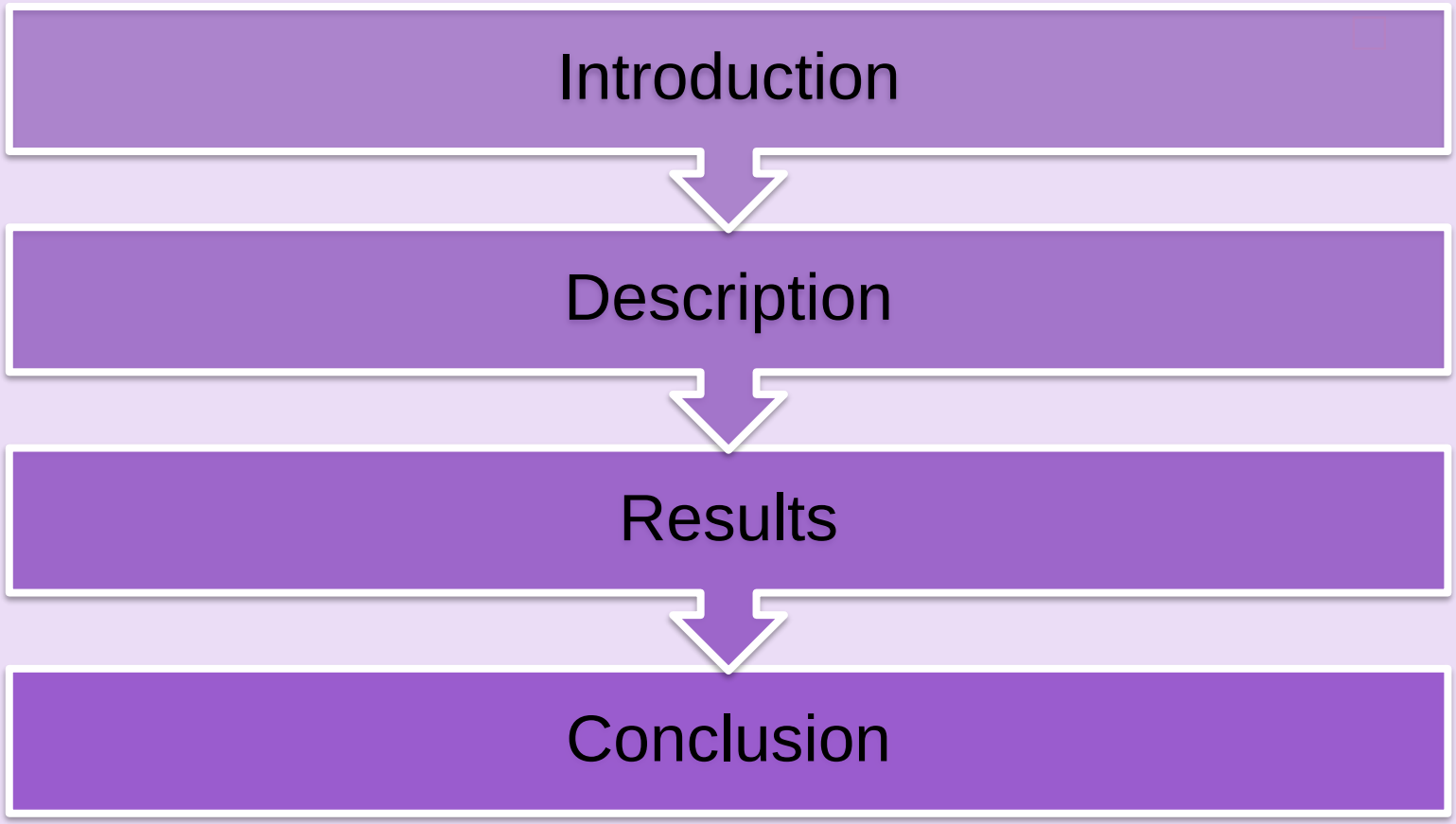
Chicago's crime data in 2021

Dr. Moustafa El-Attar

Agenda



Introduction

In practically every nation, crime is rising quickly. Analyzing many aspects of crime and identifying criminal trends are highly important. Security agencies throughout the world are working hard to lower these crimes, but the volume of crime data is growing quickly, making it challenging to handle such a large amount of data and to maintain records of crimes that occur over a vast geographic area and over a variety of time periods. Therefore, having a criminal information system that can analyze a lot of data quickly is essential. Also, one of the most efficient techniques to explore, analyze, find patterns, and forecast future crimes in a vast quantity of data is data analysis by using clustering, classification, and association mining.

Description

Dataset for year 2021

.csv format

48.1 MB → size

Collected from : Kaggle

Libraries used

- Data Wrangling → Panadas
- NumPy (for math operations)
- Datetime (python datetime module)
- Data Visualization
  →
  Matplotlib & seaborn

2.1 Data Gathering

```
# put dataset --> dataframe
df = pd.read_csv('Chicago_Crimes_2021.csv')
df.head(5)
```

	ID	Case Number	Date	Block	IUCR	Primary Type	Description	Location Description	Arrest	Domestic	...	Ward	Community Area	FBI Code	X Coordinate	Y Coordinate	Year	
0	12445124	JE328077	08/05/2021 10:00:00 PM	033XX S LEAVITT ST	0610	BURGLARY	FORCIBLE ENTRY	RESIDENCE - GARAGE	0	0	...	25.0	59.0	05	1162228.0	1882544.0	2021	0
1	12541099	JE445417	11/14/2021 10:58:00 PM	003XX N LAVERGNE AVE	143A	WEAPONS VIOLATION	UNLAWFUL POSSESSION - HANDGUN	STREET	1	0	...	28.0	25.0	15	1142984.0	1901704.0	2021	1
2	12344992	JE204888	04/16/2021 12:00:00 AM	011XX W 50TH ST	0820	THEFT	$500 AND UNDER	APARTMENT	0	0	...	20.0	61.0	06	1169689.0	1871646.0	2021	0
3	12345838	JE205622	03/03/2021 06:20:00 PM	012XX N WELLS ST	0820	THEFT	$500 AND UNDER	STREET	0	0	...	2.0	8.0	06	1174488.0	1908479.0	2021	0
4	12444222	JE326942	08/05/2021 11:00:00 AM	044XX N BROADWAY	0486	BATTERY	DOMESTIC BATTERY SIMPLE	APARTMENT	0	1	...	46.0	3.0	08B	1168547.0	1929701.0	2021	0

