft', 'Trafficking of drugs', 'Fraud: CIFAS',
```

'Fraud: Financial Fraud Action UK',

'Death or serious injury - unlawful driving',

'Stalking and harassment', 'Fraud: action fraud',

'Fraud: UK Finance'], dtype=object)

df.describe()

	Financial_Quarter	Number_of_Offences
count	258905.000000	258905.000000
mean	2.463031	229.901837
std	1.128399	1724.363276
min	1.000000	-268.000000
25%	1.000000	0.000000
50%	2.000000	9.000000
75%	3.000000	83.000000
max	4.000000	132693.000000

Figure 1: Picture showing description

Outliners discovered in Number of Offences

I. We can't have the number of offences being a negative number (<0)

	Financial_Year	Financial_Quarter	Force_Name	Offence_Description	Offence_Group	Offence_Subgroup	Offence_Code	Number_of_Offences
407	2012/13	1	British Transport Police	Preserved other fraud and repealed fraud offen...	Fraud offences	Fraud offences to 2012/13	53B	-1
715	2012/13	1	Cheshire	Preserved other fraud and repealed fraud offen...	Fraud offences	Fraud offences to 2012/13	53B	-3
6835	2012/13	2	Avon and Somerset	Fraud by company director (outcomes only)	Fraud offences	Fraud offences to 2012/13	51	-1
6865	2012/13	2	Avon and Somerset	Perjury	Miscellaneous crimes against society	Miscellaneous crimes against society	67	-1
6878	2012/13	2	Avon and Somerset	Preserved other fraud and repealed fraud offen...	Fraud offences	Fraud offences to 2012/13	53B	-15

Figure 2: Picture showing the Negative Number of Offences

So we have a total of 426 rows with negative numbers for the number of offences

Dropping the negative numbers

#dropping the outliners

```
df = df[df["Number_of_Offences"] >= 0]
```

```
df.shape
```

(258479, 8)

Research Questions

1. To determine the offence group with the most offences.

2. To find number times an offence code appeared.

3. Force Name involved in the highest number of crimes committed.

RESEARCH QUESTION 1

```
df.groupby(df["Offence_Group"])["Number_of_Offences"].sum().sort_value
s()
```

Offence_Group

Possession of weapons offences 419560

Robbery 758697

Miscellaneous crimes against society 964614

Sexual offences 1502567

Drug offences 1956518

Public order offences 4029470

Criminal damage and arson 6040693

Fraud offences 8278687

Violence against the person 15518904

Theft offences 20054697

Name: Number_of_Offences, dtype: int64

From the analyses, it was discovered that **Theft offenses** were the most committed crime with over 20 million crime offenses as **Possession of weapons** offenses came at least with over 400 thousand crime offenses.

#Data Visualization

```
group_rq1=
df.groupby("Offence_Group")["Number_of_Offences"].sum().sort_values()

group_rq1.plot.bar()

plt.ylabel("Sum of Offences")

plt.xlabel("Offence Group")

plt.title("Research Question 1")

plt.show()
```

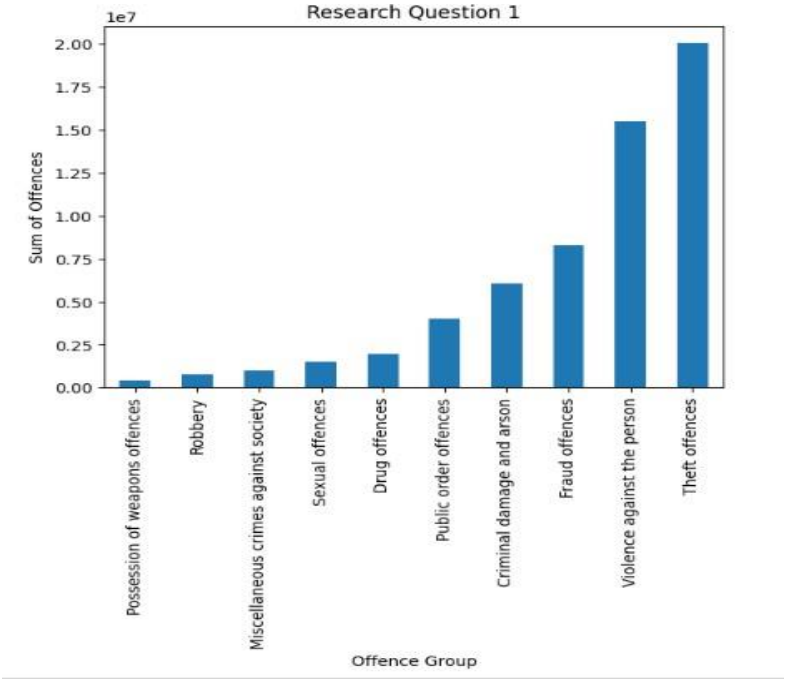


Figure 3:Research Question 1

RESEARCH QUESTION 2

 # To find number times an offence code appeared. This will print the first 10 with the highest

df["Offence_Code"].value_counts().head(10)

4.1 3605

802 1980

54 1980

58B 1980

58A 1980

58C 1980

11A 1980

46 1980

43 1980

28C 1980

Name: Offence_Code, dtype: int64

From the analyses, **4.1** recorded a higher number of occurrences with 3605

#Data Visualization

```
group_rq2 = df["Offence_Code"].value_counts().head(10)
group_rq2.plot(kind = "bar")
plt.ylabel("count")
plt.xlabel("offence code")
plt.title("Research Question 2")
plt.show()
```

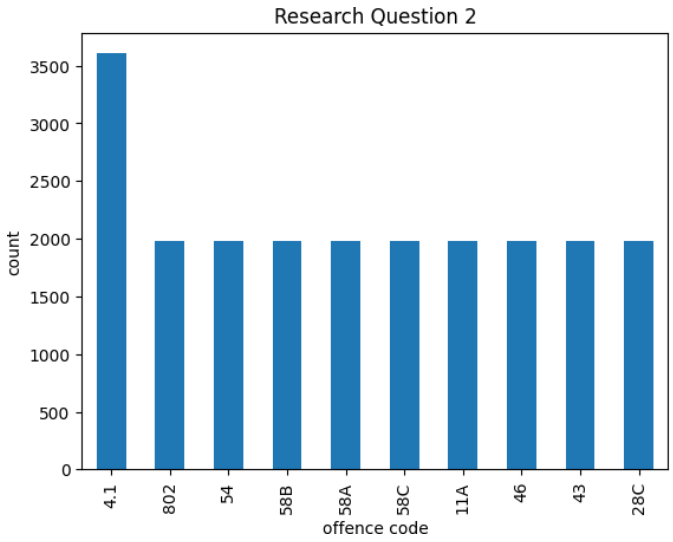


Figure 4: Research Question 2

RESEARCH QUESTION 3

#Finding the Force Name involved in the highest number of crimes committed

df.groupby("Force_Name")["Number_of_Offences"].sum().sort_values()

Force_Name

London, City of 68079

Dyfed-Powys 321478

Cumbria 343605

Warwickshire 408936

Gloucestershire 417484

Wiltshire 443198

North Yorkshire 458475

Lincolnshire 510450

Bedfordshire 518159

Dorset 522780

Financial Fraud Action UK 523390

Gwent 525375

Suffolk 531121

North Wales 560945

Durham 565120

British Transport Police 609130

Norfolk 617978

Northamptonshire 618515

Cleveland 654395

Cambridgeshire 660475

Derbyshire 730685

Surrey 733058

Hertfordshire 793224

Staffordshire 833262

Cheshire 845868

West Mercia 873201

Leicestershire 899636

Humberside 938242

Nottinghamshire 993994

Devon and Cornwall 1045108

Cifas 1077081

South Wales 1103647

UK Finance 1200608

Sussex 1213082

Northumbria 1328330

Lancashire 1344483

Merseyside 1397257

Avon and Somerset 1411830

South Yorkshire 1419554

Essex 1509933

Hampshire 1612454

Thames Valley 1684546

Kent 1693164

CIFAS 2231228

West Yorkshire 2712176

West Midlands 2754135

Greater Manchester 3118929

Action Fraud 3183118

Metropolitan Police 8963486

Name: Number_of_Offences, dtype: int64

The **Metropolitan Police** were involved with the most number of crimes while **London, City of** had the least number of crimes

#Data Visualization

