Timed Challenge 5 – Nerdy_Potatoes

Data Loading and Exploration:

Reasoning: Load the data from the excel file into a pandas DataFrame and display the first 5 rows.

```
[ ] import matplotlib.pyplot as plt
    import seaborn as sns
```

```
import pandas as pd

try:
    df = pd.read_excel('Districtwise_Crime_of_India_2001_to_2014.xlsx')
    display(df.head())
except FileNotFoundError:
    print("Error: File not found. Please ensure the file 'Districtwise_Crime_of_India_2001_to_2014.xlsx' is in the current directory or provide the correct file path.")
    df = None
```

	Unnamed: 0	STATE/UT	DISTRICT	YEAR	MURDER	ATTEMPT TO MURDER	CULPABLE HOMICIDE NOT AMOUNTING TO MURDER	RAPE	CUSTODIAL RAPE	OTHER RAPE	...	ARSON	HURT/GREVIOUS HURT	DOWRY DEATHS	ASSAULT ON WOMEN WITH INTENT TO OUTRAGE HER MODESTY	INSULT TO MODESTY OF WOMEN	CRUELTY BY HUSBAND OR HIS RELATIVES	IMPORTATION OF GIRLS FROM FOREIGN COUNTRIES	CAUSING DEATH BY NEGLIGENCE	OTHER IPC CRIMES	TOTAL IPC CRIMES
0	0	ANDHRA PRADESH	ADILABAD	2001	101	60	17	50	0	50	...	30	1131	16	149	34	175	0	181	1518	4154
1	1	ANDHRA PRADESH	ANANTAPUR	2001	151	125	1	23	0	23	...	69	1543	7	118	24	154	0	270	754	4125
2	2	ANDHRA PRADESH	CHITTOOR	2001	101	57	2	27	0	27	...	38	2088	14	112	83	186	0	404	1262	5818
3	3	ANDHRA PRADESH	CUDDAPAH	2001	80	53	1	20	0	20	...	23	795	17	126	38	57	0	233	1181	3140
4	4	ANDHRA PRADESH	EAST GODAVARI	2001	82	67	1	23	0	23	...	41	1244	12	109	58	247	0	431	2313	6507

5 rows × 34 columns

We use pd.read_excel() to load the file in dataframe.

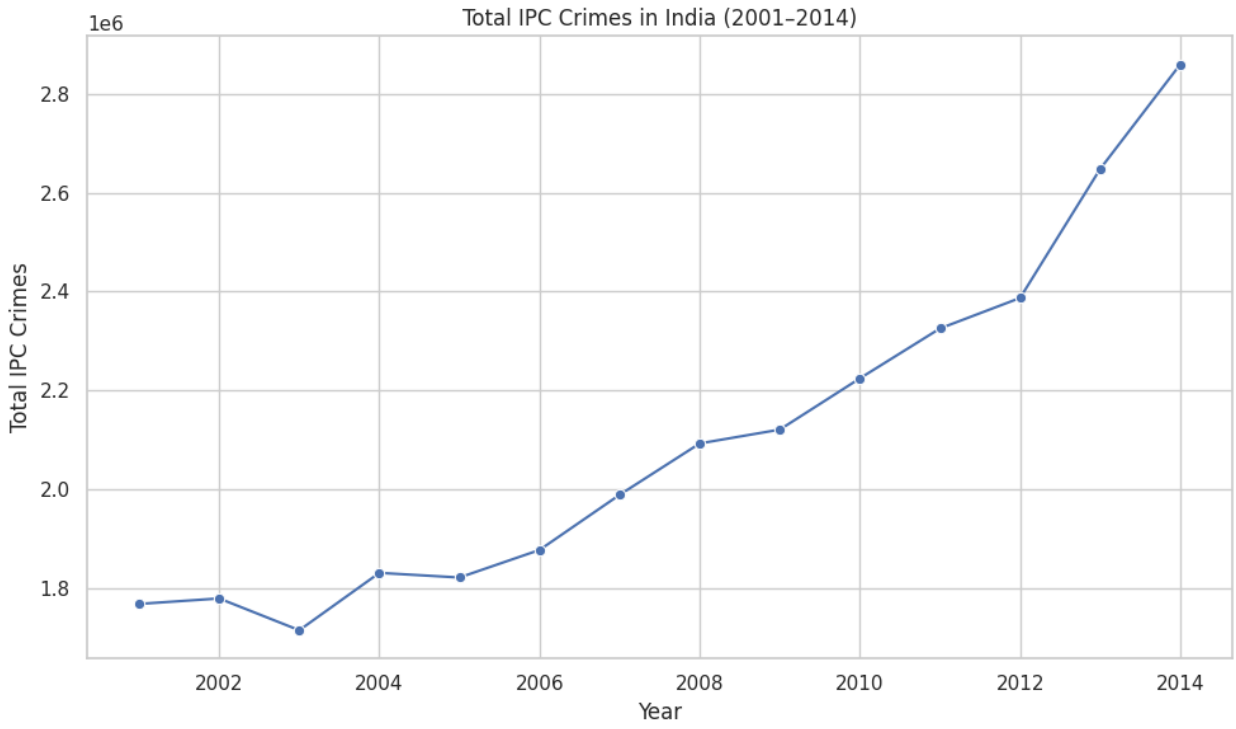
Dataset statistics:

```
🔍 Dataset Shape: (10187, 34)
📜 Years Covered: 2001 - 2014
🗺️ Unique Districts: 952
🗂️ Unique States/UTs: 38
❓ Missing Values:
 Unnamed: 0                                          0
STATE/UT                                             0
DISTRICT                                             0
YEAR                                                 0
MURDER                                               0
ATTEMPT TO MURDER                                    0
CULPABLE HOMICIDE NOT AMOUNTING TO MURDER            0
RAPE                                                 0
CUSTODIAL RAPE                                       0
OTHER RAPE                                           0
KIDNAPPING & ABDUCTION                               0
KIDNAPPING AND ABDUCTION OF WOMEN AND GIRLS          0
KIDNAPPING AND ABDUCTION OF OTHERS                   0
DACOITY                                              0
PREPARATION AND ASSEMBLY FOR DACOITY                 0
ROBBERY                                              0
BURGLARY                                             0
THEFT                                                0
AUTO THEFT                                           0
OTHER THEFT                                          0
RIOTS                                                0
CRIMINAL BREACH OF TRUST                             0
CHEATING                                             0
COUNTERFIETING                                       0
ARSON                                                0
HURT/GREVIOUS HURT                                   0
DOWRY DEATHS                                         0
ASSAULT ON WOMEN WITH INTENT TO OUTRAGE HER MODESTY  0
INSULT TO MODESTY OF WOMEN                           0
CRUELTY BY HUSBAND OR HIS RELATIVES                  0
IMPORTATION OF GIRLS FROM FOREIGN COUNTRIES          0
CAUSING DEATH BY NEGLIGENCE                          0
OTHER IPC CRIMES                                     0
TOTAL IPC CRIMES                                     0
dtype: int64
```

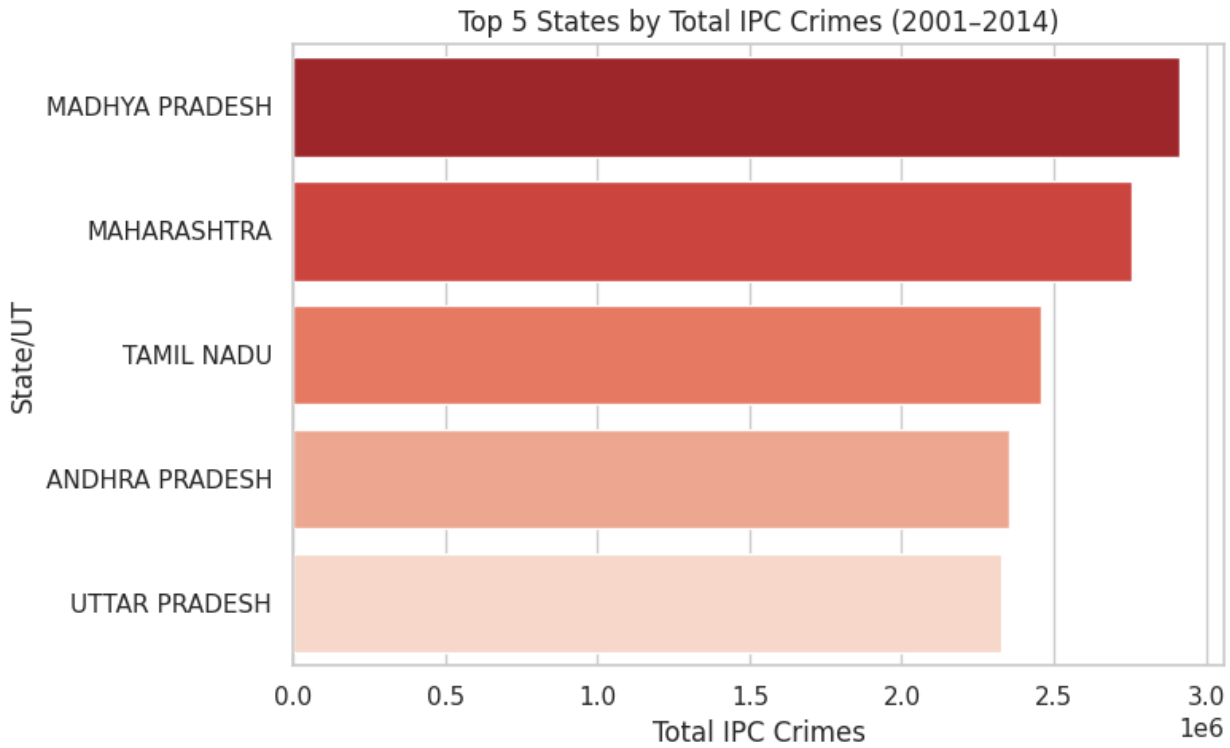
Top 5 crime types :

```
📊 Top 5 Crime Types:
 TOTAL IPC CRIMES      29447315
OTHER IPC CRIMES       11839559
THEFT                   4315616
HURT/GREVIOUS HURT      3812105
OTHER THEFT             2693521
dtype: int64
```

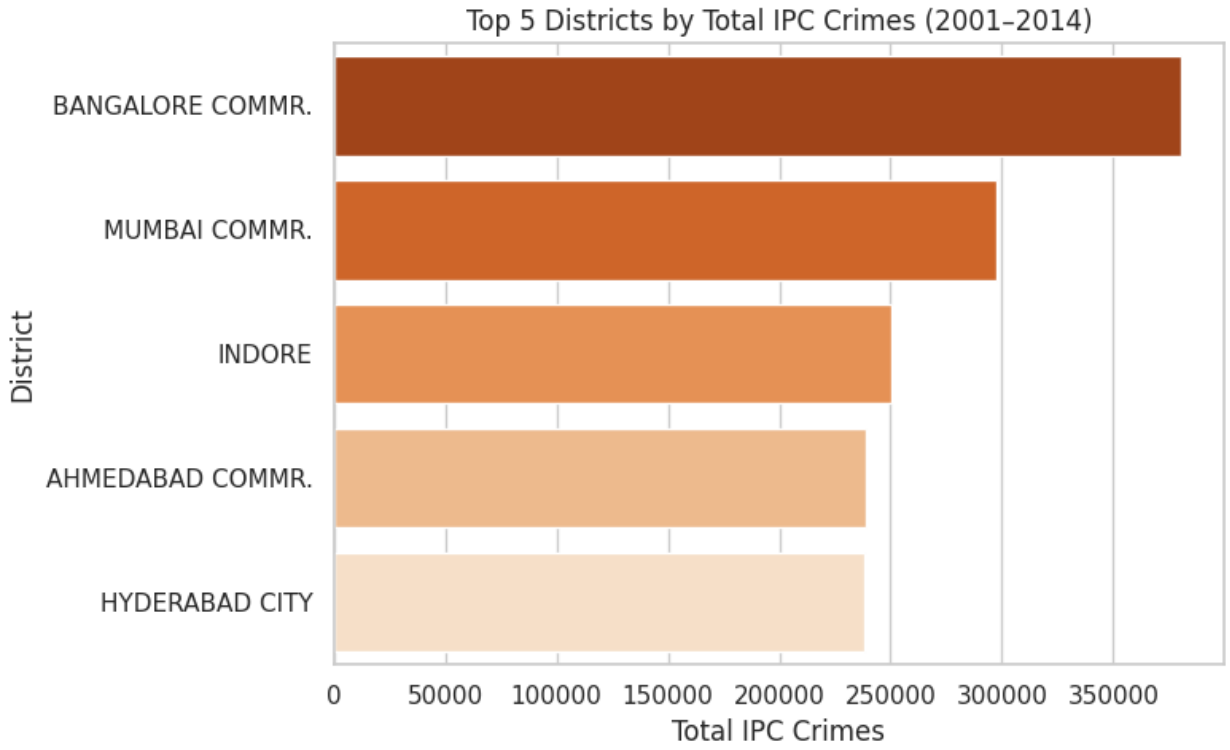
Graph of total crimes in India :