5 rows × 22 columns

Summary

21.csv

```
#Summary
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 207914 entries, 0 to 207913
Data columns (total 22 columns):
 #   Column                Non-Null Count   Dtype  
---  ------                --------------   -----  
 0   ID                    207914 non-null  int64  
 1   Case Number           207914 non-null  object 
 2   Date                  207914 non-null  object 
 3   Block                 207914 non-null  object 
 4   IUCR                  207914 non-null  object 
 5   Primary Type          207914 non-null  object 
 6   Description           207914 non-null  object 
 7   Location Description  207071 non-null  object 
 8   Arrest                207914 non-null  int64  
 9   Domestic              207914 non-null  int64  
 10  Beat                  207914 non-null  int64  
 11  District              207914 non-null  float64
 12  Ward                  207903 non-null  float64
 13  Community Area        207914 non-null  float64
 14  FBI Code              207914 non-null  object 
 15  X Coordinate          202902 non-null  float64
 16  Y Coordinate          202902 non-null  float64
 17  Year                  207914 non-null  int64  
 18  Updated On            207914 non-null  object 
 19  Latitude              202902 non-null  float64
 20  Longitude             202902 non-null  float64
 21  Location              202902 non-null  object 
dtypes: float64(7), int64(5), object(10)
memory usage: 34.9+ MB
```

84.74 GB available

Missing Values

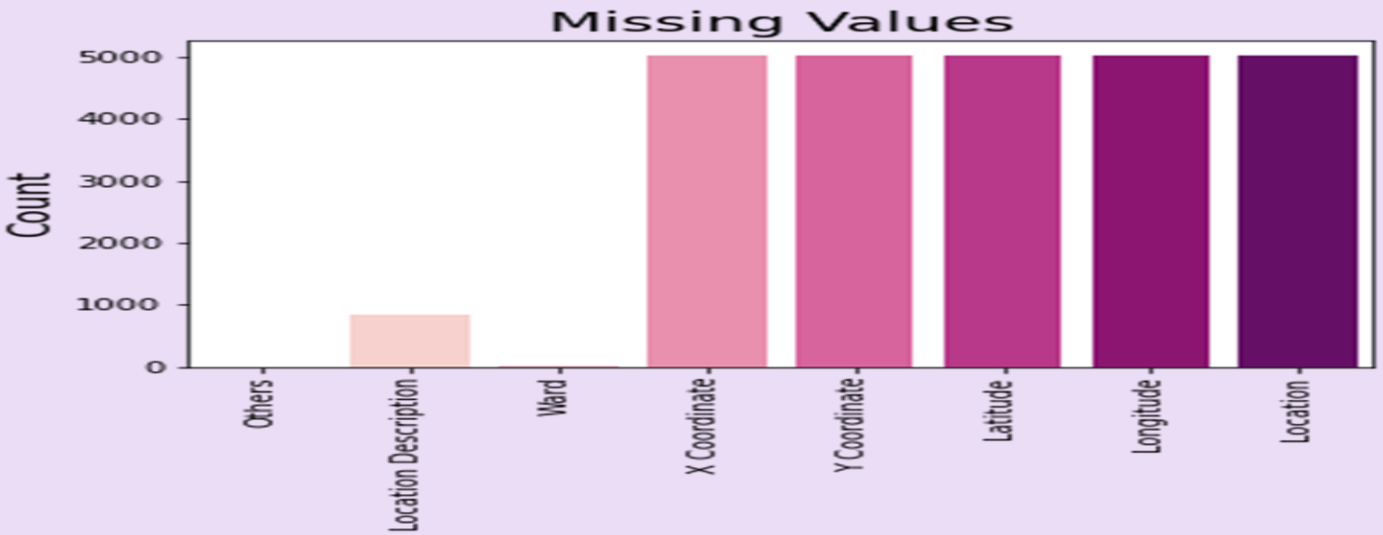
[9]

The number of missing values --> per feature is :

	Others	Location Description	Ward	X Coordinate	Y Coordinate	Latitude	Longitude	Location
Missing Values	0	843	11	5012	5012	5012	5012	5012