```
group_rq3 =
df.groupby("Force_Name")["Number_of_Offences"].sum().sort_values()

plt.figure(figsize=(10,8))
group_rq3.plot.bar()
plt.ylabel("sum of offences")
plt.xlabel("Force Name")
plt.title("Research Question 3")
plt.show()
```

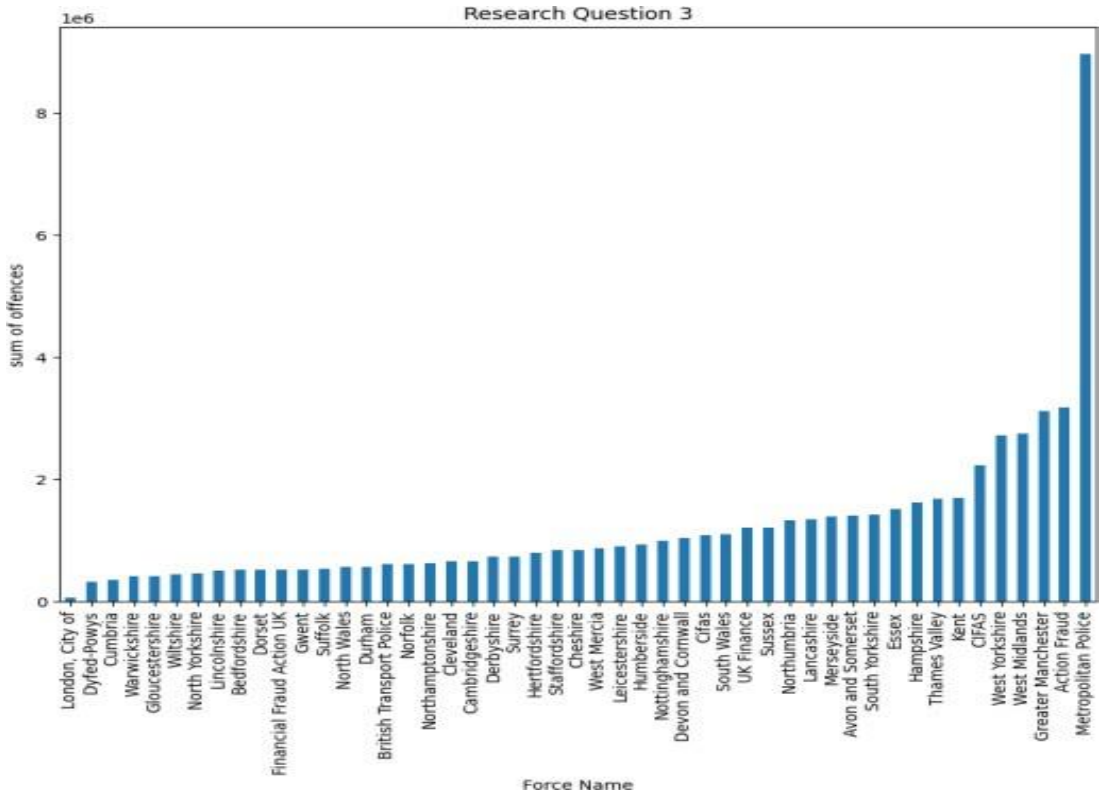


Figure 5: Research Question 3

#saving the merged dataset

df.to_csv("merged.csv", index = False)

1.2 MACHINE LEARNING ALGORITHMS FOR CLASSIFICATION

#Dropping columns that are not necessary for model creation

df.drop(["Financial_Year", "Financial_Quarter", "Offence_Code", "Number_of_Offences"], axis=1, inplace=True)

df.head()

	Force_Name	Offence_Description	Offence_Group	Offence_Subgroup
0	Action Fraud	Fraud offences recorded by Action Fraud	Fraud offences	Fraud: Action Fraud
1	Avon and Somerset	Abandoning child under two years (outcomes only)	Violence against the person	Violence without injury
2	Avon and Somerset	Absconding from lawful custody	Miscellaneous crimes against society	Miscellaneous crimes against society
3	Avon and Somerset	Abuse of children through prostitution and por...	Sexual offences	Other sexual offences
4	Avon and Somerset	Abuse of position of trust of a sexual nature	Sexual offences	Other sexual offences

Figure 6: New Data Frame

#Encoding all categorical values to Numerical values

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

df["Force_Name"] = le.fit_transform(df["Force_Name"])
df["Offence_Description"] = le.fit_transform(df["Offence_Description"])
df["Offence_Group"] = le.fit_transform(df["Offence_Group"])
df["Offence_Subgroup"] = le.fit_transform(df["Offence_Subgroup"])

df.head()
```

	Force_Name	Offence_Description	Offence_Group	Offence_Subgroup
0	0	90	2	6
1	1	0	9	30
2	1	1	3	12
3	1	2	7	14
4	1	4	7	14

Figure 7: After Encoding

#Splitting data

```
from sklearn.model_selection import train_test_split

x = df.drop(["Offence_Group"], axis = 1)
y = df["Offence_Group"]

x_train, x_test, y_train, y_test = train_test_split(x,y, test_size = 0.2,
random_state = 2)
```

#Importing all three ML algorithms for Classification

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
```

#For Decision Tree Classifier

```
dt = DecisionTreeClassifier()
dt_model = dt.fit(x_train,y_train)
dt_model_prediction = dt.predict(x_test)

from sklearn.metrics import accuracy_score

data_accuracy = accuracy_score(dt_model_prediction, y_test)
print(data_accuracy)
```

1.0

#For Random Forest Classfier

```
rf = RandomForestClassifier()
rf_model = rf.fit(x_train,y_train)

rf_model_prediction = rf.predict(x_test)
data_accuracy_rf = accuracy_score(rf_model_prediction, y_test)
print(data_accuracy_rf)
```

1.0

#For Support Vector Machines

```
sv = SVC()
sv_model = sv.fit(x_train,y_train)
sv_model_prediction = sv.predict(x_test)

data_accuracy_sv = accuracy_score(sv_model_prediction, y_test)
print(data_accuracy_sv)
```

0.7574280408542247

1.3 ANALYSIS USING SPARK

I. LOADING OF DATASETS

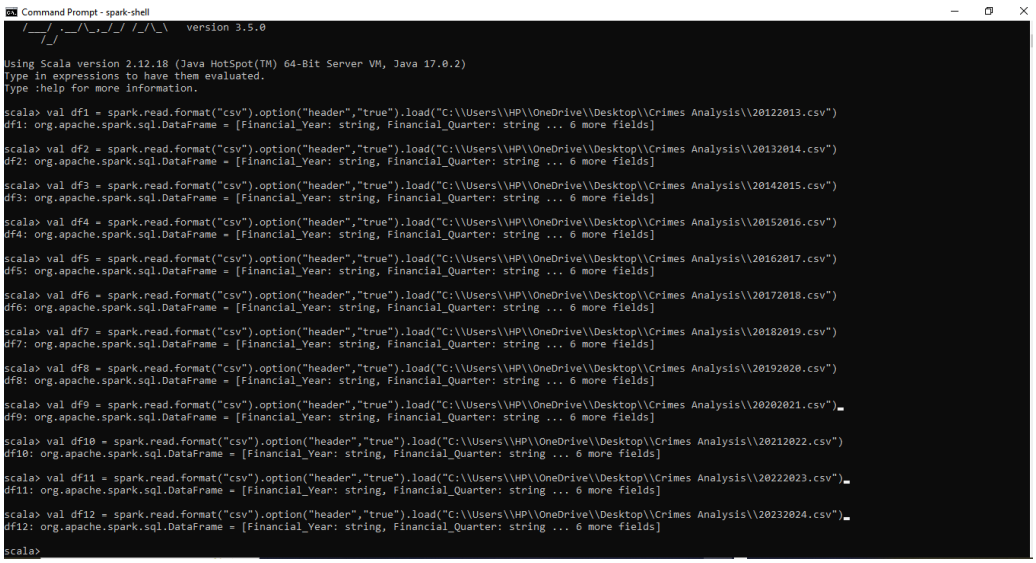


Figure 8: Loading of Datasets

II. MERGING THE DATASETS TOGETHER

The dataset from 2012/13 till 2023/24 are merged together using Full Outer Join.