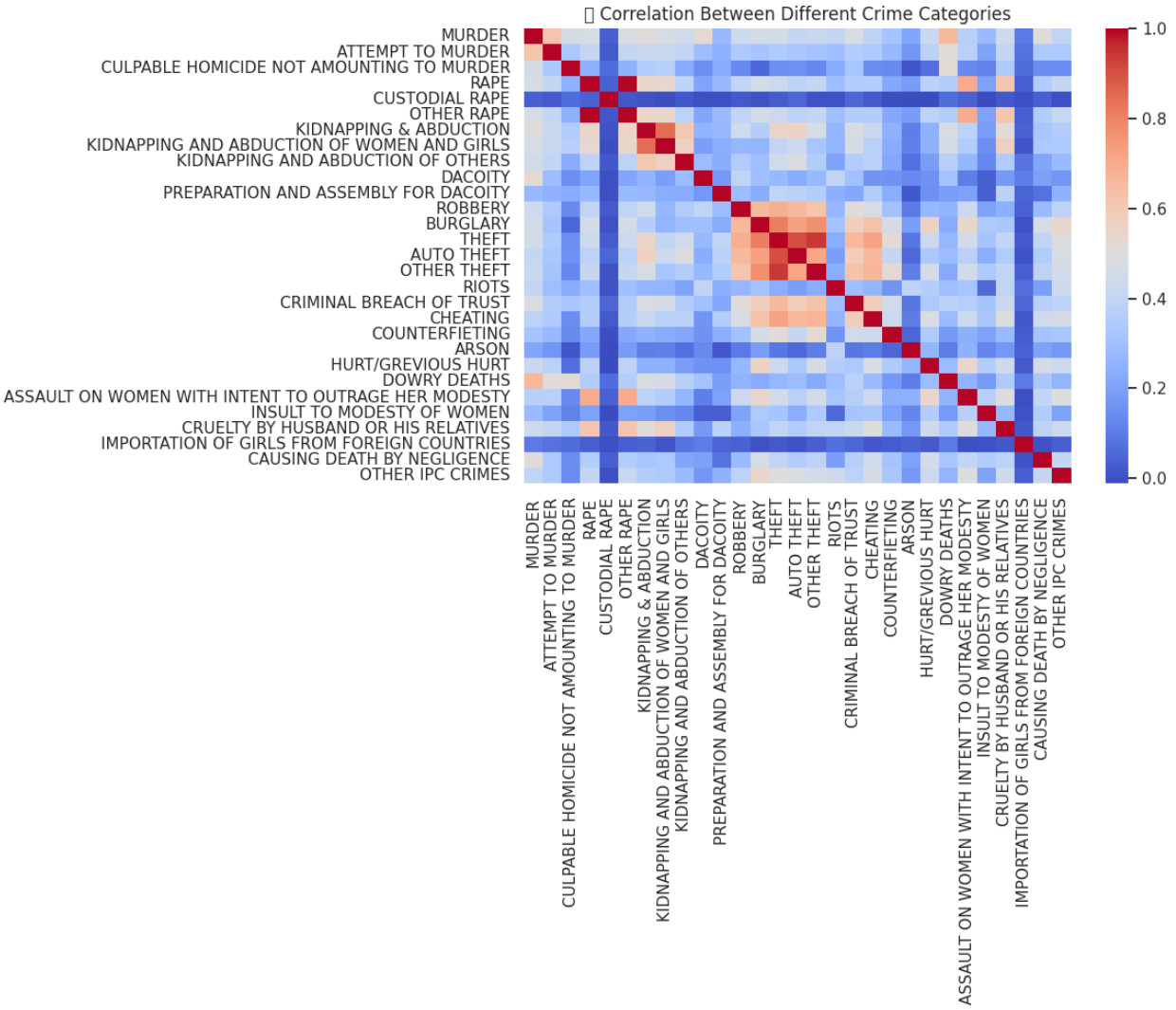
Top 5 States by IPC crimes :



Top 5 Districts by IPC crimes :



Correlation Matrix :



EDA Questions

- Determine the total number of crimes recorded across all districts and the average number of murders per district.

This shows total crimes and overall average

```python
import pandas as pd

# Load Excel file
file_path = 'Districtwise_Crime_of_India_2001_to_2014.xlsx'
df = pd.read_excel(file_path, sheet_name='Sheet1')

# Clean column names
df.columns = df.columns.str.strip()

# Remove rows where 'DISTRICT' is a TOTAL summary
df = df[~df['DISTRICT'].str.upper().str.contains('TOTAL')]

# Calculate average murders per district
average_murders_per_district = df.groupby(['STATE/UT', 'DISTRICT'])['MURDER'].mean().reset_index()

# Rename column for clarity
average_murders_per_district.rename(columns={'MURDER': 'Average Murders'}, inplace=True)

# Sort by highest average murders
average_murders_per_district = average_murders_per_district.sort_values(by='Average Murders', ascending=False)

# Display all rows (optional)
pd.set_option('display.max_rows', None)

# Print the result
print(average_murders_per_district)
```

```
            STATE/UT                DISTRICT  Average Murders
140            BIHAR                   PATNA       370.642857
553      MAHARASHTRA                  MUMBAI       250.800000
995       WEST BENGAL       SOUTH 24 PARGANAS       248.000000
400        KARNATAKA          BENGALURU CITY       241.000000
833        TELANGANA           MAHABOOB NAGAR       240.000000
394        KARNATAKA         BANGALORE COMMR.       234.461538
554      MAHARASHTRA           MUMBAI COMMR.       210.555556
388        JHARKHAND                  RANCHI       206.285714
914    UTTAR PRADESH                  MEERUT       204.428571
917    UTTAR PRADESH          MUZAFFARNAGAR       200.571429
951       WEST BENGAL      24 PARGANAS NORTH       189.615385
```

The above Screenshot shows average murders per district.

- Examine how crime distributions vary across different states, and identify the top 5 districts with the highest total IPC crimes.

```python
# Group data by state and sum total crimes
state_crime_distribution = df.groupby('STATE/UT')['TOTAL IPC CRIMES'].sum()

# Print the distribution
print("Crime Distribution across States:\n", state_crime_distribution)
```

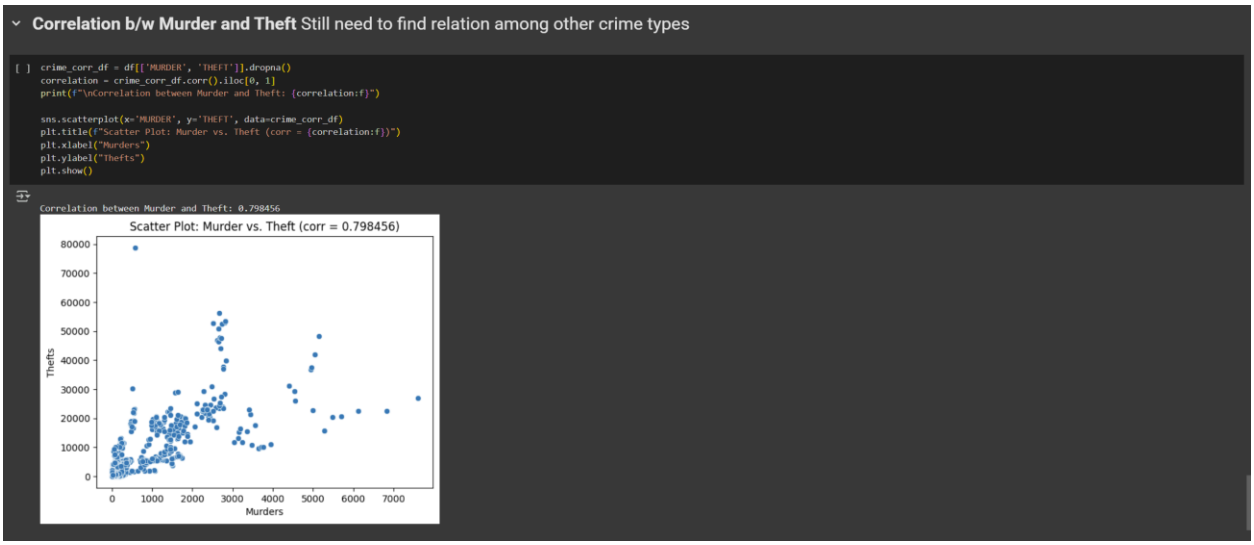
```
Crime Distribution across States:
 STATE/UT
A & N ISLANDS          19428
A&N ISLANDS             1492
ANDHRA PRADESH       4703200
ARUNACHAL PRADESH      66542
ASSAM                1558574
BIHAR                3382686
CHANDIGARH             96210
CHHATTISGARH         1352194
D & N HAVELI            9930
D&N HAVELI               554
DAMAN & DIU             6854
DELHI UT             1738024
GOA                    81658
GUJARAT              3349190
HARYANA              1494696
HIMACHAL PRADESH      365716
JAMMU & KASHMIR       616786
JHARKHAND            1031788
KARNATAKA            3510180
KERALA               3641164
LAKSHADWEEP             1728
MADHYA PRADESH       5827292
MAHARASHTRA          5515310
MANIPUR                83782
MEGHALAYA              64374
MIZORAM                59990
NAGALAND               31012
ODISHA               1588466
PUDUCHERRY            122912
PUNJAB                915920
RAJASTHAN            4525116
SIKKIM                 17832
TAMIL NADU           4913910
TELANGANA             213660
TRIPURA               128886
UTTAR PRADESH        4649988
UTTARAKHAND           243812
WEST BENGAL          2963774
Name: TOTAL IPC CRIMES, dtype: int64
```

Top 5 Districts by Total IPC Crimes

```python
filtered_df = df[~df['DISTRICT'].str.upper().str.contains('TOTAL')]
top_5_districts = filtered_df.groupby(['STATE/UT', 'DISTRICT'])['TOTAL IPC CRIMES'].sum().sort_values(ascending=False).head(5)
print("\nTop 5 Districts by Total IPC Crimes:\n", top_5_districts)
```

```
Top 5 Districts by Total IPC Crimes:
 STATE/UT        DISTRICT
KARNATAKA       BANGALORE COMMR.    380665
MAHARASHTRA     MUMBAI COMMR.       297871
MADHYA PRADESH  INDORE              250639
GUJARAT         AHMEDABAD COMMR.    239263
ANDHRA PRADESH  HYDERABAD CITY      219286
Name: TOTAL IPC CRIMES, dtype: int64
```

- Further, analyze how crime patterns differ across various crime categories in urban vs. rural districts (or using a proxy like population if urban/rural data is unavailable) and investigate whether there is a correlation between different crime types such as murder and theft.

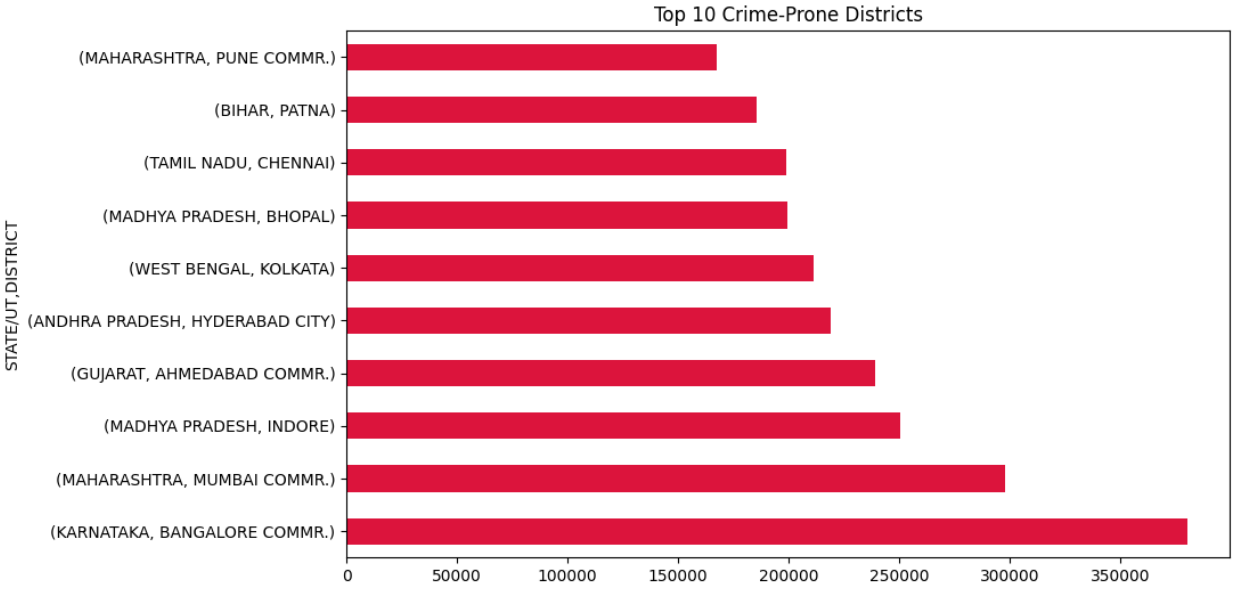


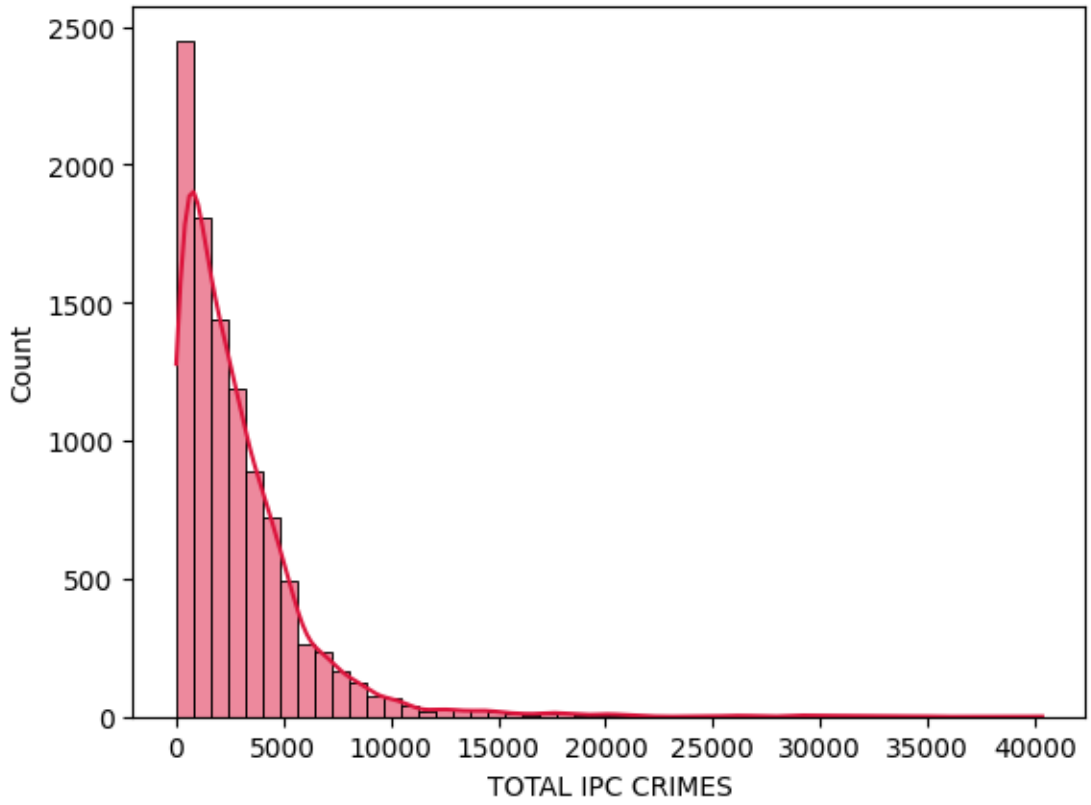
The correlation between Murder and theft is 0.79 which is usually between -1 and 1. Since it is 0.79 it has a positive correlation between them, which means if one increases the other will increase too.