21.csv

```
# Plotting the missing values in the dataset
```



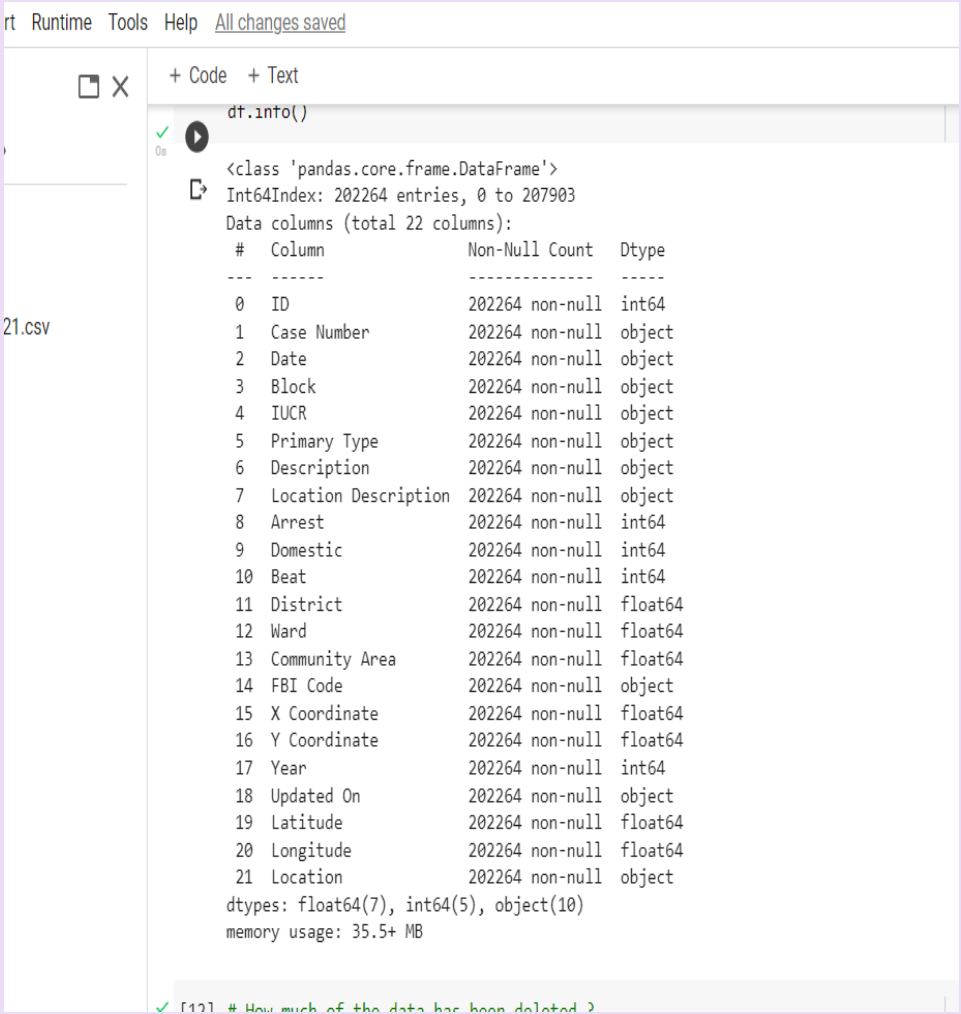
Missing Values Cont.

The simplest cleaning technique here

would be to drop all the rows with at least

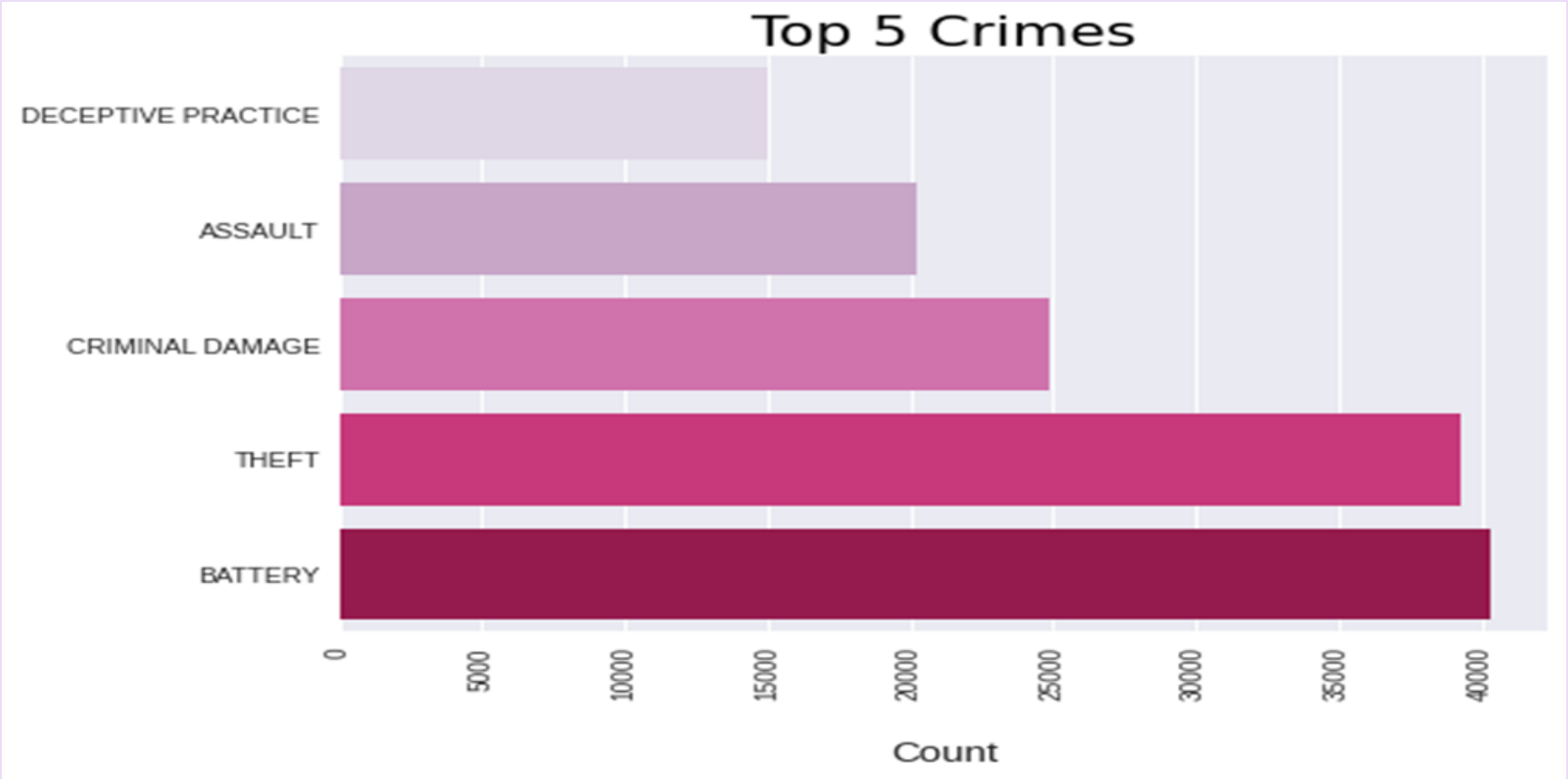
one missing value --> as every feature in