```
scala> val joinedtable = df1.join(df2, Seq("Financial_Year","Financial_Quarter","Force_Name","Offence_Description","Offence_Group","Offence_Subgroup","Offence_Code","Nu
mber_of_Offences"),"outer").join(df3, Seq("Financial_Year","Financial_Quarter","Force_Name","Offence_Description","Offence_Group","Offence_Subgroup","Offence_Code","Num
ber_of_Offences"),"outer").join(df4, Seq("Financial_Year","Financial_Quarter","Force_Name","Offence_Description","Offence_Group","Offence_Subgroup","Offence_Code","Numb
er_of_Offences"),"outer").join(df5, Seq("Financial_Year","Financial_Quarter","Force_Name","Offence_Description","Offence_Group","Offence_Subgroup","Offence_Code","Numbe
r_of_Offences"),"outer").join(df6, Seq("Financial_Year","Financial_Quarter","Force_Name","Offence_Description","Offence_Group","Offence_Subgroup","Offence_Code","Number
_of_Offences"),"outer").join(df7, Seq("Financial_Year","Financial_Quarter","Force_Name","Offence_Description","Offence_Group","Offence_Subgroup","Offence_Code","Number_
of_Offences"),"outer").join(df8, Seq("Financial_Year","Financial_Quarter","Force_Name","Offence_Description","Offence_Group","Offence_Subgroup","Offence_Code","Number_o
f_Offences"),"outer").join(df9, Seq("Financial_Year","Financial_Quarter","Force_Name","Offence_Description","Offence_Group","Offence_Subgroup","Offence_Code","Number_of
_Offences"),"outer").join(df10, Seq("Financial_Year","Financial_Quarter","Force_Name","Offence_Description","Offence_Group","Offence_Subgroup","Offence_Code","Number_of
_Offences"),"outer").join(df11, Seq("Financial_Year","Financial_Quarter","Force_Name","Offence_Description","Offence_Group","Offence_Subgroup","Offence_Code","Number_of
_Offences"),"outer").join(df12, Seq("Financial_Year","Financial_Quarter","Force_Name","Offence_Description","Offence_Group","Offence_Subgroup","Offence_Code","Number_of
_Offences"),"outer")
joinedtable: org.apache.spark.sql.DataFrame = [Financial_Year: string, Financial_Quarter: string ... 6 more fields]
```

Figure 9: Merging Datasets

III. DROPPING THE ROWS WITH NEGATIVE NUMBER OF OFFENCES

```
scala> val result = joinedtable.filter($"Number_of_Offences" >= 0)
result: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [Financial_Year: string, Financial_Quarter: string ... 6 more fields]

scala> result.show()
+--------------+-----------------+--------------+--------------------+--------------------+--------------------+------------+-----------------+
|Financial_Year|Financial_Quarter|   Force_Name| Offence_Description|       Offence_Group|    Offence_Subgroup|Offence_Code|Number_of_Offences|
+--------------+-----------------+--------------+--------------------+--------------------+--------------------+------------+-----------------+
|       2012/13|                1|  Action Fraud|Fraud offences re...|     Fraud offences| Fraud: Action Fraud|          AF|            20323|
|       2012/13|                1|Avon and Somerset|Absconding from l...|Miscellaneous cri...|Miscellaneous cri...|          80|                1|
|       2012/13|                1|Avon and Somerset|Actually bodily h...|Violence against ...|Violence with injury|          8G|                0|
|       2012/13|                1|Avon and Somerset|Arson endangering...|Criminal damage a...|               Arson|         56A|               25|
|       2012/13|                1|Avon and Somerset| Assault with injury|Violence against ...|Violence with injury|          8N|             2064|
|       2012/13|                1|Avon and Somerset|Assault without i...|Violence against ...|Violence without ...|         104|              140|
|       2012/13|                1|Avon and Somerset|    Attempted murder|Violence against ...|Violence with injury|           2|                2|
|       2012/13|                1|Avon and Somerset|Causing sexual ac...|     Sexual offences|Other sexual offe...|         22A|                1|
|       2012/13|                1|Avon and Somerset|Concealing an inf...|Miscellaneous cri...|Miscellaneous cri...|          15|                0|
|       2012/13|                1|Avon and Somerset|Conspiracy to murder|Violence against ...|Violence without ...|          3A|                0|
|       2012/13|                1|Avon and Somerset|Criminal damage t...|Criminal damage a...|     Criminal damage|         58B|              560|
|       2012/13|                1|Avon and Somerset|Distraction burgl...|      Theft offences|   Domestic burglary|         28C|               22|
|       2012/13|                1|Avon and Somerset|Forgery or use of...|Miscellaneous cri...|Miscellaneous cri...|          60|                2|
|       2012/13|                1|Avon and Somerset|Incest or familia...|     Sexual offences|Other sexual offe...|          23|                7|
|       2012/13|                1|Avon and Somerset|Inflicting grievo...|Violence against ...|Violence with injury|          8F|                0|
|       2012/13|                1|Avon and Somerset|          Kidnapping|Violence against ...|Violence without ...|          36|                6|
|       2012/13|                1|Avon and Somerset|Making or supplyi...|Miscellaneous cri...|Miscellaneous cri...|         53H|               10|
|       2012/13|                1|Avon and Somerset|Offender Manageme...|Miscellaneous cri...|Miscellaneous cri...|          69|               10|
|       2012/13|                1|Avon and Somerset| Other drug offences|       Drug offences| Possession of drugs|         92C|               73|
|       2012/13|                1|Avon and Somerset|Other knives offe...|Possession of wea...|Possession of wea...|          90|                0|
+--------------+-----------------+--------------+--------------------+--------------------+--------------------+------------+-----------------+
only showing top 20 rows
```

Figure 10: Dropping rows with a Negative number of offenses

IV. RESEARCH QUESTION 1

To determine the offense group with the most offenses.

```
scala> val rq1 = result.groupBy("Offence_Group").agg(sum("Number_of_Offences").alias("TotalOffences")).orderBy(desc("TotalOffences"))
rq1: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [Offence_Group: string, TotalOffences: double]

scala> rq1.show()
+--------------------+-------------+
|       Offence_Group|TotalOffences|
+--------------------+-------------+
|      Theft offences|  2.0054697E7|
|Violence against ...|  1.5518904E7|
|      Fraud offences|    8278687.0|
|Criminal damage a...|    6040693.0|
|Public order offe...|    4029470.0|
|       Drug offences|    1956518.0|
|     Sexual offences|    1502567.0|
|Miscellaneous cri...|     964614.0|
|             Robbery|     758697.0|
|Possession of wea...|     419560.0|
+--------------------+-------------+
```

Figure 11: Research Question 1

V. RESEARCH QUESTION 2

To find out the number of times an offence code appeared.