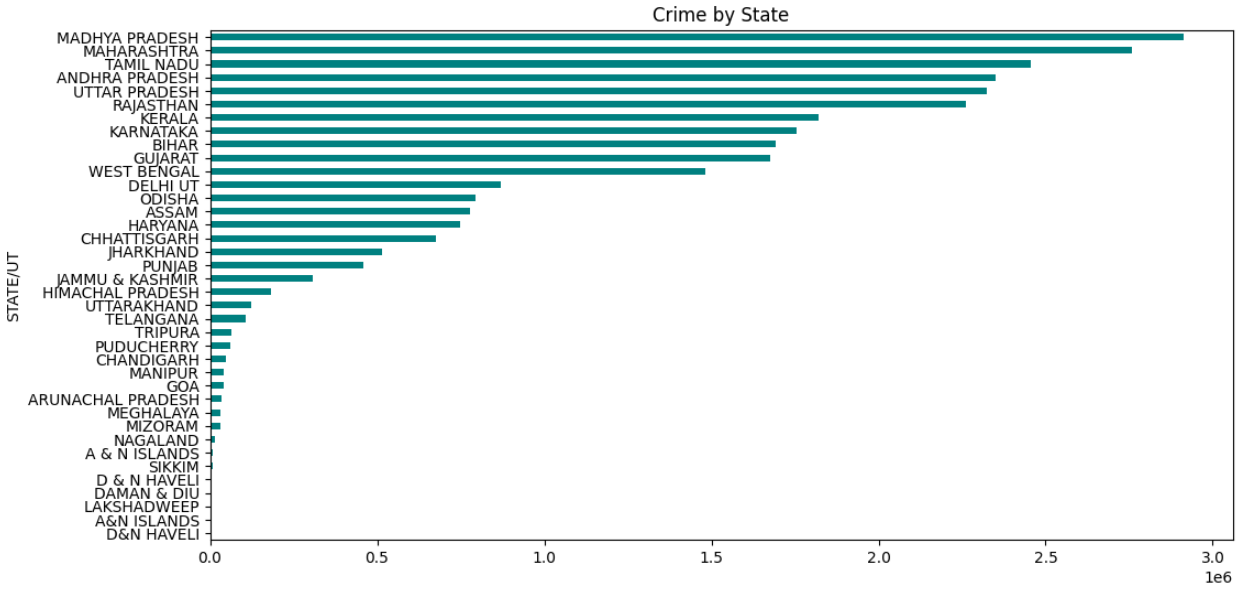
Visualization Questions

- How can visualizations be used to explore crime patterns in India by identifying the top 10 districts with the highest crime rates, understanding the overall distribution of total IPC crimes, analyzing crime density across different states, and comparing trends in violent crimes such as murder and rape across various districts?

  We have some of the basic visualizations that answers the above questions.

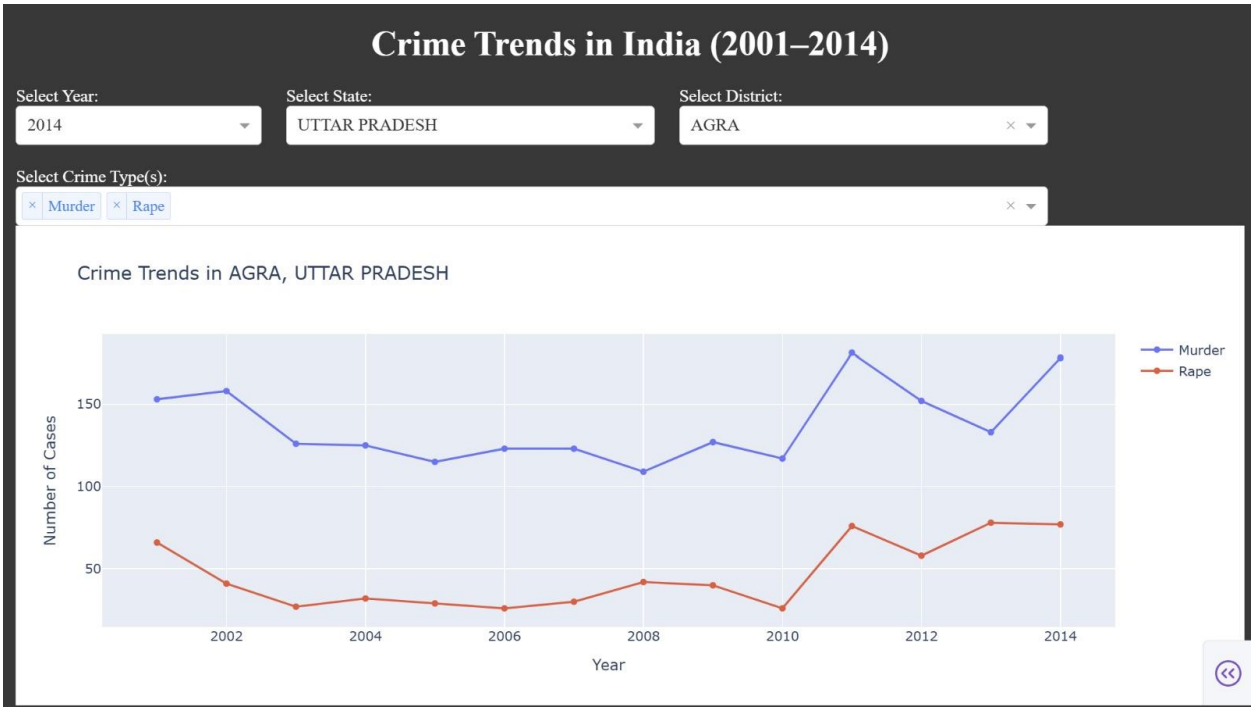

Top 10 Crime-Prone Districts
STATE/UT,DISTRICT
(MAHARASHTRA, PUNE COMMR.)
(BIHAR, PATNA)
(TAMIL NADU, CHENNAI)
(MADHYA PRADESH, BHOPAL)
(WEST BENGAL, KOLKATA)
(ANDHRA PRADESH, HYDERABAD CITY)
(GUJARAT, AHMEDABAD COMMR.)
(MADHYA PRADESH, INDORE)
(MAHARASHTRA, MUMBAI COMMR.)
(KARNATAKA, BANGALORE COMMR.)
0
50000
100000
150000
200000
250000
300000
350000


Count
2500
2000
1500
1000
500
0
0
5000
10000
15000
20000
25000
30000
35000
40000
TOTAL IPC CRIMES



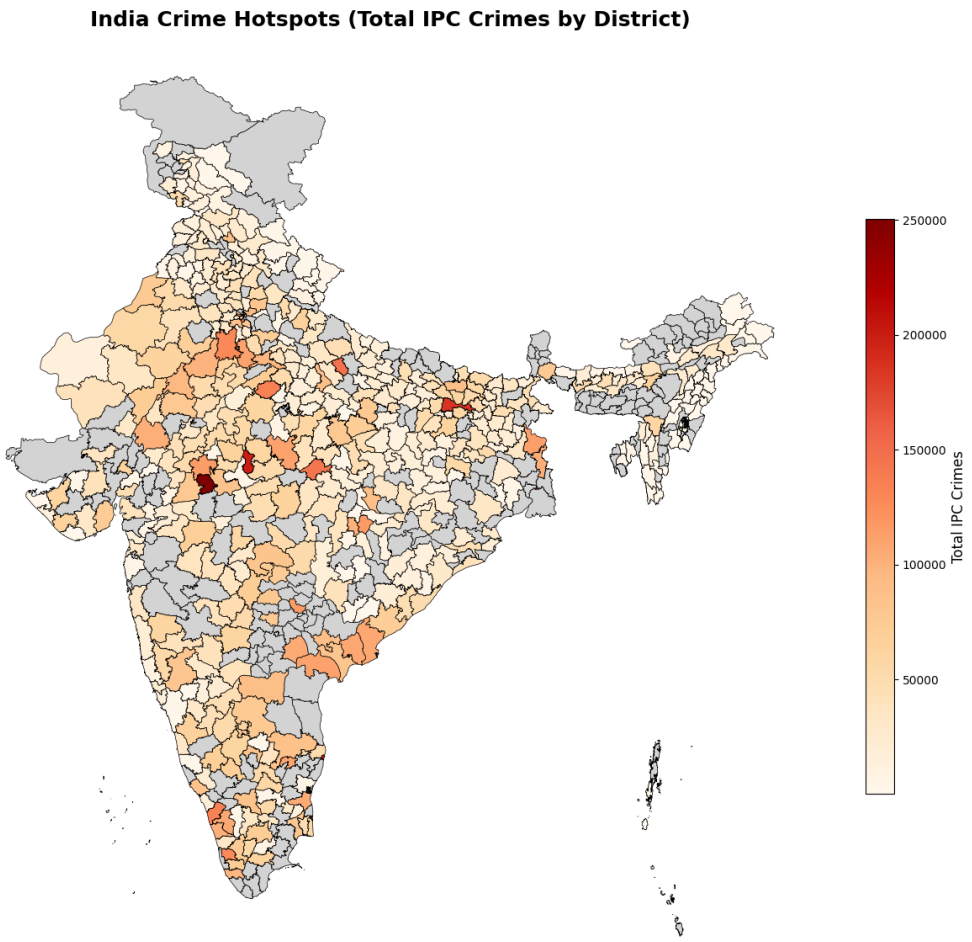
- Create an interactive dashboard that allows users to filter crime data by year, state, and district.

  We have used plotly dash to create a interactive dashboard with various filter options.



- Use a geospatial map to visualize crime hot spots across India. (Matplotlib)

You can see geospatial analysis where districts, we have used geojson file In background.



Advanced Questions

- Identify the state with the lowest crime rate and analyze why it might be lower than others.

```python
import pandas as pd

# Load and prepare data
df = pd.read_csv("Districtwise_Crime_of_India_2001_to_2014 - Sheet1.csv")
df.columns = df.columns.str.strip()
df = df[~df['DISTRICT'].str.upper().str.contains('TOTAL')]
df['TOTAL IPC CRIMES'] = pd.to_numeric(df['TOTAL IPC CRIMES'], errors='coerce')

# Aggregate total crimes by state
state_crimes = df.groupby('STATE/UT')['TOTAL IPC CRIMES'].sum().sort_values()

# Identify state with the lowest total IPC crimes
lowest_crime_state = state_crimes.idxmin()
lowest_crime_value = state_crimes.min()

print(f"State with the lowest crime rate: {lowest_crime_state}")
print(f"Total IPC crimes recorded (2001-2014): {lowest_crime_value:,}")
```

```
State with the lowest crime rate: D&N HAVELI
Total IPC crimes recorded (2001-2014): 277
```

- Find the most common type of crime committed in each district.