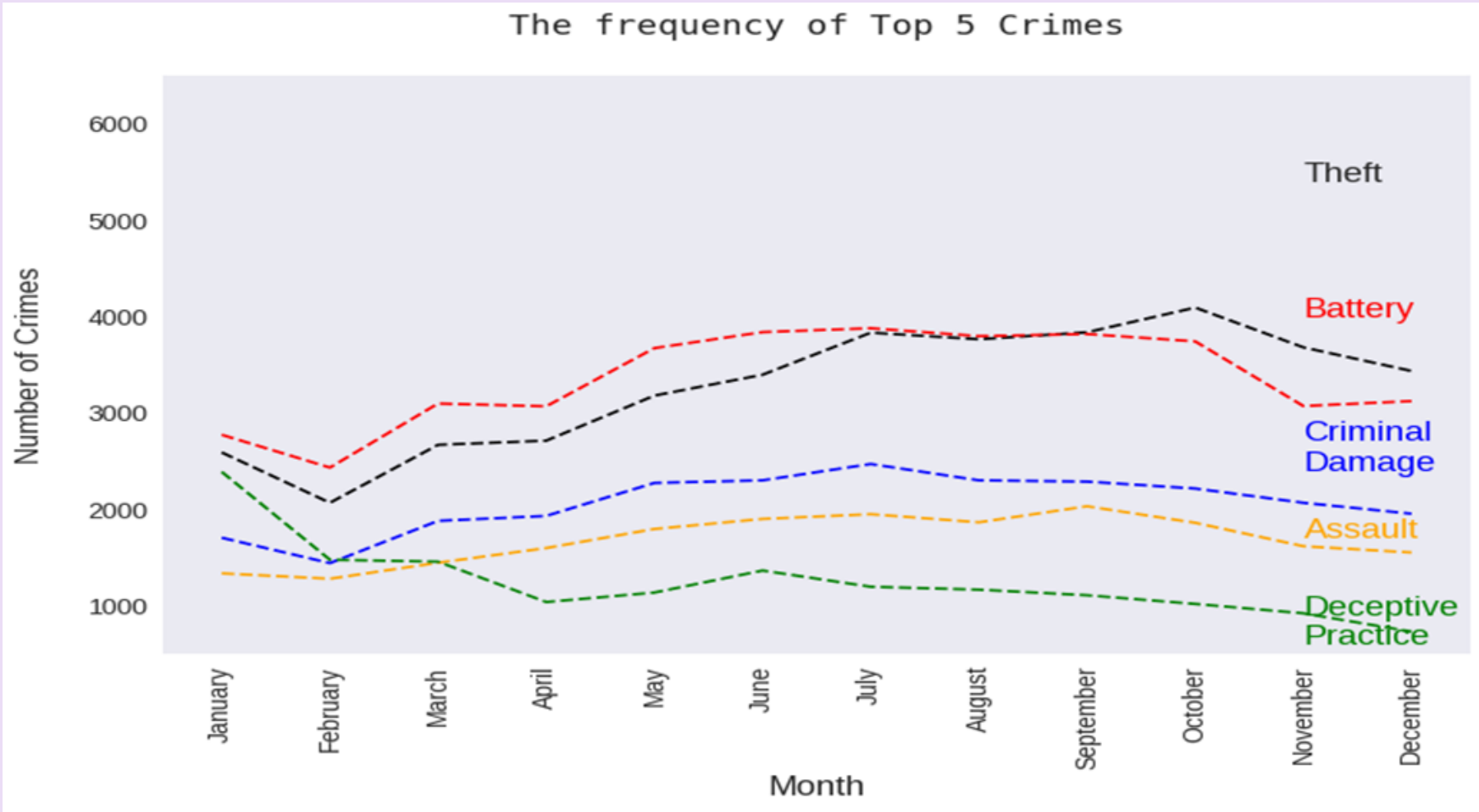
the dataset is important

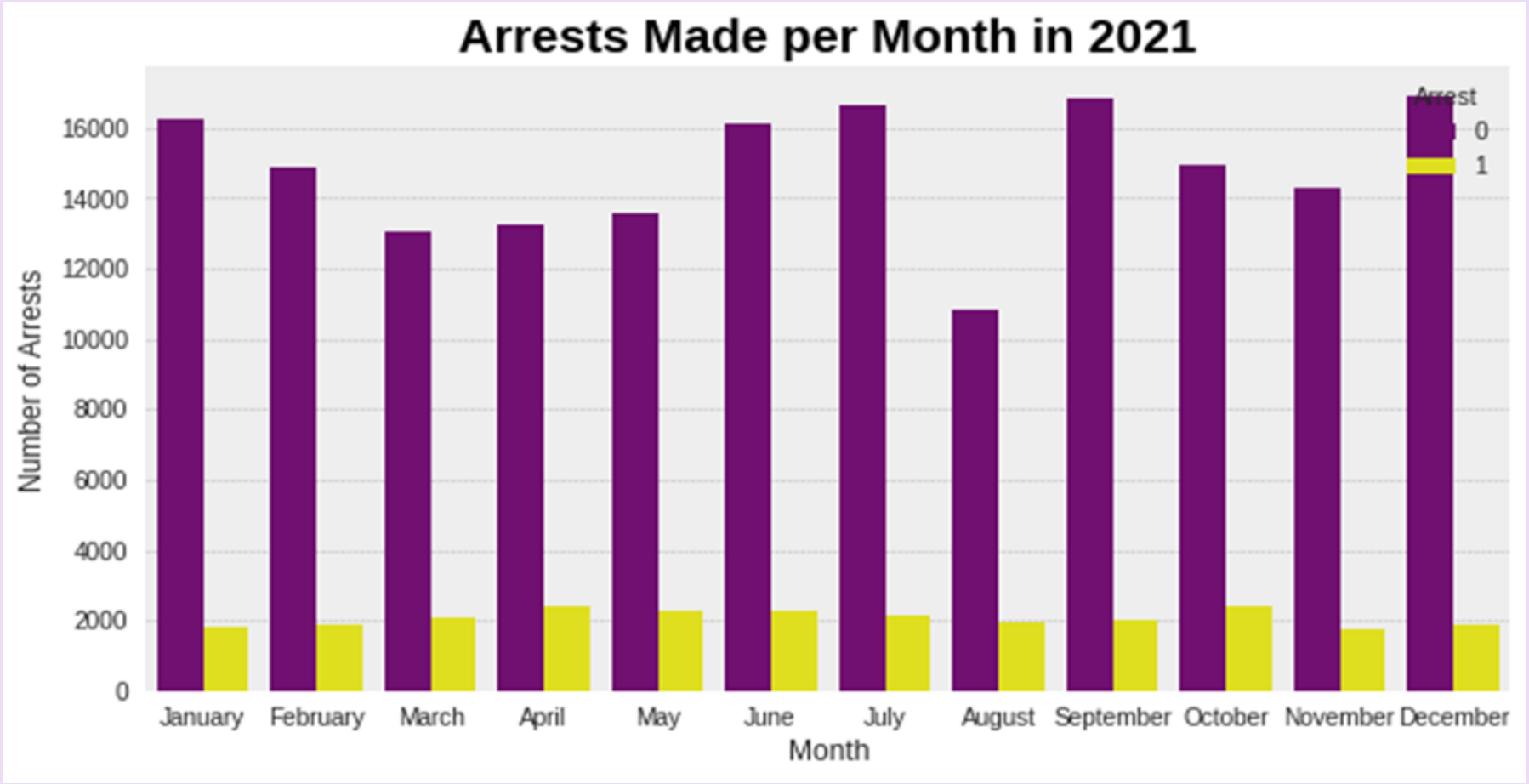


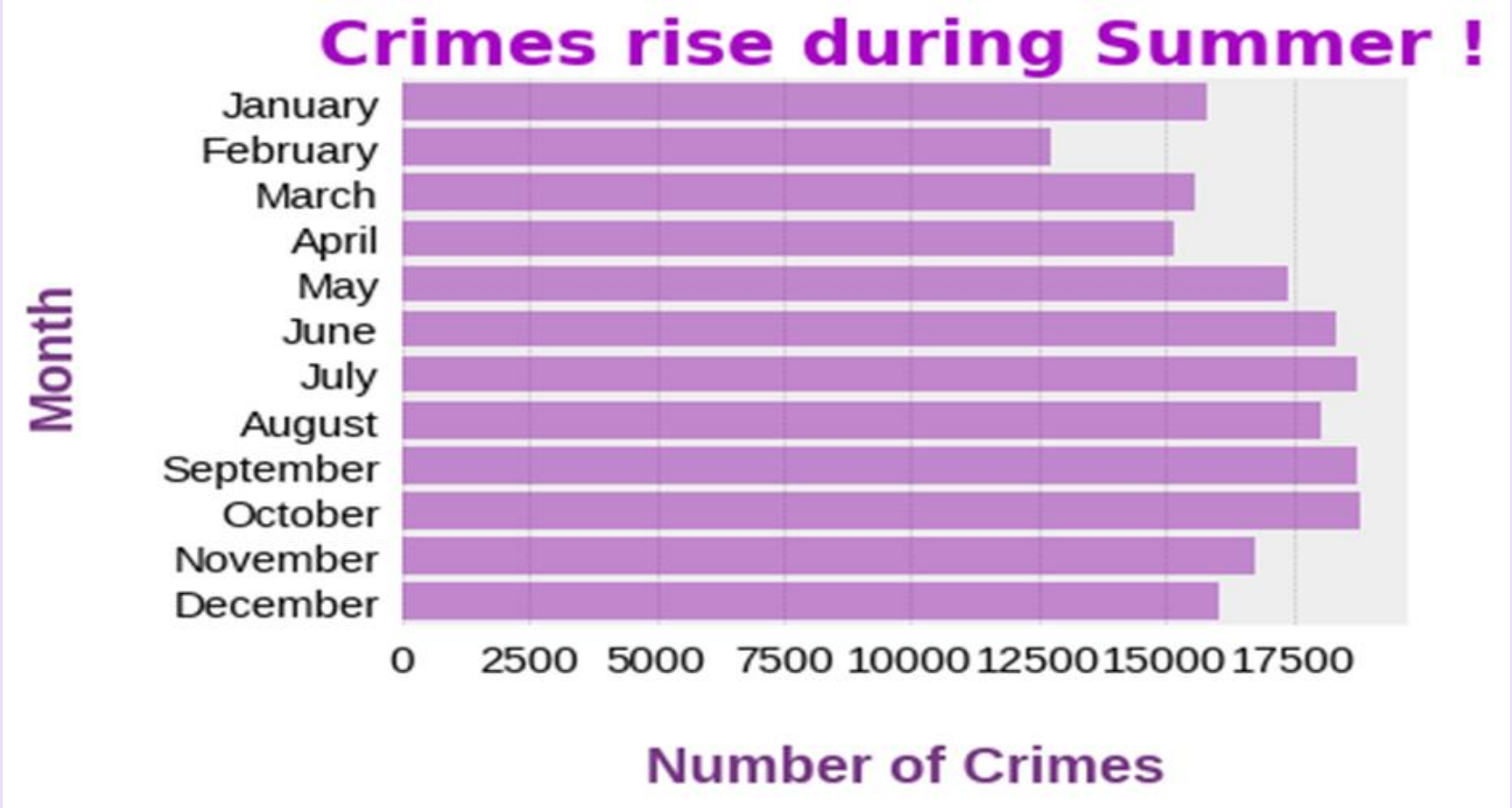
Observations

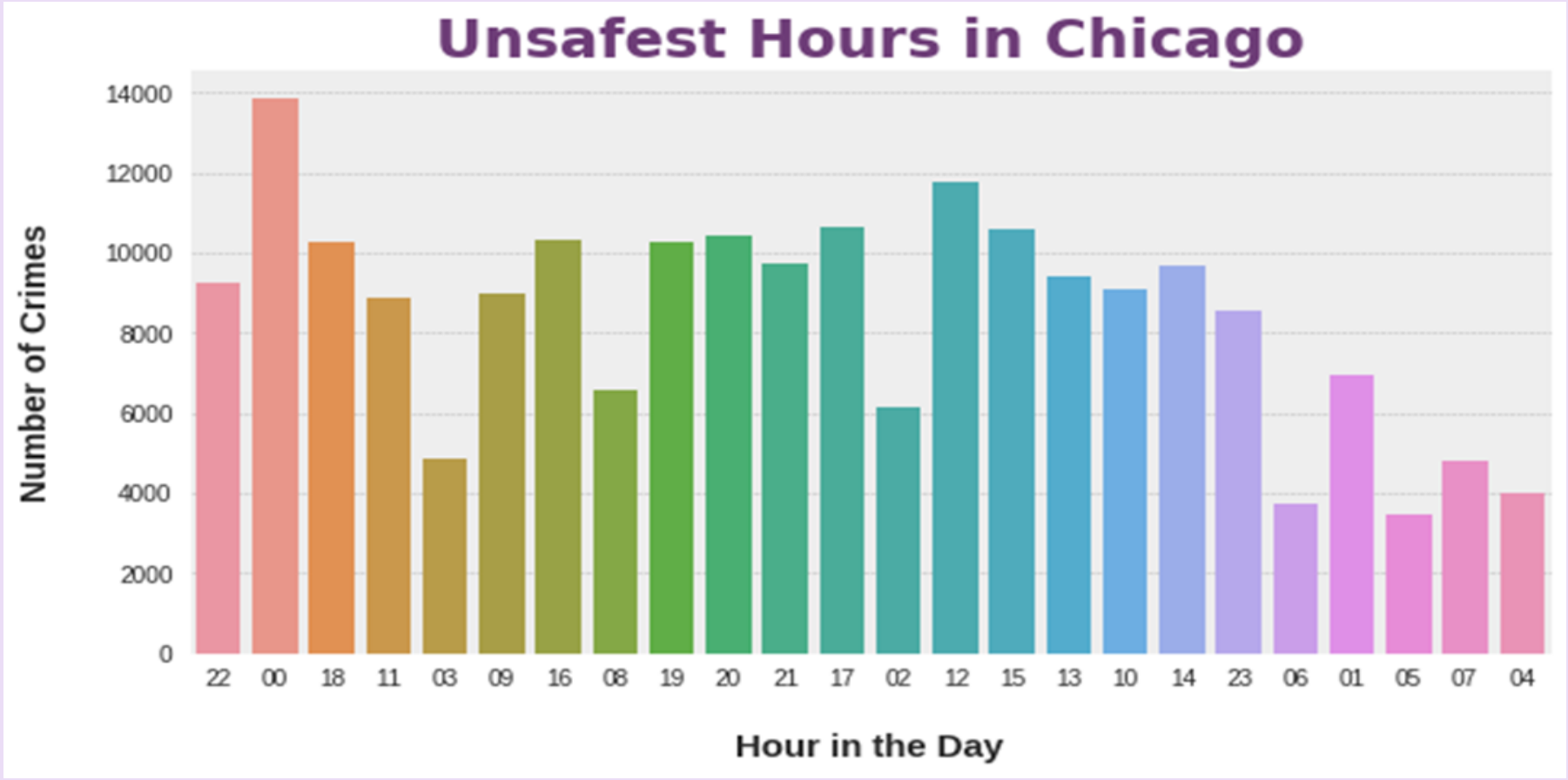
- Most occurring top 5 crimes in Chicago in 2021
- Modify the date column to a pythonic format
- Convert the date-time column to a known format
- Top crimes in Chicago in the year 2021
- Create a new column month
- The probability of arrest of Chicago
- Distribution of arrests across the months
- What are the most unsafe hours
- Is your house safe from burglary during the day
- Visualize a crime pattern for 24 hours
- Crime & time
- Crime & location
- Most common Occurrences per district
- Which region is given crime concentrated in?
- Visualizing Narcotics (to get an understanding of the most prevalent of drugs in the area
- Visualizing with heatmaps
- Using Supervised machine learning to predict crime hotspots