```
scala> val rq2 = result.groupBy("Offence_Code").agg(count("*").alias("OffenceCount")).orderBy(desc("OffenceCount"))
rq2: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [Offence_Code: string, OffenceCount: bigint]

scala> rq2.show()
+------------+------------+
|Offence_Code|OffenceCount|
+------------+------------+
|         4.1|        3605|
|          76|        1980|
|         30A|        1980|
|         58A|        1980|
|          40|        1980|
|         58D|        1980|
|          44|        1980|
|         19E|        1980|
|         28C|        1980|
```

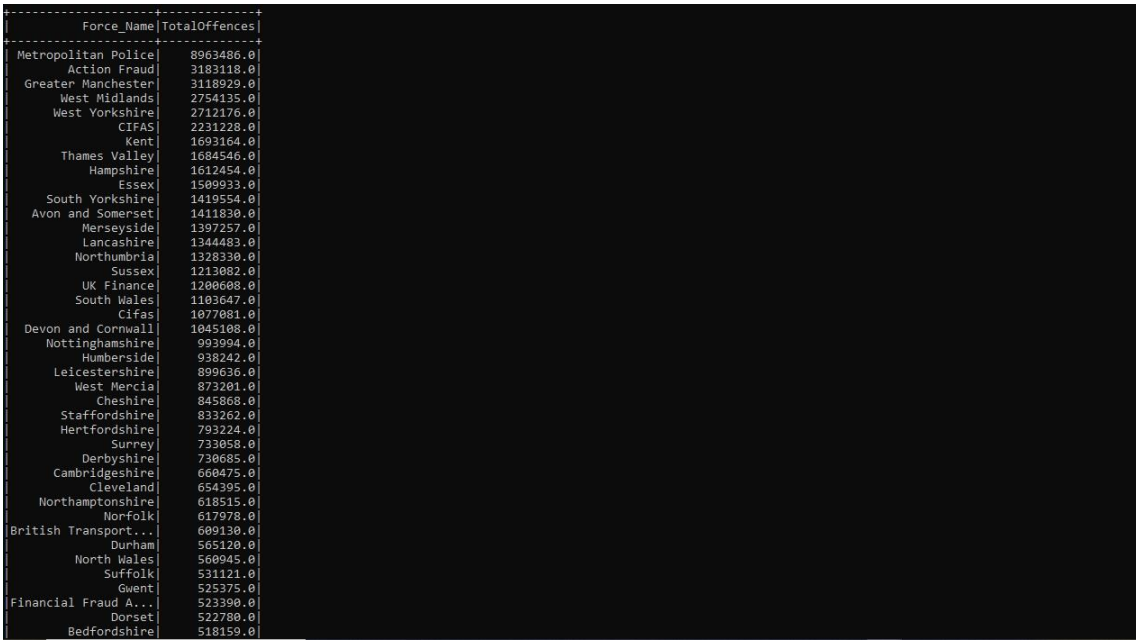
Figure 12: Research Question 2

VI. RESEARCH QUESTION 3

Force Name involved in the highest number of crimes committed.

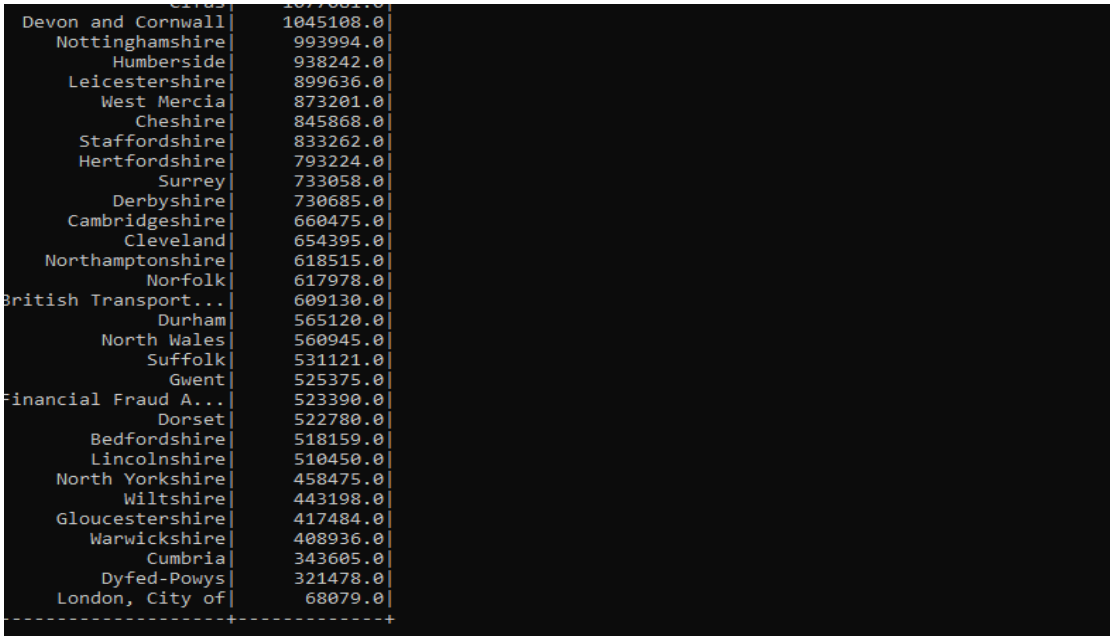
```
scala> val rq3 = result.groupBy("Force_Name").agg(sum("Number_of_Offences").alias("TotalOffences")).orderBy(desc("TotalOffences"))
rq3: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [Force_Name: string, TotalOffences: double]

scala> rq3.show()
+------------------+-------------+
|         Force_Name|TotalOffences|
+------------------+-------------+
|Metropolitan Police|    8963486.0|
|       Action Fraud|    3183118.0|
| Greater Manchester|    3118929.0|
|      West Midlands|    2754135.0|
|     West Yorkshire|    2712176.0|
|              CIFAS|    2231228.0|
|              Kent|    1693164.0|
|      Thames Valley|    1684546.0|
|          Hampshire|    1612454.0|
|             Essex|    1509933.0|
|    South Yorkshire|    1419554.0|
|  Avon and Somerset|    1411830.0|
|         Merseyside|    1397257.0|
|         Lancashire|    1344483.0|
|        Northumbria|    1328330.0|
|            Sussex|    1213082.0|
|         UK Finance|    1200608.0|
|        South Wales|    1103647.0|
|              Cifas|    1077081.0|
| Devon and Cornwall|    1045108.0|
+------------------+-------------+
only showing top 20 rows