Most Common Crime Type by District

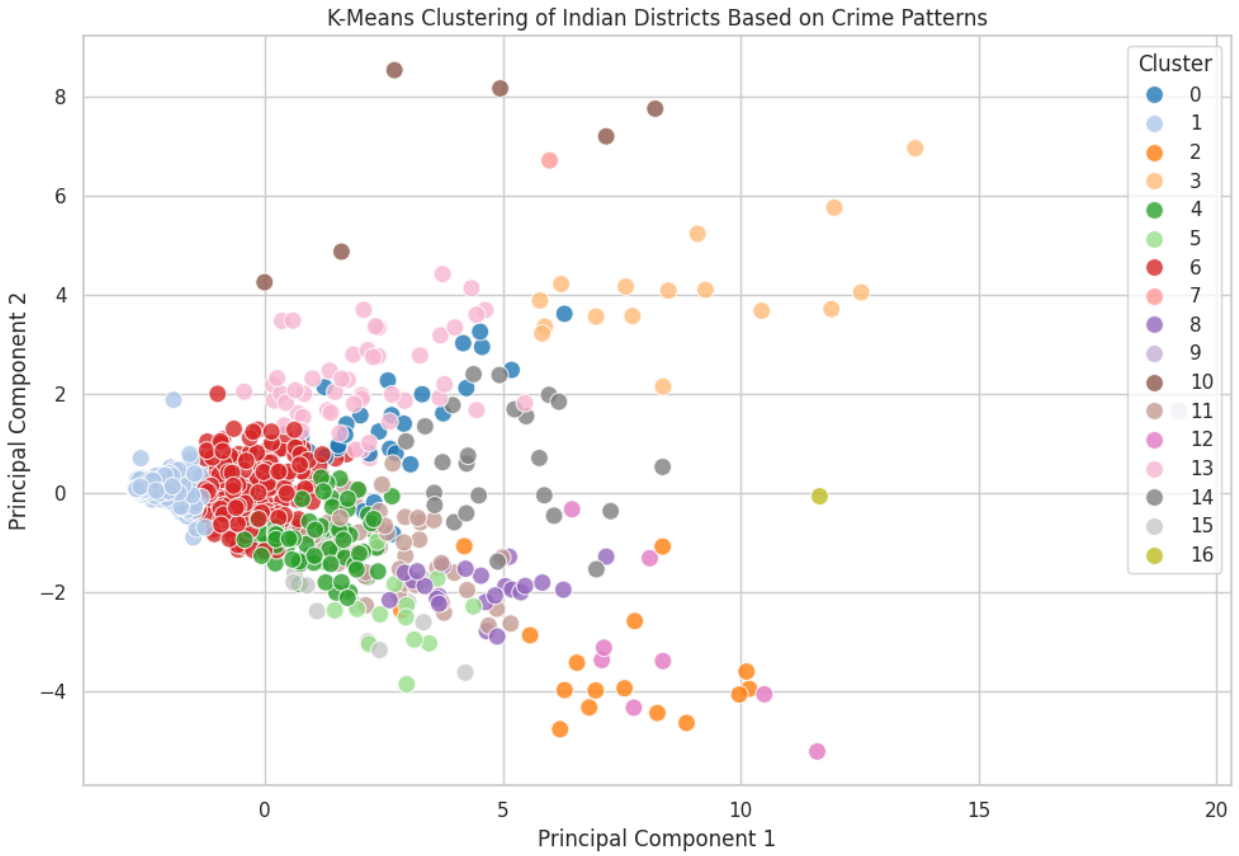
Select a State:

UTTAR PRADESH

District	Most Common Crime	Count
AGRA	OTHER IPC CRIMES	23901
ALIGARH	OTHER IPC CRIMES	17893
ALLAHABAD	OTHER IPC CRIMES	27052
AMBEDKAR NAGAR	OTHER IPC CRIMES	4640
AMETHI	OTHER IPC CRIMES	1006
AMROHA	OTHER IPC CRIMES	1986
AURAIYA	OTHER IPC CRIMES	5192
AZAMGARH	OTHER IPC CRIMES	7733
BADAUN	OTHER IPC CRIMES	13004
BAGHPAT	OTHER IPC CRIMES	5454
BAHRAICH	OTHER IPC CRIMES	8836

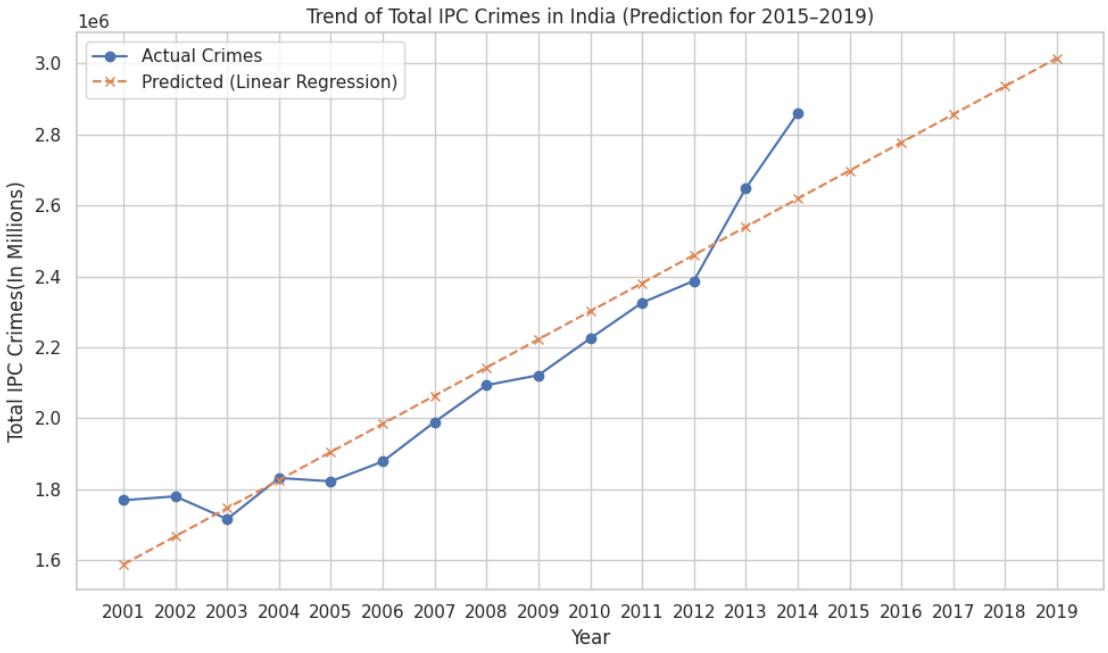
- Apply clustering algorithms (e.g., K-Means) to group districts based on crime patterns.

  We have used k means to cluster each crime category and districts as shown below.

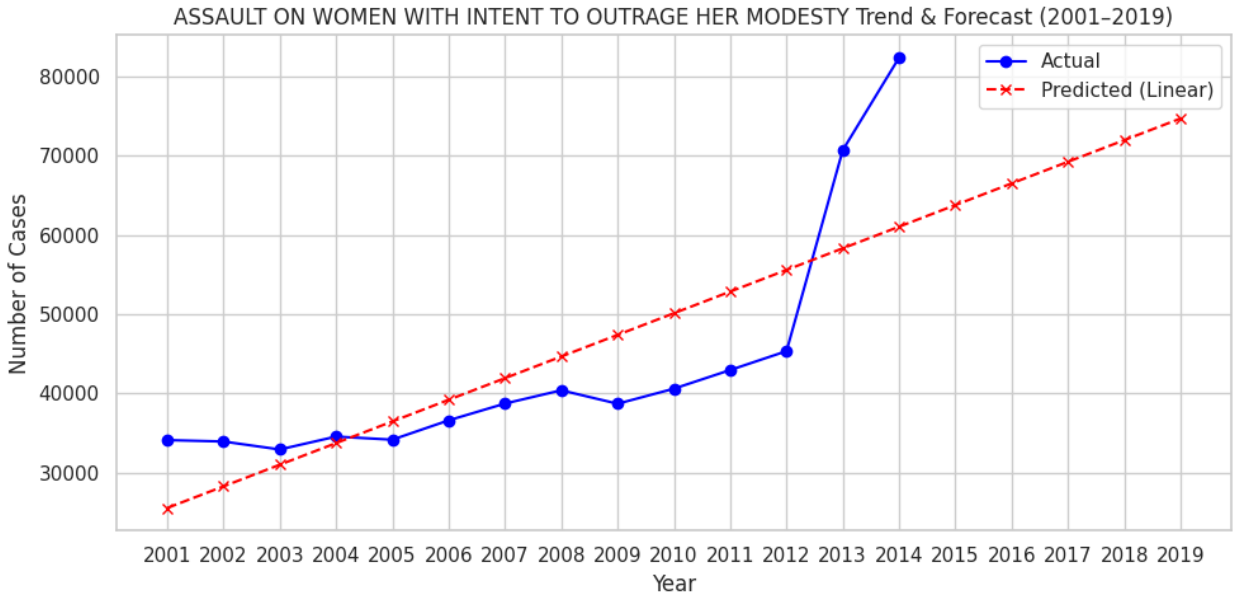


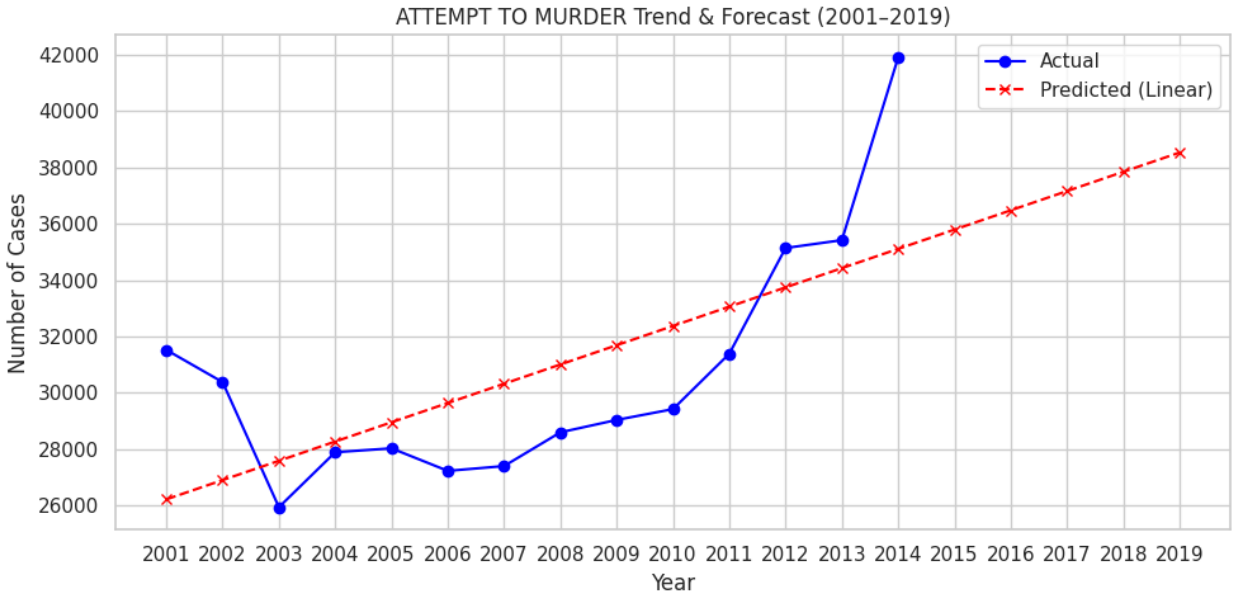
- Predict future crime trends using regression analysis.

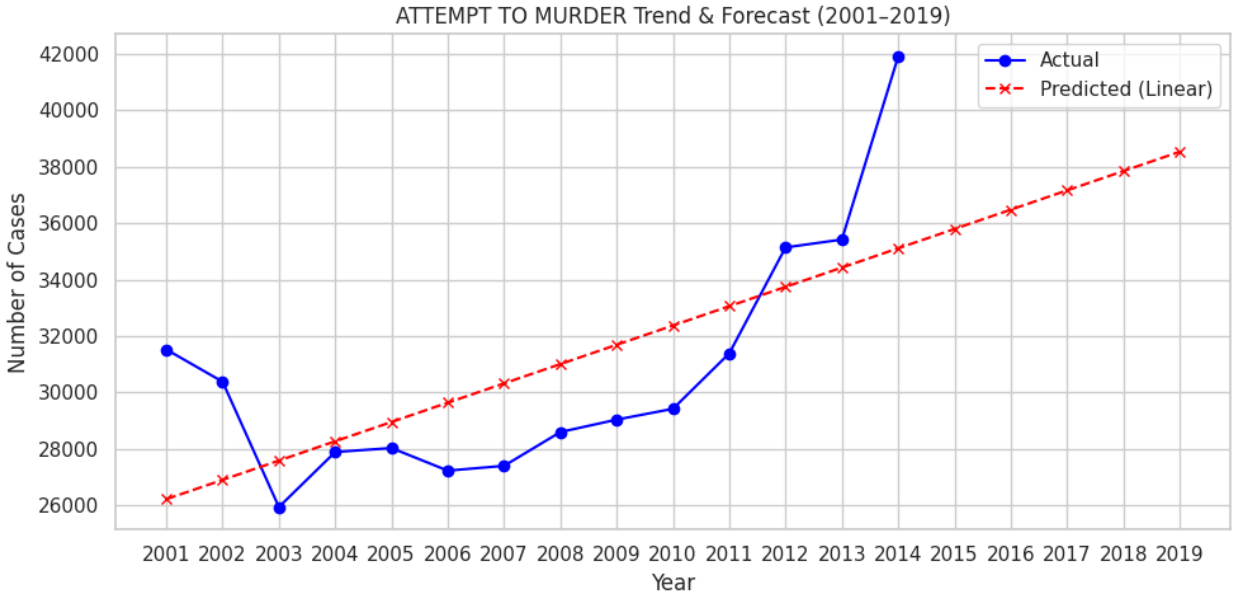
  Below is the regression graph for total IPC crimes in India