Top 5 Crimes
DECEPTIVE PRACTICE
ASSAULT
CRIMINAL DAMAGE
THEFT
BATTERY
0
5000
10000
15000
20000
25000
30000
35000
40000
Count

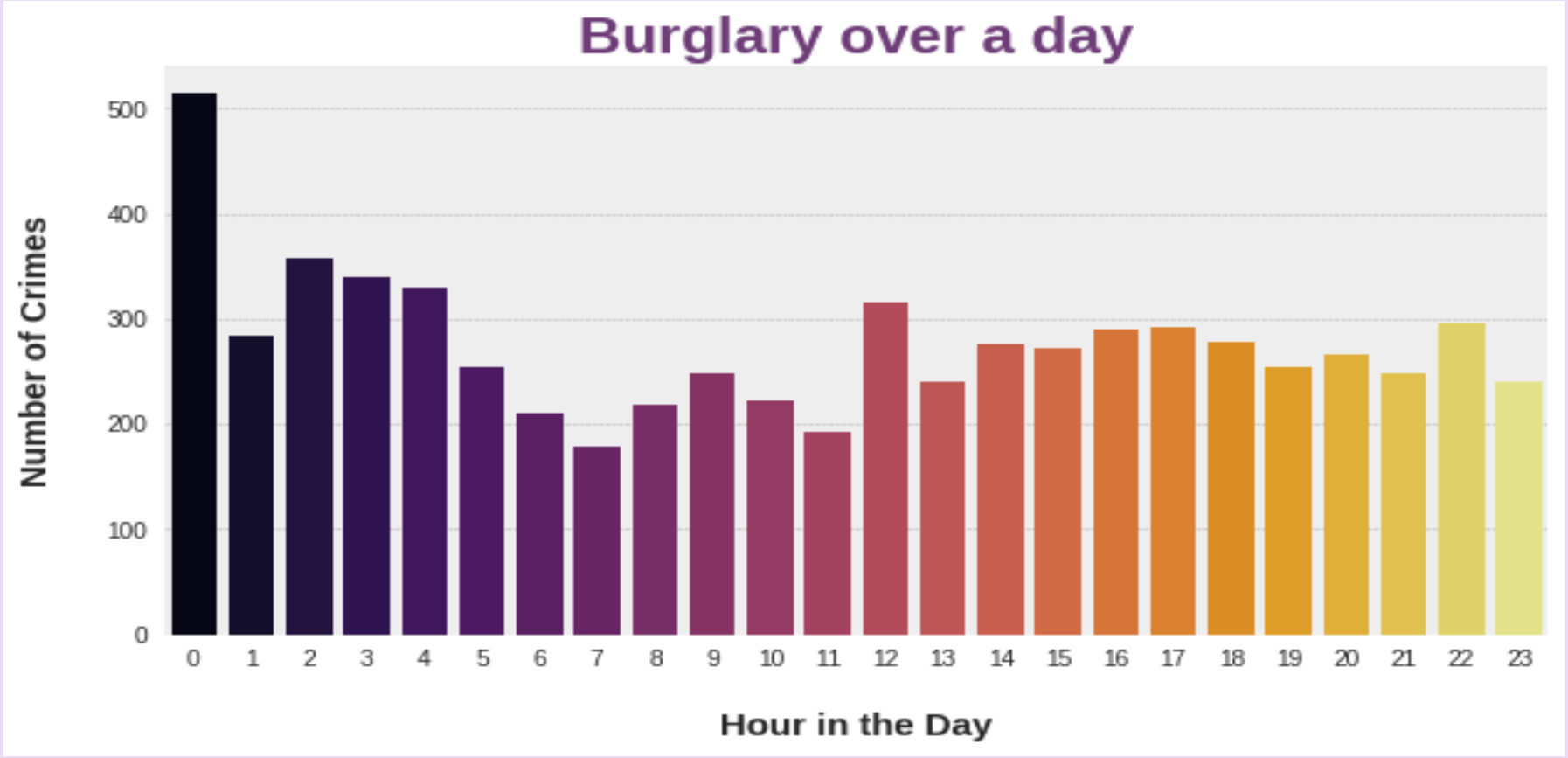

The frequency of Top 5 Crimes
Number of Crimes
6000
5000
4000
3000
2000
1000
Theft
Battery
Criminal Damage
Assault
Deceptive Practice
January
February
March
April
May
June
July
August
September
October
November
December
Month


Arrests Made per Month in 2021
Arrest
0
1
Number of Arrests
16000
14000
12000
10000
8000
6000
4000
2000
0
January
February
March
April
May
June
July
August
September
October
November
December
Month


Crimes rise during Summer !
January
February
March
April
May
June
July
August
September
October
November
December
Month
0
2500
5000
7500
10000
12500
15000
17500
Number of Crimes


Unsafest Hours in Chicago
Number of Crimes
14000
12000
10000
8000
6000
4000
2000
0
22 00 18 11 03 09 16 08 19 20 21 17 02 12 15 13 10 14 23 06 01 05 07 04
Hour in the Day

Is the house safe from a burglary during the day ?