scala> rq3.show(50)
+--------------------+-------------+
|          Force_Name|TotalOffences|
+--------------------+-------------+
| Metropolitan Police|    8963486.0|
|        Action Fraud|    3183118.0|
|  Greater Manchester|    3118929.0|
|       West Midlands|    2754135.0|
|      West Yorkshire|    2712176.0|
|               CIFAS|    2231228.0|
```

Figure 13: Research Question 3(A)



Force_Name	TotalOffences
Metropolitan Police	8963486.0
Action Fraud	3183118.0
Greater Manchester	3118929.0
West Midlands	2754135.0
West Yorkshire	2712176.0
CIFAS	2231228.0
Kent	1693164.0
Thames Valley	1684546.0
Hampshire	1612454.0
Essex	1509933.0
South Yorkshire	1419554.0
Avon and Somerset	1411830.0
Merseyside	1397257.0
Lancashire	1344483.0
Northumbria	1328330.0
Sussex	1213082.0
UK Finance	1200608.0
South Wales	1103647.0
Cifas	1077081.0
Devon and Cornwall	1045108.0
Nottinghamshire	993994.0
Humberside	938242.0
Leicestershire	899636.0
West Mercia	873201.0
Cheshire	845868.0
Staffordshire	833262.0
Hertfordshire	793224.0
Surrey	733058.0
Derbyshire	730685.0
Cambridgeshire	660475.0
Cleveland	654395.0
Northamptonshire	618515.0
Norfolk	617978.0
British Transport...	609130.0
Durham	565120.0
North Wales	560945.0
Suffolk	531121.0
Gwent	525375.0
Financial Fraud A...	523390.0
Dorset	522780.0
Bedfordshire	518159.0

Figure 14: Research Question 3(B)



Devon and Cornwall	1045108.0
Nottinghamshire	993994.0
Humberside	938242.0
Leicestershire	899636.0
West Mercia	873201.0
Cheshire	845868.0
Staffordshire	833262.0
Hertfordshire	793224.0
Surrey	733058.0
Derbyshire	730685.0
Cambridgeshire	660475.0
Cleveland	654395.0
Northamptonshire	618515.0
Norfolk	617978.0
British Transport...	609130.0
Durham	565120.0
North Wales	560945.0
Suffolk	531121.0
Gwent	525375.0
Financial Fraud A...	523390.0
Dorset	522780.0
Bedfordshire	518159.0
Lincolnshire	510450.0
North Yorkshire	458475.0
Wiltshire	443198.0
Gloucestershire	417484.0
Warwickshire	408936.0
Cumbria	343605.0
Dyfed-Powys	321478.0
London, City of	68079.0

Figure 15: Research Question 3(C)

1.4 ANALYSIS USING HIVE

To find out the number of times an offence code appeared.

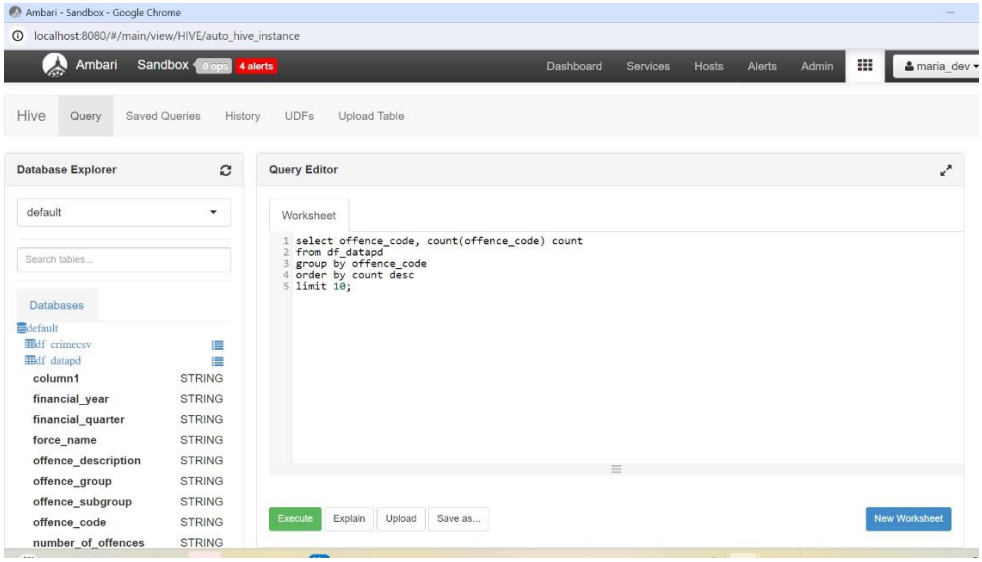


Figure 16: Code using HIVE

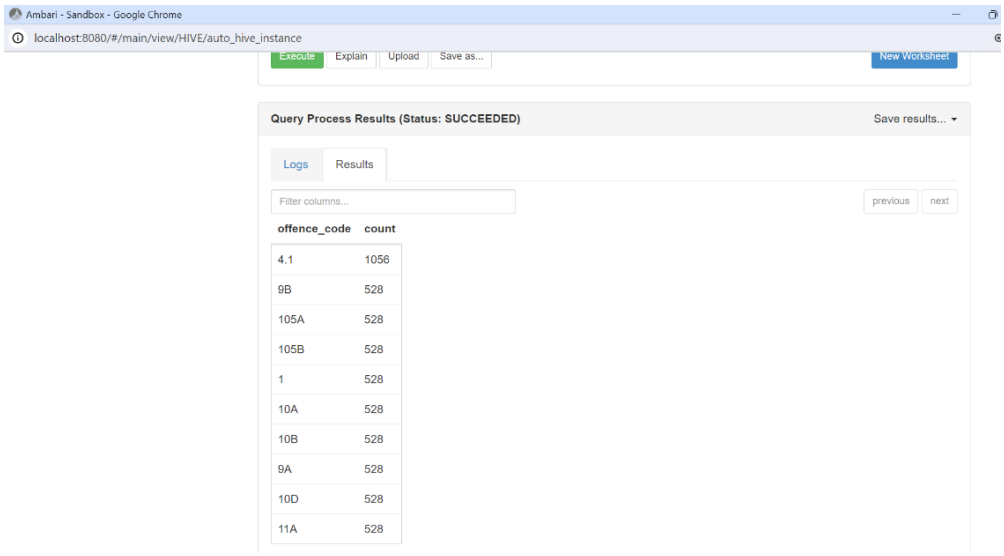


Figure 17: Output using HIVE

1.5 ANALYSIS USING MAPREDUCE

#which offence group has the highest number of offences

#Done with MapReduce