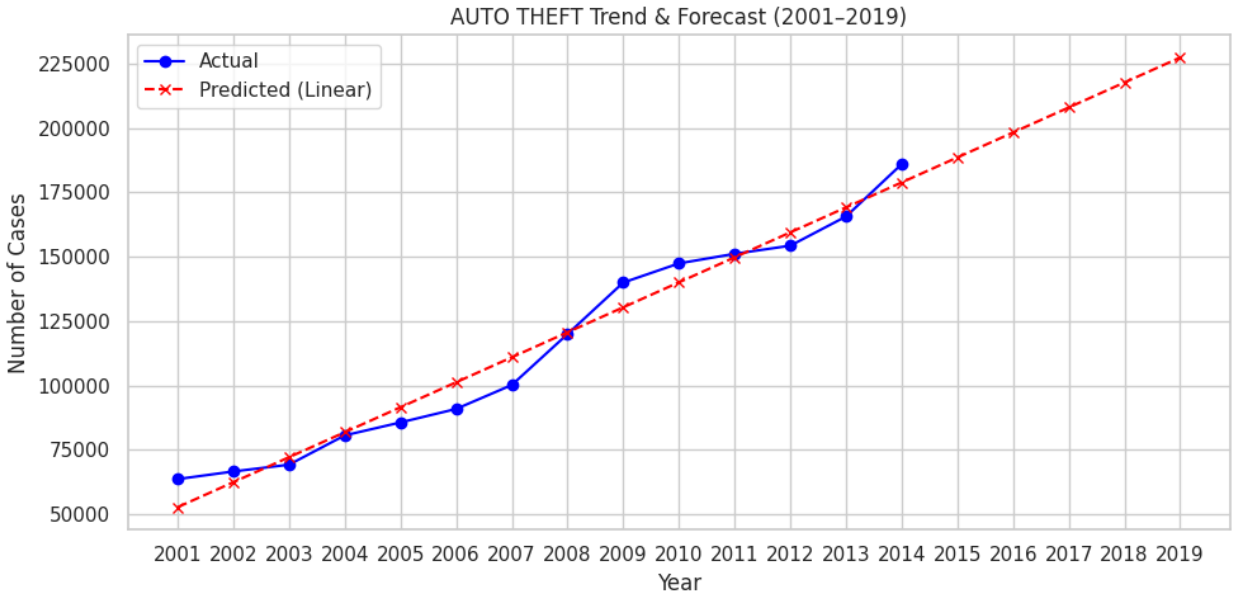


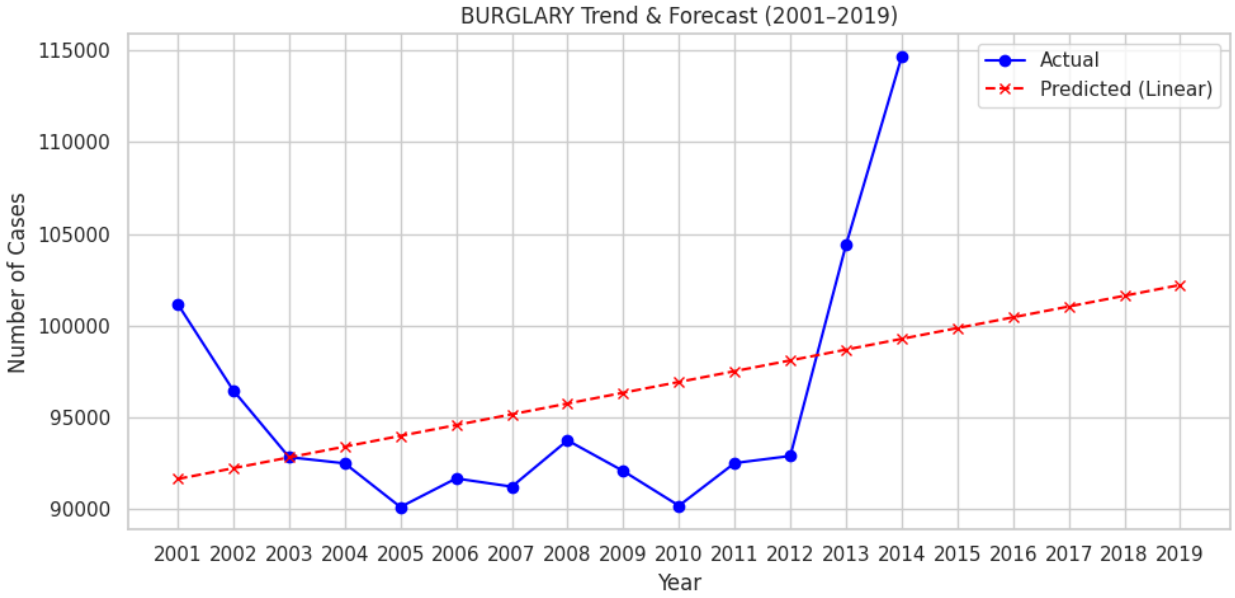
Below are the regressions for each type of crime.

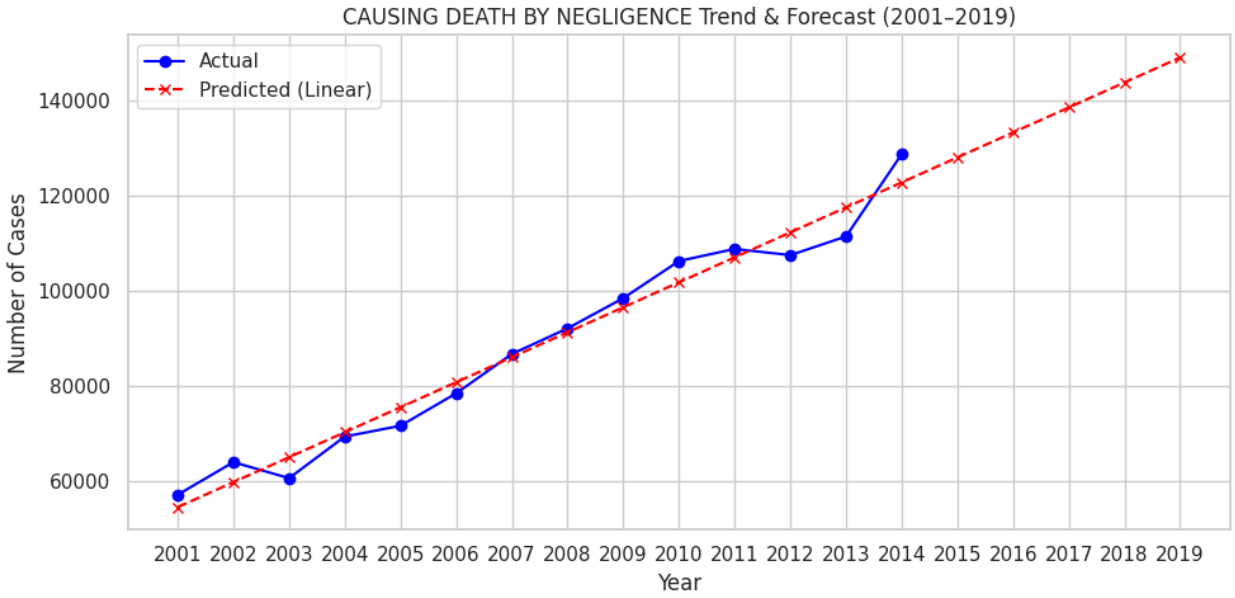


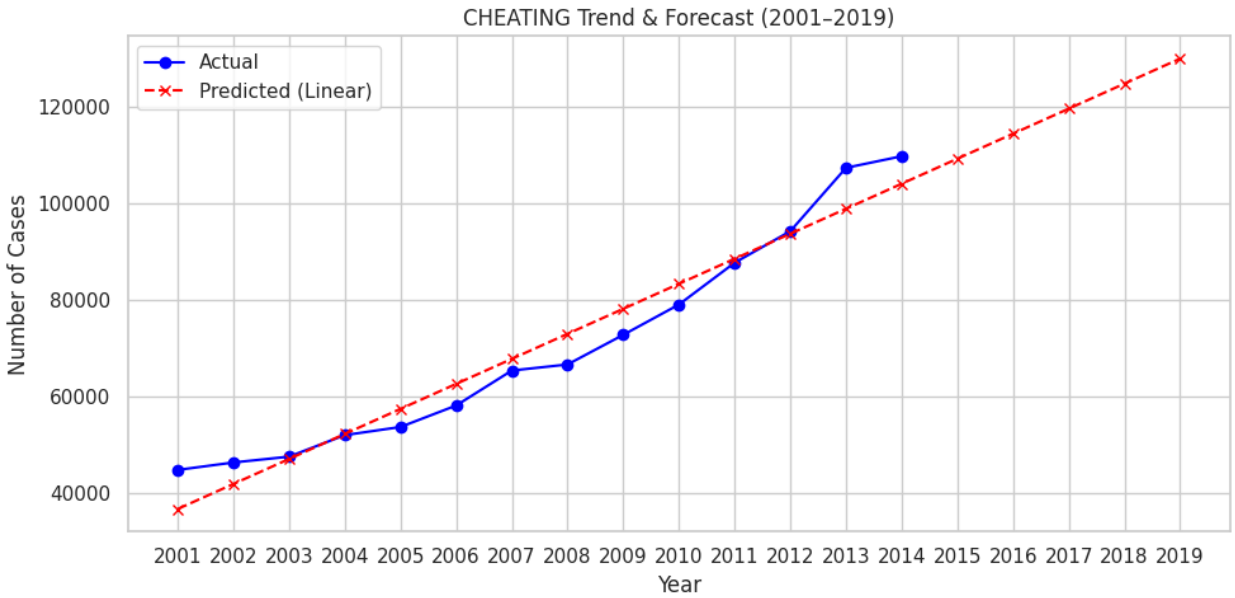

ATTEMPT TO MURDER Trend & Forecast (2001–2019)
Actual
Predicted (Linear)
Number of Cases
Year
42000
40000
38000
36000
34000
32000
30000
28000
26000
2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019

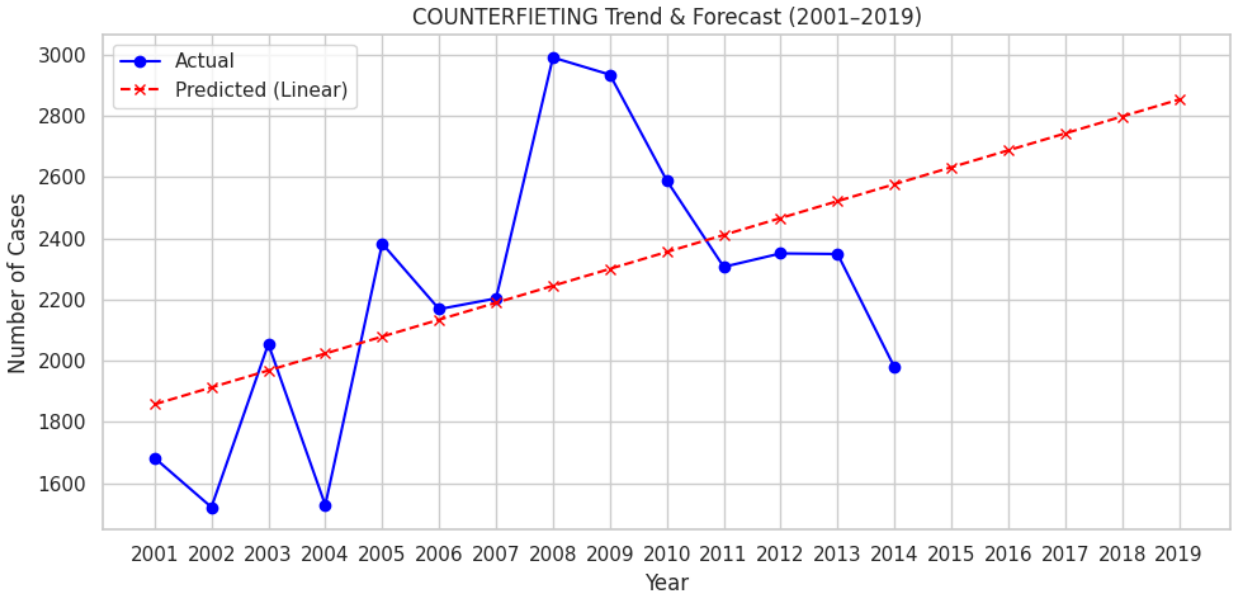

ATTEMPT TO MURDER Trend & Forecast (2001–2019)
Actual
Predicted (Linear)
Number of Cases
Year
42000
40000
38000
36000
34000
32000
30000
28000
26000
2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019

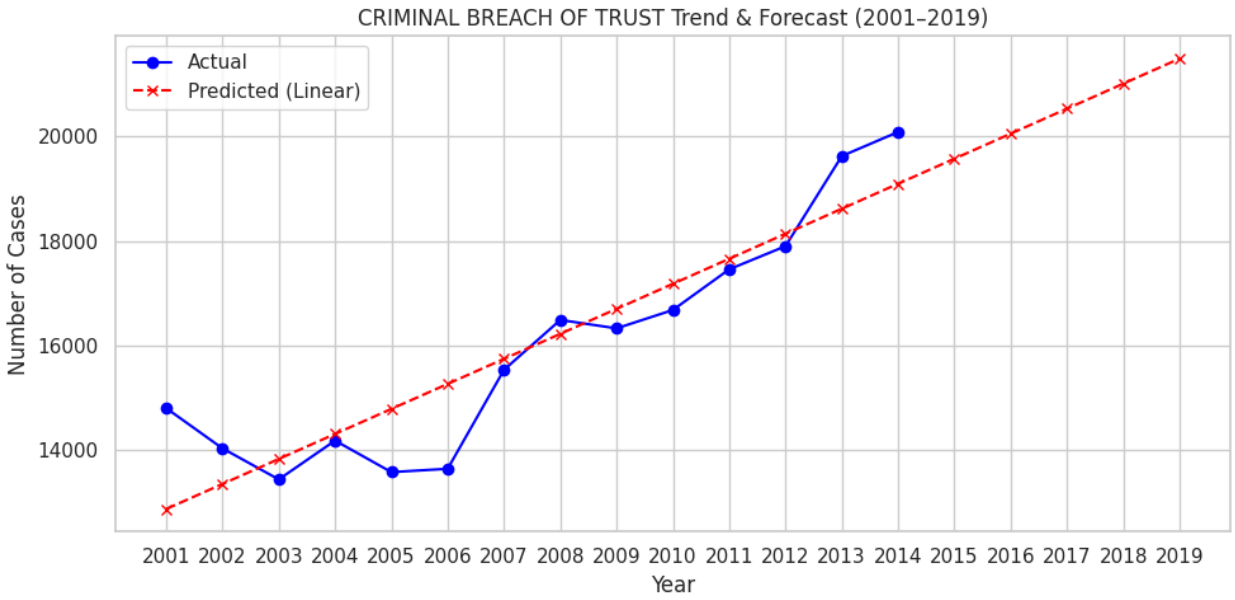

AUTO THEFT Trend & Forecast (2001–2019)
Actual
Predicted (Linear)
Number of Cases
225000
200000
175000
150000
125000
100000
75000
50000
2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019
Year