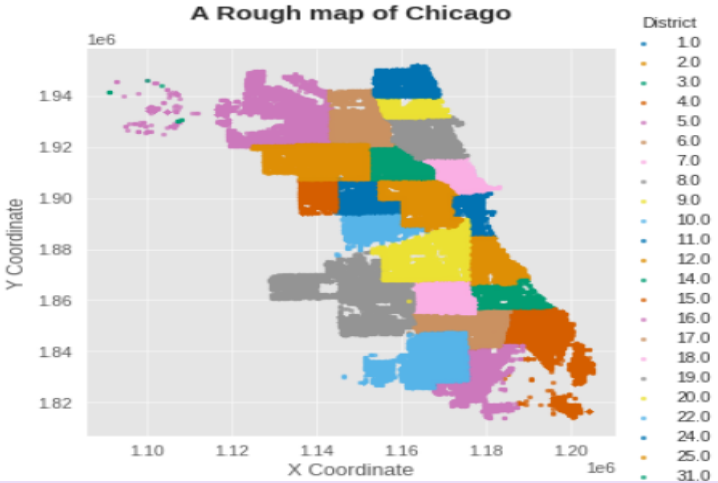


Crime vs Locations

+ Code + Text

```
sns.lmplot('X Coordinate',
           'Y Coordinate',
           data=df,
           fit_reg=False,
           hue="District",
           palette='colorblind',
           height=5,
           scatter_kws={"marker": "+",
                        "s": 10})
ax = plt.gca()
ax.set_title("A Rough map of Chicago\n", fontdict={'fontsize': 15}, weight="bold")
plt.show()
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following
  warnings.warn(
```



Heatmap


District vs Month
District
Month
1.0 34.000000 37.000000 18.000000 20.000000 21.000000 34.000000 44.000000 25.000000 41.000000 39.000000 34.000000 43.000000
2.0 40.000000 68.000000 64.000000 35.000000 46.000000 54.000000 58.000000 43.000000 47.000000 86.000000 76.000000 77.000000
3.0 78.000000 106.000000 79.000000 69.000000 90.000000 99.000000 89.000000 103.000000 92.000000 88.000000 101.000000 103.000000
4.0 91.000000 106.000000 80.000000 59.000000 110.000000 97.000000 112.000000 78.000000 109.000000 94.000000 112.000000 112.000000
5.0 103.000000 142.000000 132.000000 87.000000 105.000000 155.000000 107.000000 106.000000 134.000000 121.000000 111.000000 130.000000
6.0 102.000000 143.000000 87.000000 105.000000 135.000000 138.000000 140.000000 179.000000 139.000000 127.000000 109.000000 132.000000
7.0 122.000000 120.000000 72.000000 86.000000 126.000000 138.000000 174.000000 103.000000 156.000000 94.000000 125.000000 118.000000
8.0 93.000000 110.000000 107.000000 81.000000 97.000000 95.000000 101.000000 94.000000 114.000000 120.000000 115.000000 138.000000
9.0 68.000000 79.000000 98.000000 43.000000 69.000000 90.000000 86.000000 72.000000 83.000000 94.000000 90.000000 84.000000
10.0 105.000000 85.000000 99.000000 84.000000 100.000000 111.000000 97.000000 99.000000 77.000000 93.000000 113.000000 86.000000
11.0 278.000000 171.000000 145.000000 296.000000 311.000000 177.000000 254.000000 365.000000 316.000000 185.000000 176.000000 186.000000
12.0 48.000000 55.000000 58.000000 34.000000 53.000000 54.000000 52.000000 56.000000 55.000000 87.000000 67.000000 61.000000
14.0 45.000000 36.000000 47.000000 35.000000 35.000000 57.000000 45.000000 34.000000 41.000000 63.000000 78.000000 51.000000
15.0 73.000000 96.000000 63.000000 67.000000 65.000000 77.000000 98.000000 90.000000 97.000000 78.000000 87.000000 73.000000
16.0 52.000000 35.000000 54.000000 24.000000 43.000000 30.000000 43.000000 30.000000 55.000000 30.000000 42.000000 45.000000
17.0 31.000000 28.000000 45.000000 21.000000 53.000000 48.000000 26.000000 38.000000 36.000000 43.000000 44.000000 38.000000
18.0 34.000000 42.000000 53.000000 36.000000 45.000000 47.000000 61.000000 32.000000 43.000000 70.000000 73.000000 69.000000
19.0 43.000000 39.000000 67.000000 29.000000 45.000000 48.000000 51.000000 23.000000 25.000000 60.000000 53.000000 41.000000
20.0 19.000000 12.000000 21.000000 12.000000 17.000000 14.000000 11.000000 19.000000 7.000000 25.000000 24.000000 16.000000
22.0 72.000000 63.000000 54.000000 52.000000 50.000000 46.000000 50.000000 55.000000 44.000000 46.000000 75.000000 72.000000
24.0 29.000000 26.000000 36.000000 27.000000 29.000000 22.000000 37.000000 41.000000 21.000000 31.000000 42.000000 35.000000
25.0 95.000000 89.000000 100.000000 74.000000 102.000000 111.000000 116.000000 118.000000 101.000000 88.000000 105.000000 104.000000
31.0 1.000000
April August December February January July June March May November October September
350
300
250
200
150
100
50

Supervised Machine Learning

[60]

34465	10	5	31.0	7	1
34464	10	5	31.0	2	1
497	1	0	31.0	0	1

```
[61] cri4 = cri4[['Month_num','Day','Hour','Primary Type','District']]
     cri4.head()
     cri4.shape
```

```
(42047, 5)
```

```
[62] print(cri4['Primary Type'].max(),cri4['Primary Type'].min())
```

```
32 1
```

```
[63] print("Average no. of crime per month per day per district per hour :",cri4['Primary Type'].sum()/42774,".")
```

```
Average no. of crime per month per day per district per hour : 4.728666947210923 .
```

```
[64] # Feature Engineer and create a new feature
     def crime_rate_assign(x):
         if(x<=7):
             return 0
         elif(x>7 and x<=15):
             return 1
         else:
             return 2
     cri4['Alarm'] = cri4['Primary Type'].apply(crime_rate_assign)
     cri4 = cri4[['Month_num','Day','Hour','District','Primary Type','Alarm']]
     cri4.head()
```

B available


THANK YOU