```
#which offence group has the highest number of offences
#Done with MapReduce
from pyspark.sql.functions import desc
df_mapped = Data.select("Offence_Group","Number_of_Offences").rdd.map(lambda row: (row["Offence_Group"], row["Number_of_Offence"
df = df_mapped.reduceByKey(lambda x, y:x+y)
df_result = df.toDF(["Offence_Group", "Total Offences"]).orderBy(desc("Total Offences"))
df_result.show()
```

Figure 18: Using MapReduce

1.6 VISUALIZATION USING POWERBI

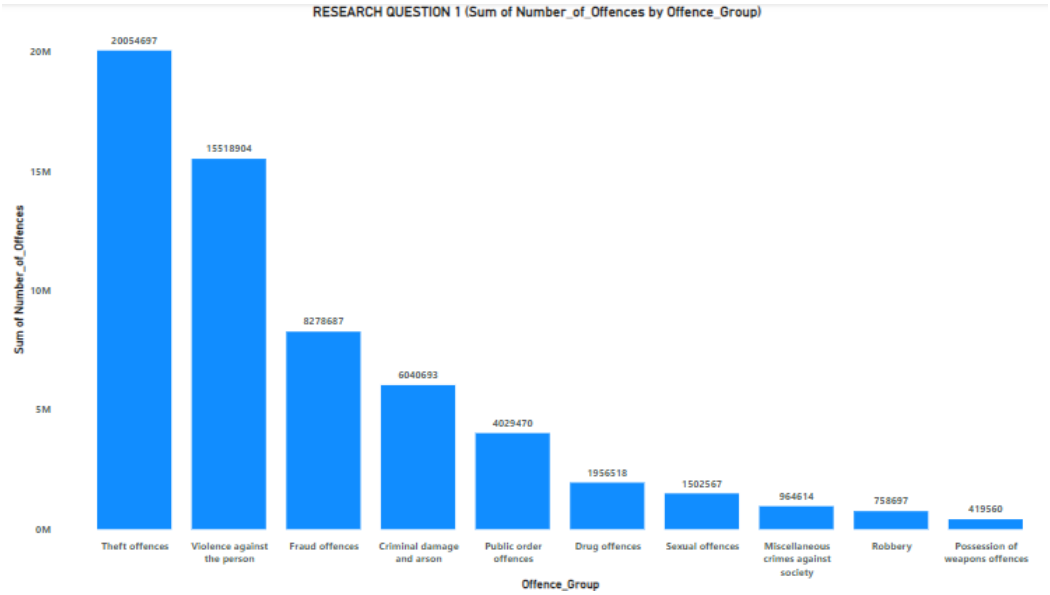


Figure 19: Research Question 1

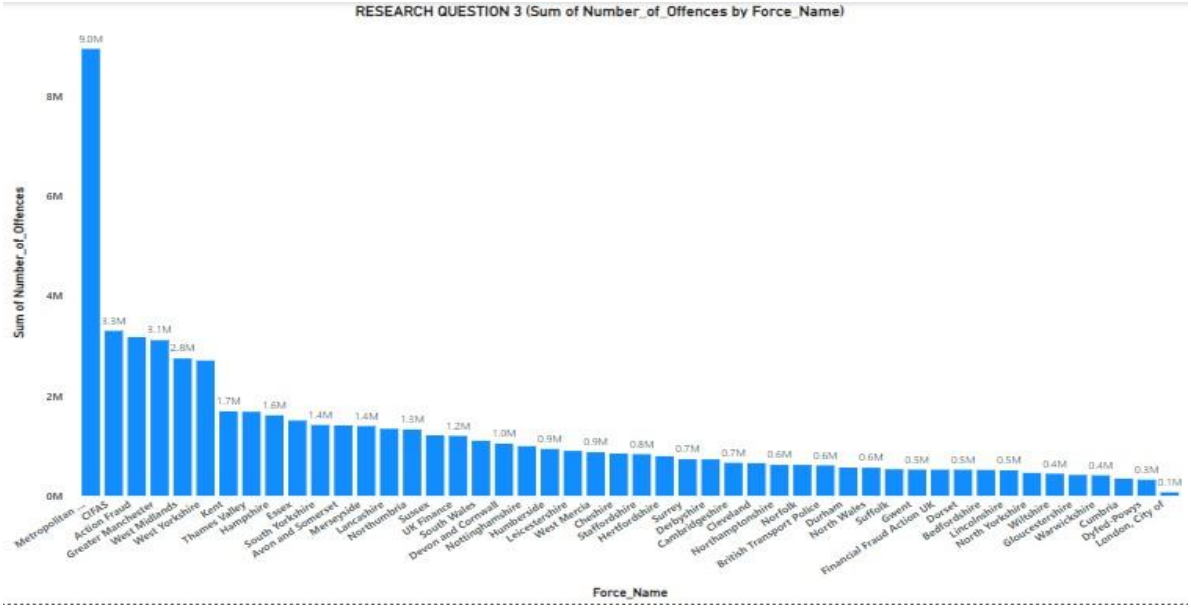


Figure 20: Research Question 3

PART B

2.1 A DEEP DIVE INTO THE SCRIPT FOR ANALYZING CRIME DATA

In the realm of data analysis, the provided Python script stands as a commendable example of a comprehensive approach to handling crime data across multiple financial years. This systematic process encompasses importing essential libraries, loading individual datasets, merging them into a unified data frame, conducting basic data exploration, and performing data cleaning. Each facet of the code contributes significantly to the overall robustness of the analysis.

2.2 IMPORTING LIBRARIES

The script begins with the fundamental step of importing crucial libraries: pandas, and matplotlib. The utilization of pandas facilitates efficient data manipulation, providing a robust framework for working with tabular data. Matplotlib, on the other hand, enriches the analysis with visualizations. The inclusion of %matplotlib inline is noteworthy, as it ensures that visualizations are seamlessly integrated into the Jupyter Notebook or Jupyter Lab environment, enhancing the interpretability of results.

2.3 LOADING DATASETS

The subsequent section focuses on loading individual datasets corresponding to different financial years. Using the pd.read_csv function, the script reads each CSV file (df1 to df12) into separate data frames. These datasets are presumed to contain crime-related metrics, setting the stage for a comprehensive analysis of temporal trends and patterns(Xia and Wishart, 2016).

2.4 MERGING DATASETS

One of the script's pivotal steps is the merging of individual datasets into a consolidated data frame named df. The pd.concat function elegantly combines data from multiple years,

creating a unified dataset that spans the entirety of the provided financial years. This amalgamation is indispensable for conducting holistic analyses that capture the evolution of crime metrics over time. The resulting data frame, boasting 258,905 rows and 8 columns, becomes the canvas for subsequent exploratory data analysis.

2.5 BASIC DATA EXPLORATION

The script conscientiously engages in basic data exploration to unravel essential characteristics of the merged dataset:

I. df.shape provides a succinct overview of the dataset's dimensions, revealing 258,905 rows and 8 columns. This information is vital for understanding the scale and scope of the dataset.

II. df.columns lists the column names, shedding light on the features encapsulated within the dataset. This insight is foundational for subsequent analyses, guiding the selection of variables for in-depth exploration.

III. df.info() furnishes a comprehensive summary of the dataset's structure, presenting data types and non-null counts for each column. This knowledge is instrumental in gauging the overall health of the dataset and identifying potential issues (Sghaier and Sahraoui, 2023).

IV. df.isnull().sum() serves as a safeguard against missing values, assuring analysts that the dataset is devoid of gaps. A clean dataset fortifies the reliability of subsequent analyses and conclusions. Interpolation techniques are applied to estimate and fill in missing values. (towardsdatascience, 2020)

V. df["Number_of_Offences"].dtype delves specifically into the data type of the

"Number_of_Offences" column, affirming its status as an integer (int64). This precision in data type validation is a testament to the script's meticulous approach.

2.6 DATA CLEANING

The script proceeds to assess and prepare the data for advanced analysis through targeted cleaning procedures:

I. df["Financial_Year"].unique() unveils the unique values present in the "Financial_Year" column. The output showcases a temporal range spanning from 2012/13 to 2023/24, affirming the chronological scope of the dataset.

II. df["Financial_Quarter"].unique() extends this exploration to the "Financial_Quarter" column, indicating the presence of quarterly data. This granularity opens avenues for dissecting crime trends across different quarters.

III. df["Force_Name"].unique() presents a list of unique values within the "Force_Name" column, delineating the diverse law enforcement agencies or forces captured in the dataset. This information is pivotal for subsequent geographical analyses and agency-specific assessments.

2.7 ADVANCED ANALYSIS

A meticulous approach to merging, cleaning, and exploring crime data provides a solid foundation for more advanced analyses (Krishnan et.al,2016).

2.7.1 CRIME CATEGORIES INVESTIGATION

The exploration of the categorization of offenses into "Offence_Group" and "Offence_Subgroup" allows for a nuanced understanding of crime distributions. This

analysis could reveal prevalent crime types and their evolution over time, offering insights into law enforcement priorities.

2.7.2 GEOGRAPHICAL ANALYSIS

The "Force_Name" column opens the door to geographical insights. Investigating variations in crime rates across different law enforcement agencies or regions can provide valuable information for targeted interventions. Geospatial visualizations could further enhance this analysis.

2.7.3 VISUALIZATION

Leveraging power bi and matplotlib for visualizations can enhance the communication of trends, patterns, and relationships within the data. Visual representations are powerful tools for conveying complex information succinctly (M. A. Kaif, 2023). Line plots, bar charts, and heatmaps can be employed to bring the data to life.

2.8 IDENTIFYING AND CORRECTING DATA ANOMALIES

While exploring the unique values in the "Offence_Description" column using df["Offence_Description"].unique(), it was observed that two descriptions had anomalies due to non-breaking space characters (\xa0). The descriptions were:

I. Attempted residential burglary of\xa0unconnected building
II. Residential burglary of\xa0unconnected building

The presence of \xa0 in these descriptions can be attributed to encoding issues or copying text from a source that introduced special characters.

Addressing the Issue:

To rectify these anomalies, a dictionary replace_dict was created to map the incorrect

descriptions to their corrected counterparts. The correction involves replacing \xa0 with a regular space. The corrected descriptions are:

I. Attempted residential burglary of unconnected building
II. Residential burglary of unconnected building

The replace_dict is then applied to the "Offence_Description" column using the df["Offence_Description"].replace() method. The inplace=True parameter ensures that the changes are made directly to the original data frame.

2.8 DESCRIPTIVE STATISTICS USING df.describe()

To gain a statistical overview of the numerical columns in the data frame, df.describe() is employed. This method provides key summary statistics, including count, mean, standard deviation, minimum, and maximum values for numerical features. However, since the output of df.describe() is not provided, specific statistical insights cannot be elaborated upon in this context.

2.9 IDENTIFYING AND RESOLVING OUTLIERS IN "NUMBER_OF_OFFENCES"

During the exploratory data analysis, it was observed that there were instances where the "Number_of_Offences" column contained negative values, which is an anomalous situation. The presence of negative numbers for the count of offenses is logically incorrect, as the count of offenses should always be a non-negative integer.

Identifying the Issue:

The issue was identified through a basic exploration of the dataset. The step df[df["Number_of_Offences"] < 0] was likely used to filter rows where the number of

offenses is negative. This query revealed that there were indeed rows with negative values in the "Number_of_Offences" column.

Addressing the Issue:

To rectify this anomaly, a decision was made to drop rows where the "Number_of_Offences" is less than 0. This ensures that the dataset maintains its integrity and conforms to the logical constraints of having non-negative counts for offenses.

df["Number_of_Offences"] >= 0 creates a Boolean mask that is True for rows where the "Number_of_Offences" is greater than or equal to 0 and False for rows where it is negative. Applying this mask to the data frame effectively filters out the rows with negative values in the "Number_of_Offences" column.