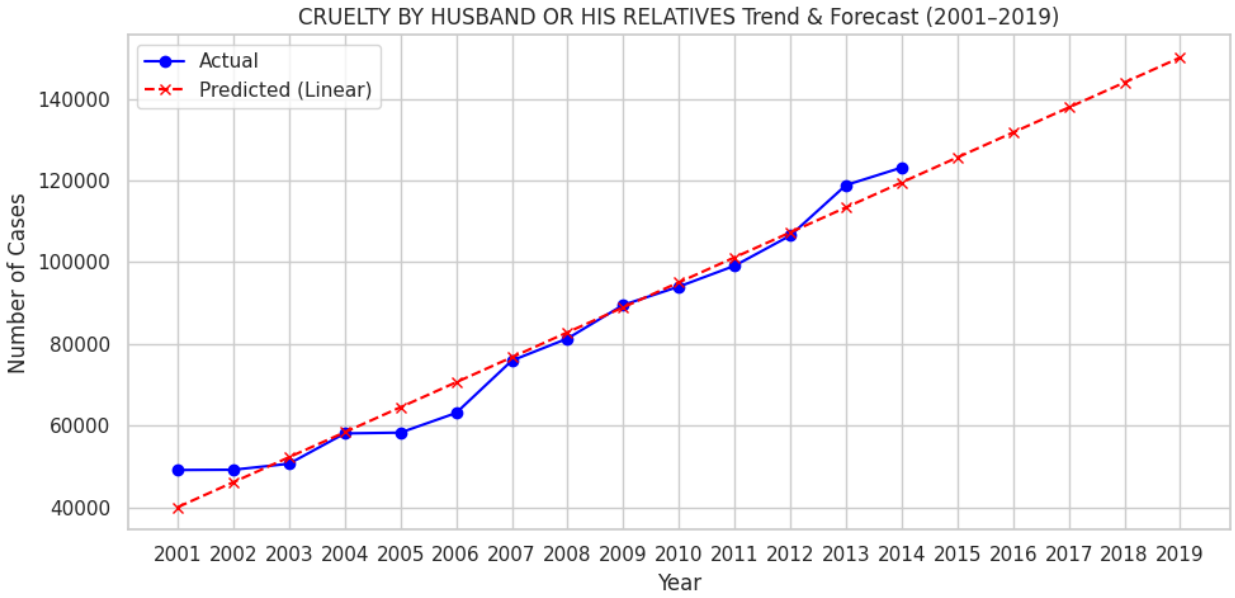

BURGLARY Trend & Forecast (2001–2019)
Actual
Predicted (Linear)
Number of Cases
115000
110000
105000
100000
95000
90000
2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019
Year

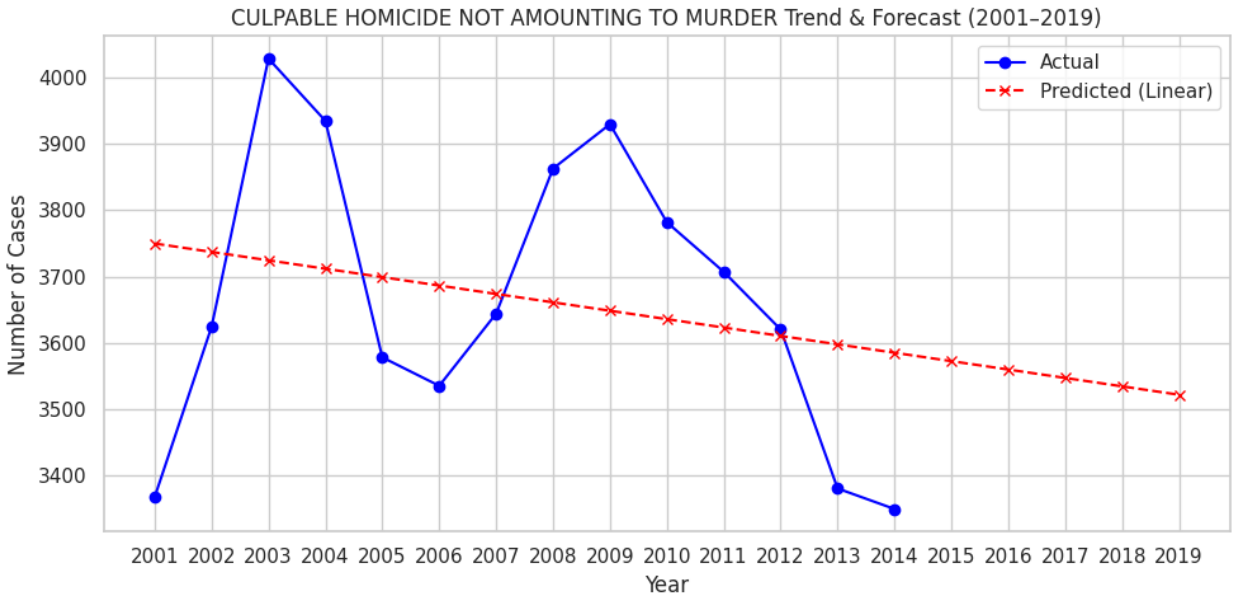

CAUSING DEATH BY NEGLIGENCE Trend & Forecast (2001–2019)
Actual
Predicted (Linear)
Number of Cases
140000
120000
100000
80000
60000
2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019
Year

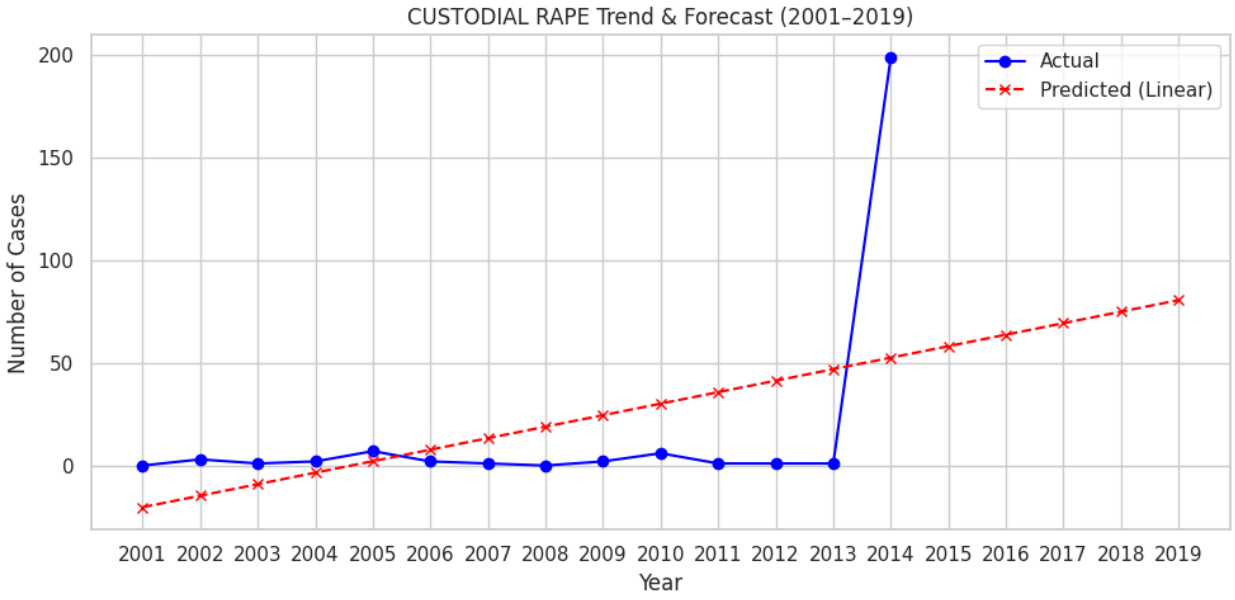

CHEATING Trend & Forecast (2001–2019)
Actual
Predicted (Linear)
Number of Cases
120000
100000
80000
60000
40000
2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019
Year

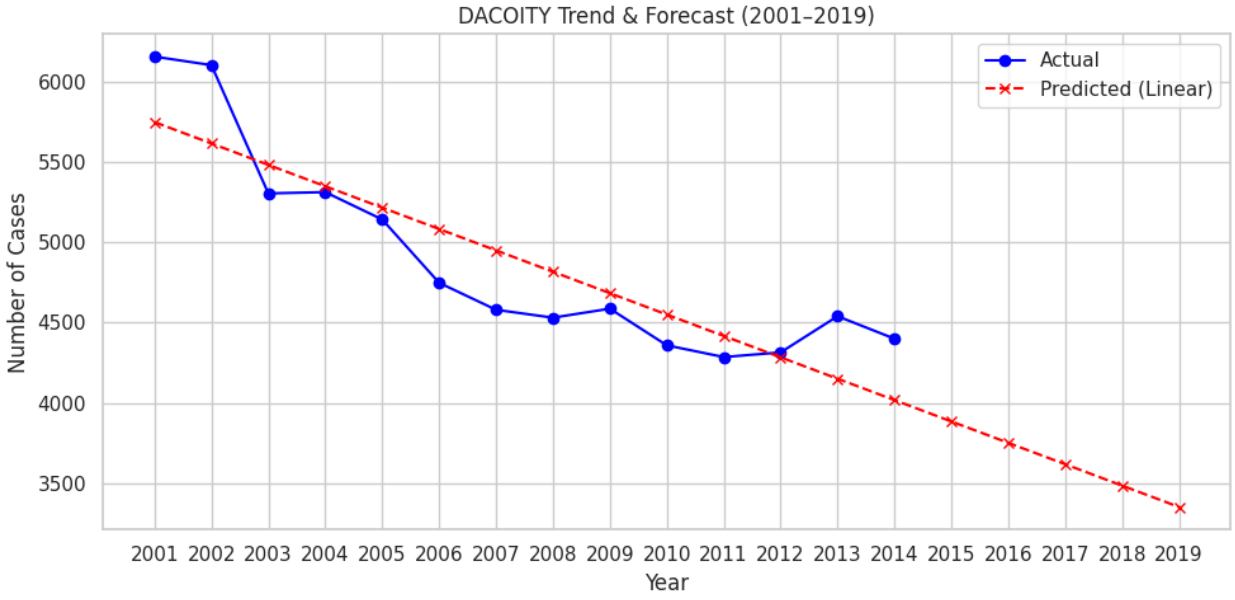

COUNTERFIETING Trend & Forecast (2001–2019)
Actual
Predicted (Linear)
Number of Cases
3000
2800
2600
2400
2200
2000
1800
1600
2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019
Year

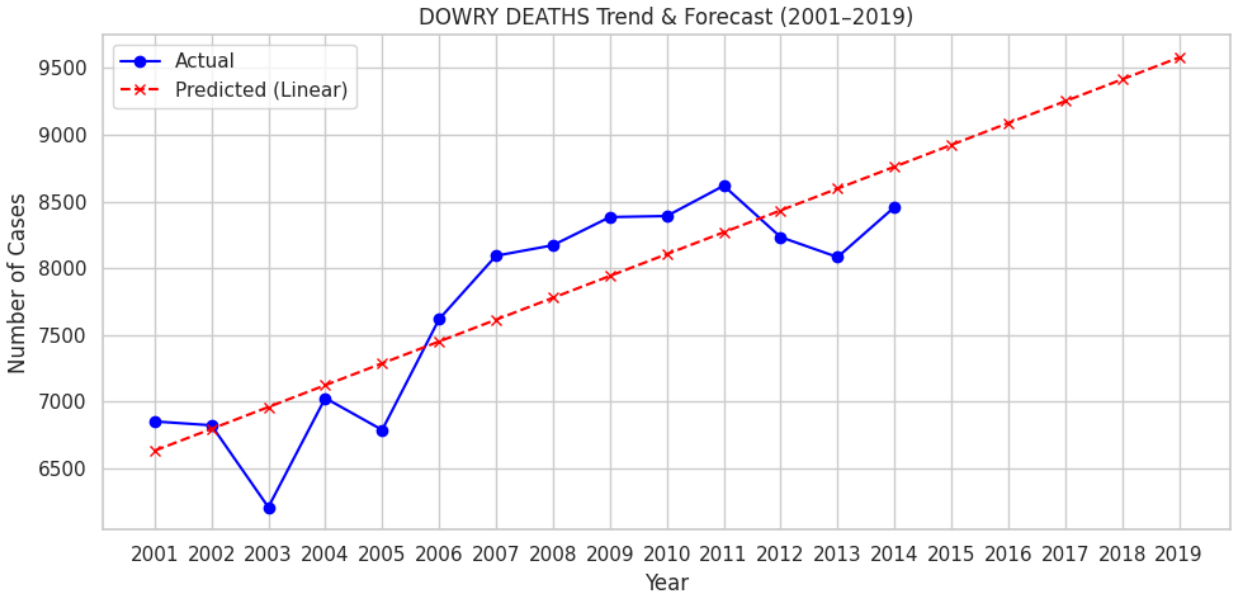

CRIMINAL BREACH OF TRUST Trend & Forecast (2001–2019)
Actual
Predicted (Linear)
Number of Cases
20000
18000
16000
14000
2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019
Year