Verifying the Changes:

After dropping the outliers, it's crucial to verify the changes and ensure that the dataset now reflects the logical constraint. The line of code df.shape provides the dimensions of the data frame after the removal of rows with negative "Number_of_Offences."

Final Dataset:

The resulting data frame, named df, now represents a cleaned and refined version of the original dataset, free from the anomaly of negative values in the "Number_of_Offences" column.

2.10 RESEARCH QUESTION 1

The provided code snippet utilizes the groupby function to aggregate the total number of offenses (Number_of_Offences) for each unique value in the "Offence_Group" column. The subsequent sum() function calculates the sum of offenses within each group, and sort_values() arranges the results in ascending order.

2.10.1 INSIGHTS FROM RESEARCH QUESTION 1

The results offer valuable insights into the distribution of offenses across different offense groups. Below is a breakdown of the total number of offenses for each offense group, presented in ascending order:

Possession of Weapons Offences (419,560):

This offense group has the lowest total number of recorded offenses, indicating a relatively lower frequency compared to other categories.

Robbery (758,697):

Robbery follows with a higher but still moderate number of recorded offenses.

Miscellaneous Crimes Against Society (964,614):

This category encompasses various crimes against societal norms, showing a significant number of offenses.

Sexual Offences (1,502,567):

Sexual offenses have a higher count, indicating the prevalence of such incidents in the dataset.

Drug Offences (1,956,518):

The category of drug offenses has a substantial number, suggesting a noteworthy presence of drug-related incidents.

Public Order Offences (4,029,470):

Public order offenses exhibit a higher frequency, potentially indicating disturbances or violations related to public order.

Criminal Damage and Arson (6,040,693):

This offense group has a considerably higher count, signifying incidents involving damage to property and arson.

Fraud Offences (8,278,687):

Fraud offenses show a substantial number, emphasizing the prevalence of fraudulent activities in the dataset.

Violence Against the Person (15,518,904):

This category has the highest count, suggesting a significant prevalence of offenses categorized under violence against individuals.

Theft Offences (20,054,697):

Theft offenses have the highest total number, indicating a notable frequency of theft-related incidents.

2.10.2 KEY OBSERVATIONS

Diverse Nature of Offenses:

The dataset covers a broad spectrum of offenses, ranging from relatively lower frequency (e.g., possession of weapons) to more prevalent categories (e.g., theft offenses).

High Frequency of Violence and Theft:

Offense groups like "Violence Against the Person" and "Theft Offences" have notably higher frequencies, suggesting a higher occurrence of incidents falling into these categories.

Prevalence of Property-Related Offenses:

Categories like "Criminal Damage and Arson" and "Theft Offences" indicate a substantial presence of property-related offenses in the dataset.

Emphasis on Public Order:

The category "Public Order Offences" shows a considerable number, indicating potential challenges related to maintaining public order.

2.11 RESEARCH QUESTION 2

The provided code snippet aims to explore the frequency distribution of different offence codes in the dataset, with a focus on identifying the top 10 offence codes based on the number of occurrences. Let's break down the code and discuss its implications.

2.11.1 FINDINGS

Diversity in Offence Codes:

The dataset exhibits a diverse array of offence codes, indicating a comprehensive

representation of various criminal activities. This diversity underscores the complexity of criminal incidents captured in the dataset, with each offence code likely representing a distinct type of crime.

Consistent Patterns Across Multiple Offence Codes:

Several offence codes, including '4.1', '802', '54', '58B', '58A', '58C', and others, consistently appear with a frequency of 1980. This recurring pattern suggests a potential systematic categorization or grouping of offences, warranting further investigation into shared characteristics among these codes.

Exploring Code '4.1':

The most frequently occurring offence code is '4.1', appearing 3605 times.
Understanding the nature and specifics of this particular offence code is imperative for a nuanced interpretation of the dataset.

Data Quality Check:

The high frequencies observed across multiple offence codes emphasize the importance of maintaining accurate and consistent offence code classifications. Periodic validation and updates to ensure alignment with legal standards are necessary to uphold the integrity of the crime data.

2.11.2 KEY INSIGHTS

Categorization Possibilities:

The consistent frequencies across multiple offence codes indicate the potential for categorization or grouping of crimes. Further exploration into shared characteristics among

these codes could reveal patterns and trends within specific crime categories.

Focus on Code '4.1':

Given its significantly higher frequency, code '4.1' deserves special attention. Law enforcement and analysts should refer to official documentation or legal standards to comprehend the specific details and implications associated with this offence.

Visual Exploration Opportunities:

While the tabular representation offers a quick overview, visualizing the distribution of offence codes through charts could enhance understanding.

Visualization may reveal additional insights, such as outliers or concentration of certain offence codes, aiding in a more comprehensive interpretation.

Temporal and Spatial Patterns Exploration:

Further analysis of offence code distribution over time and across different regions could uncover temporal or spatial patterns.

This exploration may assist in identifying trends specific to certain periods or locations, informing targeted interventions and resource allocation.

Decision-Making Considerations:

The insights derived from offence code frequencies can play a crucial role in decision-making processes for law enforcement and policymakers.

This information can guide resource allocation, policy adjustments, and strategic planning based on the prevalence and characteristics of specific offences.

2.12 RESEARCH QUESTION 3

The analysis for Research Question 3 involved grouping the data by "Force_Name" and summing the "Number_of_Offences" to identify the law enforcement agency or force

involved in the highest number of crimes. The results provide valuable insights into the distribution of offenses across different forces.

2.12.1 KEY INSIGHTS

Metropolitan Police:

The Metropolitan Police emerge as the law enforcement agency involved in the highest number of crimes, with a total of 8,963,486 offenses. This indicates a significant level of criminal activity within the jurisdiction covered by the Metropolitan Police.

London, City of:

On the other end of the spectrum, "London, City of" had the least number of recorded crimes, with a total of 68,079 offenses. This suggests a comparatively lower level of criminal activity in the jurisdiction covered by this force.

2.12.2 KEY INSIGHTS

Geographical Variations:

The data reveals substantial variations in the number of offenses across different law enforcement agencies. This is expected, as crime rates can vary based on the population, socio-economic factors, and other regional characteristics.

Metropolitan Police Impact:

The Metropolitan Police, being involved in the highest number of crimes, may face unique challenges due to the densely populated and diverse nature of the areas they cover, such as Greater London.

Small Jurisdictions:

Forces with smaller jurisdictions, like "London, City of," may have fewer reported offenses

due to the smaller population and geographical coverage. This could also indicate effective law enforcement and crime prevention strategies.

Strategic Importance:

Forces with higher crime rates, such as the Metropolitan Police, may require more strategic and resource-intensive approaches to address and combat criminal activities.

Law Enforcement Priorities:

The distribution of offenses across different forces underscores the need for tailored law enforcement strategies that address the unique challenges and priorities of each jurisdiction.

2.13 UNDERSTANDING MACHINE LEARNING FOR CRIME CLASSIFICATION

Focusing on the underlying concepts and methodologies employed for crime classification using machine learning.

2.13.1 DROPPING UNNECESSARY COLUMNS