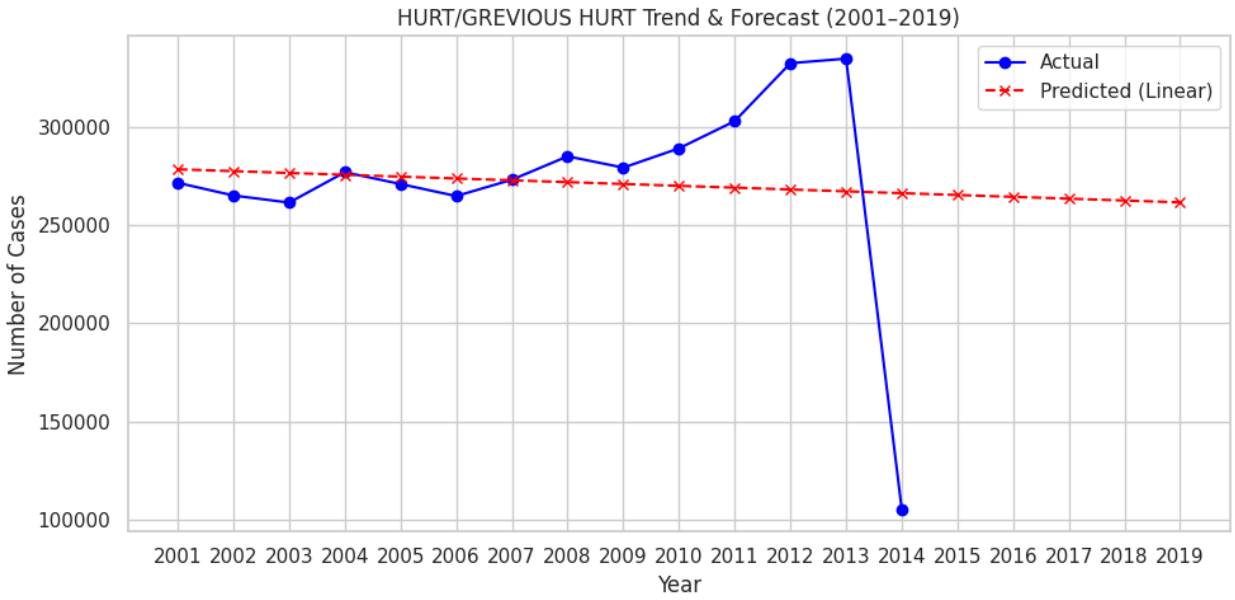

CRUELTY BY HUSBAND OR HIS RELATIVES Trend & Forecast (2001–2019)
Actual
Predicted (Linear)
Number of Cases
140000
120000
100000
80000
60000
40000
2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019
Year

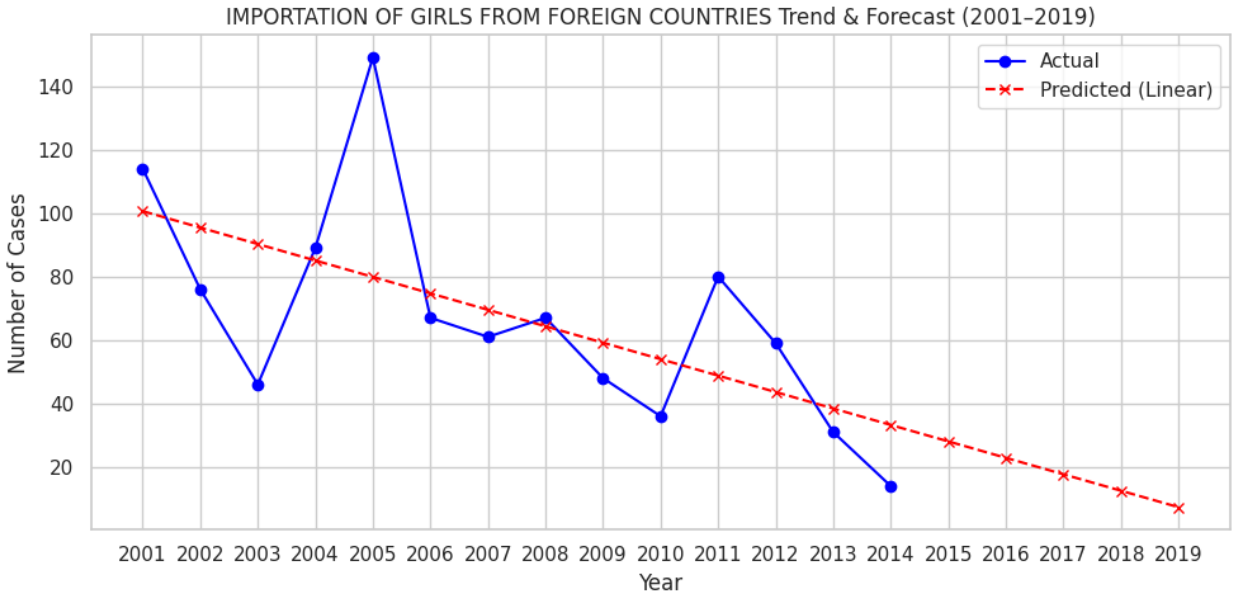

CULPABLE HOMICIDE NOT AMOUNTING TO MURDER Trend & Forecast (2001–2019)
Actual
Predicted (Linear)
Number of Cases
4000
3900
3800
3700
3600
3500
3400
2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019
Year

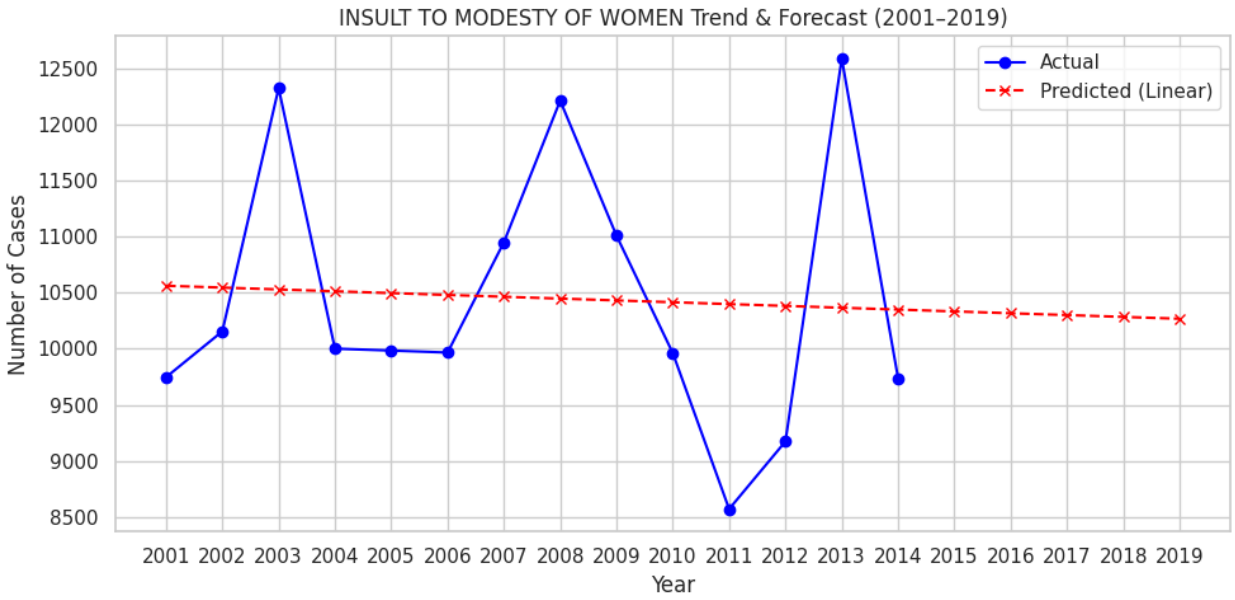

CUSTODIAL RAPE Trend & Forecast (2001–2019)
Actual
Predicted (Linear)
Number of Cases
Year
200
150
100
50
0
2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019

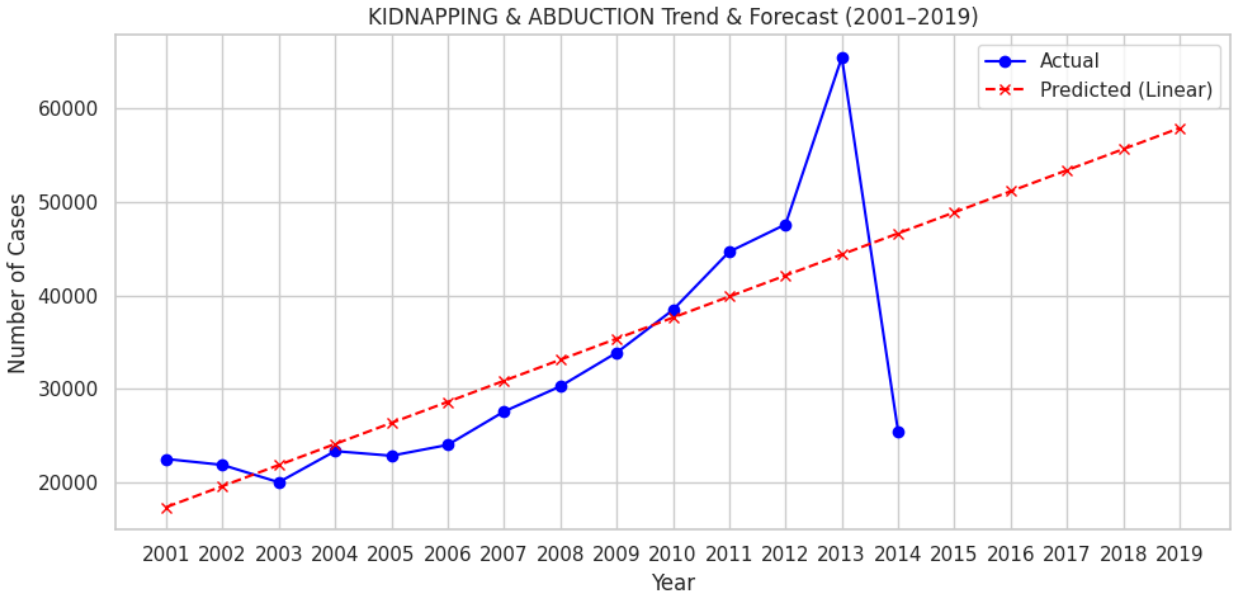

DACOITY Trend & Forecast (2001–2019)
Actual
Predicted (Linear)
Number of Cases
Year
6000
5500
5000
4500
4000
3500
2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019

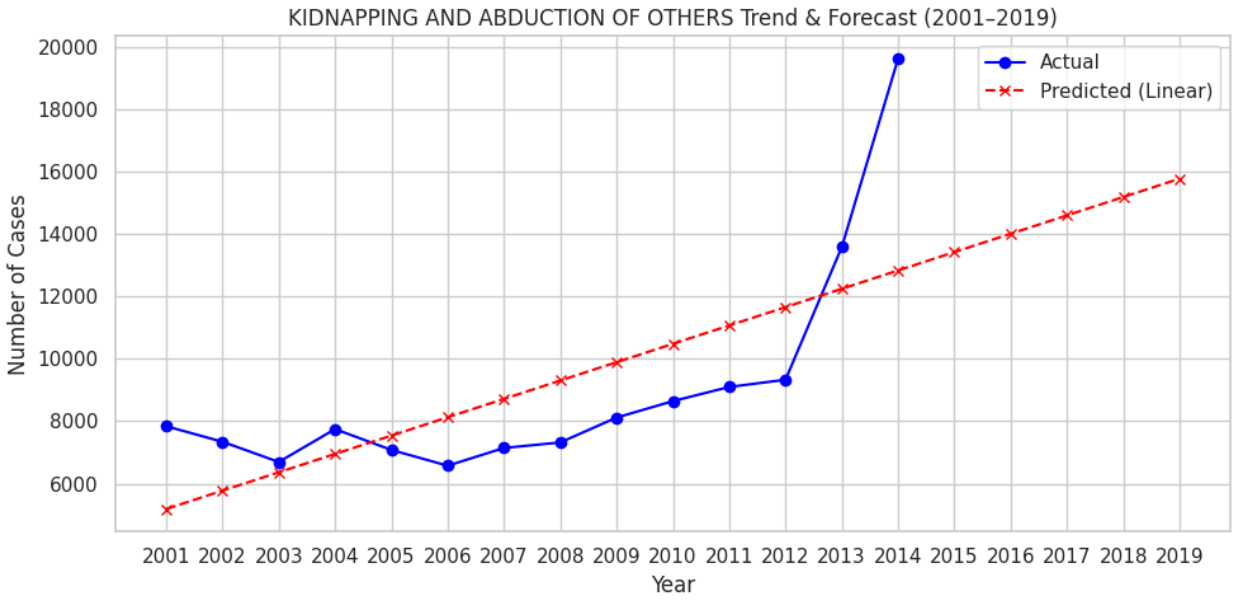

DOWRY DEATHS Trend & Forecast (2001–2019)
Actual
Predicted (Linear)
Number of Cases
9500
9000
8500
8000
7500
7000
6500
2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019
Year

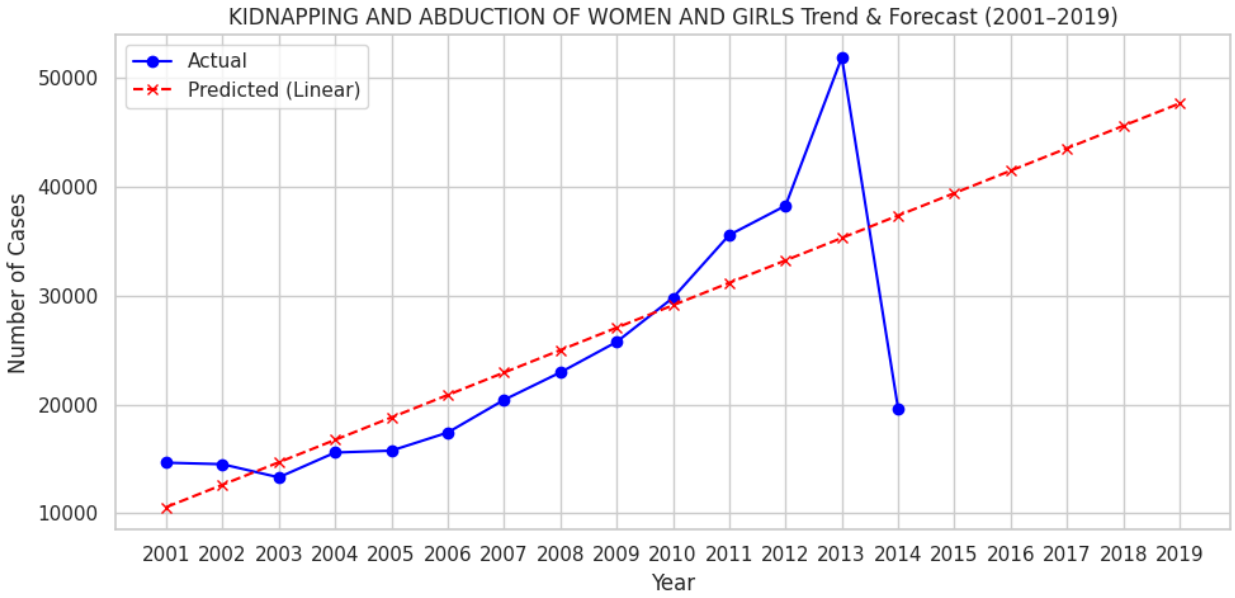

HURT/GREVIOUS HURT Trend & Forecast (2001–2019)
Actual
Predicted (Linear)
Number of Cases
300000
250000
200000
150000
100000
2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019
Year