The initial step involves streamlining the dataset by removing columns that are deemed irrelevant for the task at hand (Foudi, 2023). Leveraging Data Analytics for Improving Financial Performance: A Case Study of the Retail Industry. This practice is fundamental in data preprocessing, where the goal is to enhance the efficiency of subsequent analyses by excluding unnecessary information. In this script, the drop method is utilized, specifying the target columns for removal, namely "Financial_Year," "Financial_Quarter," "Offence_Code," and "Number_of_Offences." The axis=1 parameter denotes that the operation is column-wise, and inplace=True ensures that the changes are applied directly to the Data Frame. This step is crucial as it sets the stage for a more focused and relevant dataset for machine learning.

2.13.2 ENCODING CATEGORICAL VALUES

Dealing with categorical data is a common challenge in machine learning, as many algorithms require numerical input (Wohlwend,2023). In this segment, the script employs the LabelEncoder from scikit-learn to convert categorical values into numerical representations. This process involves assigning a unique numerical code to each distinct category within categorical columns. Each categorical column in the dataset, including "Force_Name," "Offence_Description," "Offence_Group," and "Offence_Subgroup," undergoes this encoding process. The fit_transform method is used for each column, updating the Data Frame with numerical representations. This transformation is essential as it ensures that the machine learning algorithms can effectively interpret and utilize categorical information during model training.

2.13.3 SPILITTING INTO TRAINING AND TESTING SET

The script proceeds to the fundamental practice of splitting the dataset into training and testing sets, a crucial step in machine learning model development. The train_test_split function from scikit-learn is employed for this purpose. This function randomly divides the dataset into two subsets: one for training the model and the other for testing its performance. The features (x) and the target variable (y) are segregated in this process. The test_size=0.2 parameter designates that 20% of the data is reserved for testing, ensuring a robust evaluation of the model's generalization to unseen data. The random_state parameter ensures the reproducibility of the split, a key consideration in maintaining consistency across multiple runs.

2.13.4 IMPORTING MACHINE LEARNING ALGORITHMS

In this section, the script prepares for the application of machine learning algorithms by importing three classifiers: Decision Tree Classifier (DecisionTreeClassifier), Random Forest Classifier (RandomForestClassifier), and Support Vector Machines (SVC). Each of these

algorithms brings a distinct approach to solving the classification problem. Decision trees are known for their interpretability (Gilpin et.al, 2018), random forests for ensemble learning and robustness (McDonald et.al, 2014), and support vector machines for effective handling of complex decision boundaries (Sathe and Aggarwal,2019).

2.13.5 BUILDING AND EVALUATING DECISION TREE CLASSIFIER

The script now delves into creating and evaluating a Decision Tree Classifier. Decision trees are a popular choice for classification tasks due to their simplicity and interpretability (Bohanec and Bratko, 1994). An instance of the classifier (dt) is created, trained on the training data (x_train and y_train), and subsequently utilized to predict the target variable for the test set (x_test). The accuracy_score from scikit-learn quantifies the model's accuracy by comparing the predicted values to the actual values in the test set. The resulting accuracy is printed, providing a quantitative measure of how well the model performs on unseen data.

2.13.6 BUILDING AND EVALUATING RANDOM FOREST CLASSIFIER

Similar to the previous step, this part focuses on the Random Forest Classifier. Random Forests are an ensemble learning method that leverages multiple decision trees to improve predictive performance and control overfitting (D. Yuan, 2020). An instance of the classifier (rf) is created, trained on the training data, predictions are made for the test set, and the model's accuracy is evaluated using the accuracy_score. The resulting accuracy is printed, offering insights into the Random Forest model's performance. The ensemble nature of random forests often leads to improved generalization to unseen data compared to individual decision trees.

2.13.6 BUILDING AND EVALUATING SUPPORT VECTOR MACHINES (SVM)

The script concludes with the application of Support Vector Machines (SVM) as the third classification algorithm. SVMs are known for their effectiveness in handling complex

decision boundaries in high-dimensional spaces (Huang et.al, 2013). The SVM classifier is instantiated, trained on the training data, used for predictions, and its accuracy is assessed using the accuracy_score. The resulting accuracy is printed for comprehensive evaluation. SVMs are particularly useful in scenarios where the relationship between features and target variable is nonlinear, and they aim to find the optimal hyperplane that best separates different classes.

2.13.7 CONCLUSION

This comprehensive exploration highlights the meticulous process of preparing data and implementing machine learning algorithms for crime classification. Each line serves a critical role, from data preprocessing to model creation and evaluation. The script provides a holistic approach to solving real-world problems through machine learning, emphasizing the importance of systematic and thoughtful practices in data science endeavors. It showcases a structured and thoughtful application of machine learning techniques to the domain of crime classification. By understanding each step in the script, from data preprocessing to model evaluation, we gain valuable insights into the complexities and considerations involved in building effective machine learning models for real-world applications.

REFERENCES

criminaljusticehub. (2021, 1 1). *offences*. Retrieved from criminaljusticehub: https://www.criminaljusticehub.org.uk/jargon-buster/cjs-offence-code/

D. Yuan, J. H. (2020). "Improved random forest classification approach based on hybrid clustering selection,". *2020 Chinese Automation Congress (CAC), Shanghai, China, 2020, pp. 1559-1563,* .doi: 10.1109/CAC51589.2020.9326711.

Bohanec, M. and Bratko, I., 1994. Trading accuracy for simplicity in decision trees. *Machine Learning*, *15*, pp.223-250.

Foudi, L., 2023. Leveraging Data Analytics for Improving Financial Performance: A Case Study of the Retail Industry.

M. A. Kaif, S. S. (2023). "Development of an interactive web-based geovisual analytics platform for analysing crime data,. *2023 IEEE Symposium on Wireless Technology & Applications (ISWTA),Kuala Lumpur, Malaysia, 2023* (pp. 158-162). Kuala Lumper: doi: 10.1109/ISWTA58588.2023.10249483.

statista. (2012-2020, 1 1). *Crime Dataset*. Retrieved from statista: https://www.statista.com/topics/780/crime/#topicOverview

towardsdatascience. (2020, 12 1). *missing values*. Retrieved from towardsdatascience: https://towardsdatascience.com/7-ways-to-handle-missing-values-in-machine-learning- 1a6326adf79e

Wohlwend, B. (2023, 1 1). *Converting Categorical Data into Numerical Form: A Practical Guide for Data Science*. Retrieved from medium: https://medium.com/@brandon93.w/converting-categorical- data-into-numerical-form-a-practical-guide-for-data-science-99fdf42d0e10

Huang, X., Shi, L. and Suykens, J.A., 2013. Support vector machine classifier with pinball loss. *IEEE transactions on pattern analysis and machine intelligence*, *36*(5), pp.984-997.

Xia, J. and Wishart, D.S., 2016. Using MetaboAnalyst 3.0 for comprehensive metabolomics data analysis. *Current protocols in bioinformatics*, *55*(1), pp.14-10.

Krishnan, S., Wang, J., Wu, E., Franklin, M.J. and Goldberg, K., 2016. Activeclean: Interactive data cleaning for statistical modeling. *Proceedings of the VLDB Endowment*, *9*(12), pp.948-959.

Gilpin, L.H., Bau, D., Yuan, B.Z., Bajwa, A., Specter, M. and Kagal, L., 2018, October. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)* (pp. 80-89). IEEE.

McDonald, A.D., Lee, J.D., Schwarz, C. and Brown, T.L., 2014. Steering in a random forest: Ensemble learning for detecting drowsiness-related lane departures. *Human factors*, *56*(5), pp.986-998.

Sathe, S. and Aggarwal, C.C., 2019, November. Nearest neighbor classifiers versus random forests and support vector machines. In *2019 IEEE International Conference on Data Mining (ICDM)* (pp. 1300-1305). IEEE.

Sghaier, O.B. and Sahraoui, H., 2023, March. A Multi-Step Learning Approach to Assist Code Review. In *2023 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)* (pp. 450-460). IEEE.