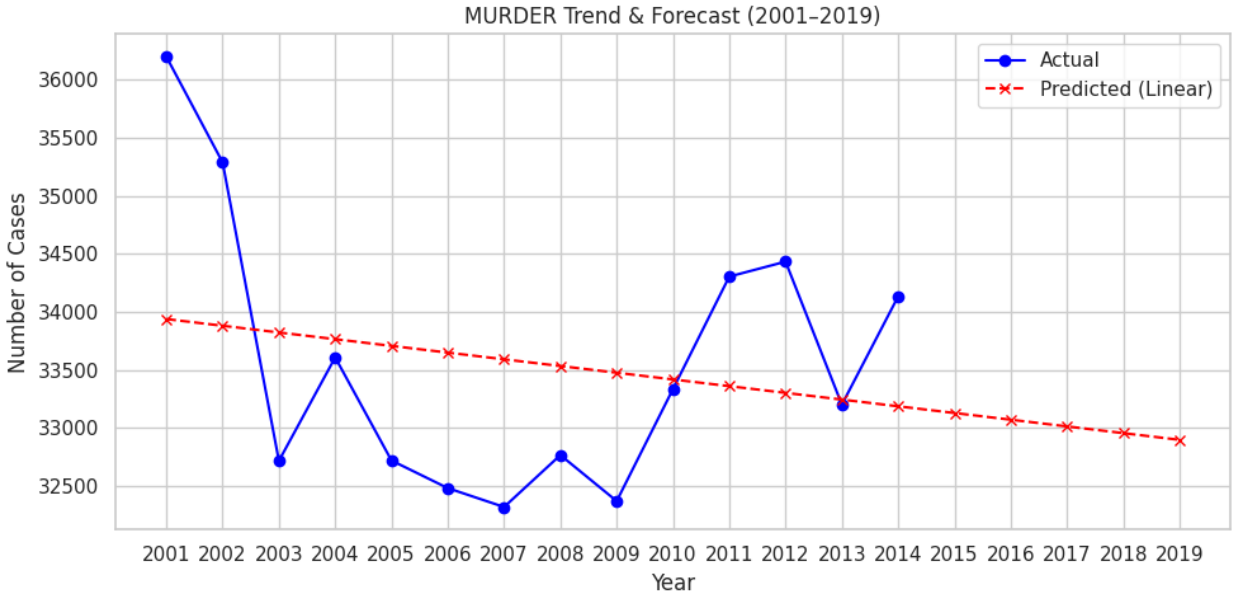

IMPORTATION OF GIRLS FROM FOREIGN COUNTRIES Trend & Forecast (2001–2019)
Actual
Predicted (Linear)
Number of Cases
Year
20
40
60
80
100
120
140
2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019

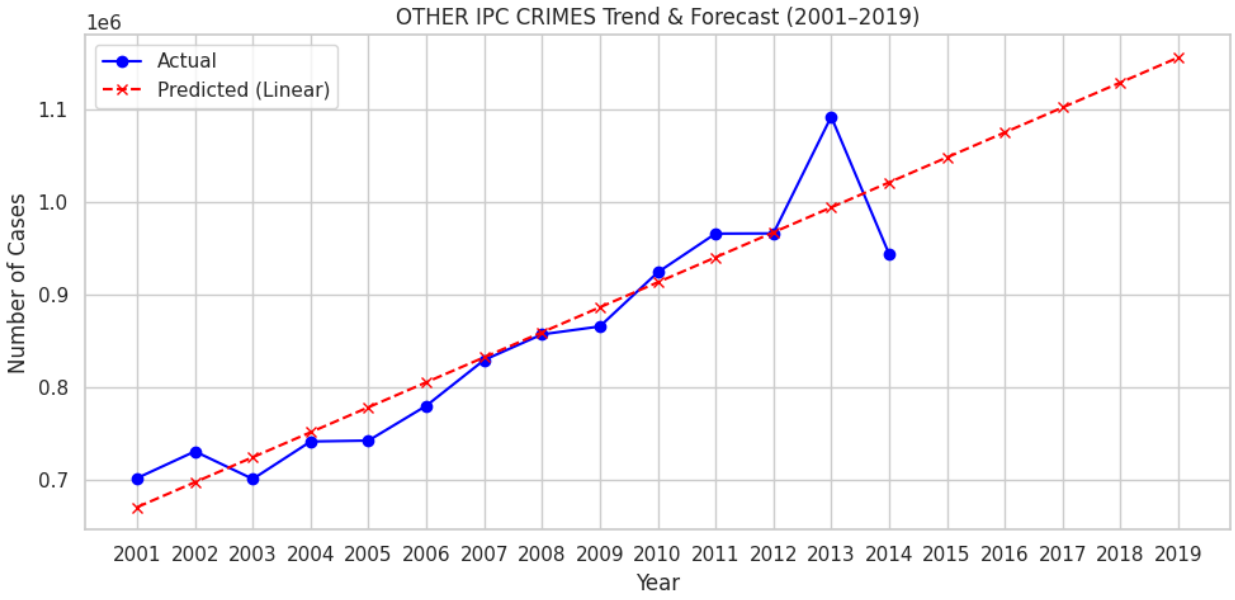

INSULT TO MODESTY OF WOMEN Trend & Forecast (2001–2019)
Actual
Predicted (Linear)
Number of Cases
Year
8500
9000
9500
10000
10500
11000
11500
12000
12500
2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019

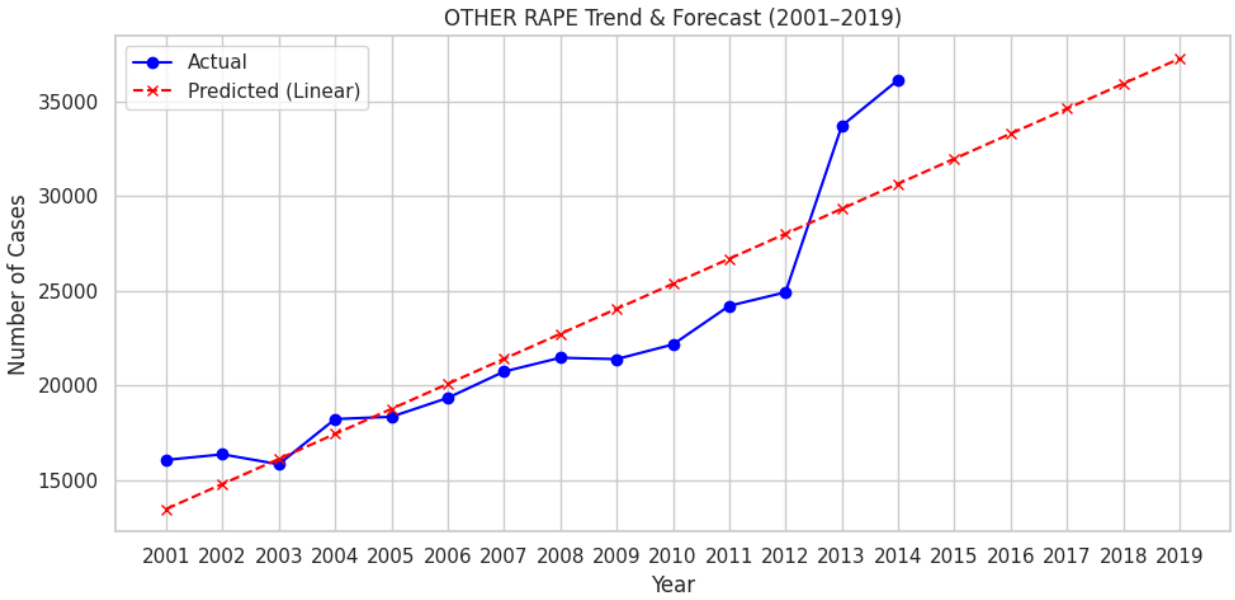

KIDNAPPING & ABDUCTION Trend & Forecast (2001–2019)
Actual
Predicted (Linear)
Number of Cases
60000
50000
40000
30000
20000
2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019
Year

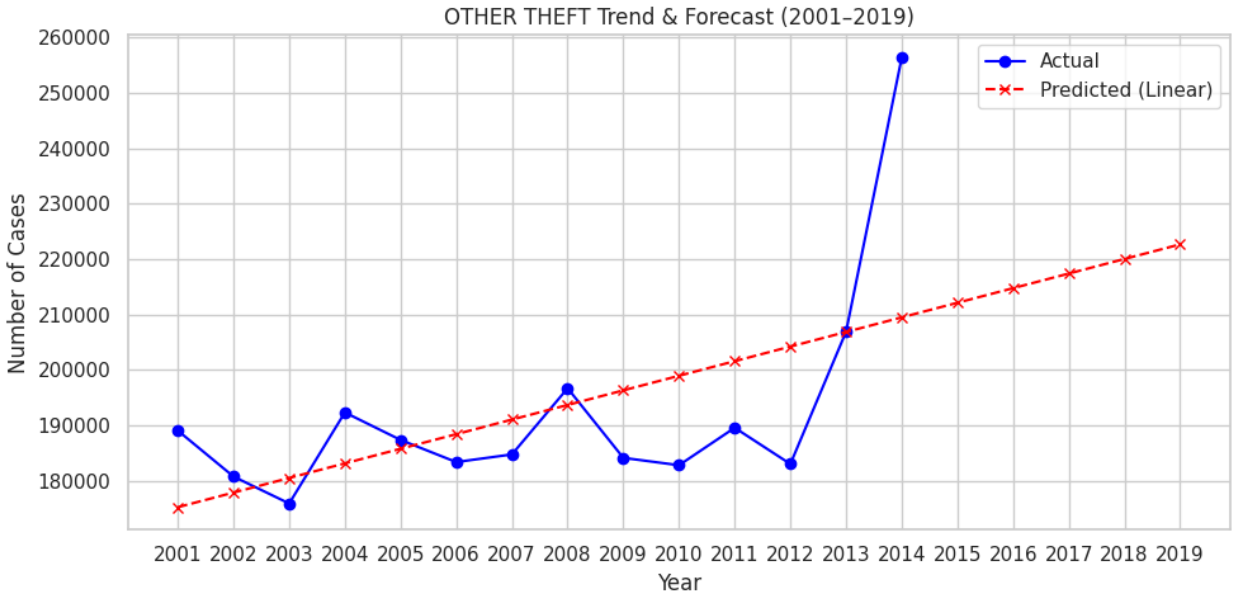

KIDNAPPING AND ABDUCTION OF OTHERS Trend & Forecast (2001–2019)
Actual
Predicted (Linear)
Number of Cases
20000
18000
16000
14000
12000
10000
8000
6000
2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019
Year

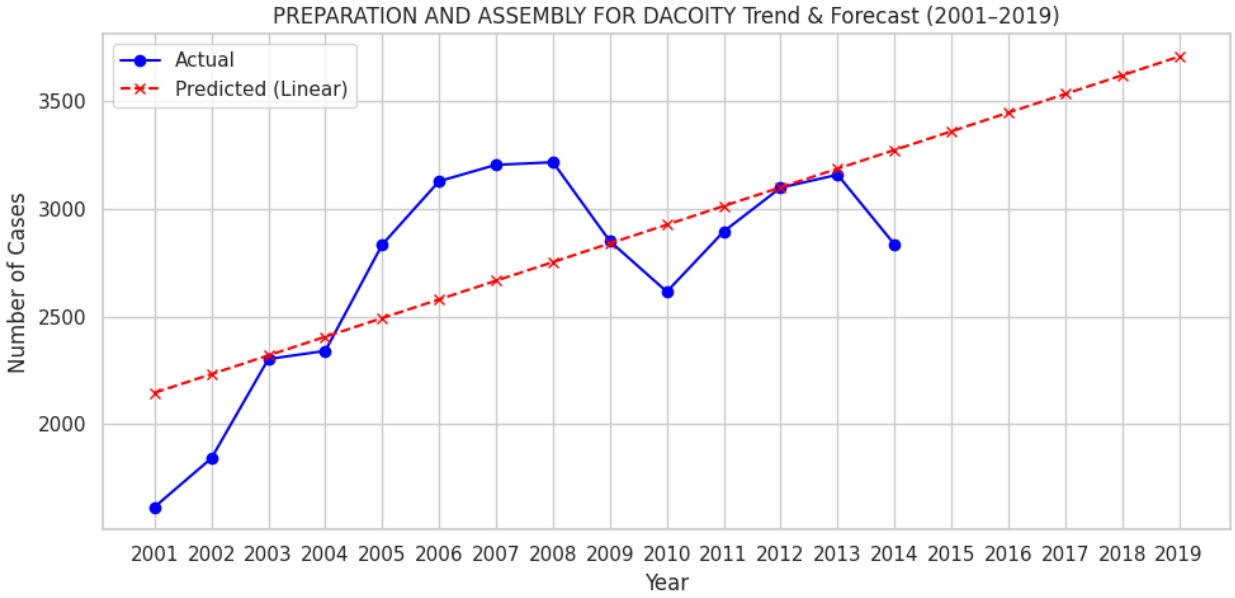

KIDNAPPING AND ABDUCTION OF WOMEN AND GIRLS Trend & Forecast (2001–2019)
Actual
Predicted (Linear)
Number of Cases
Year
10000
20000
30000
40000
50000
2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019

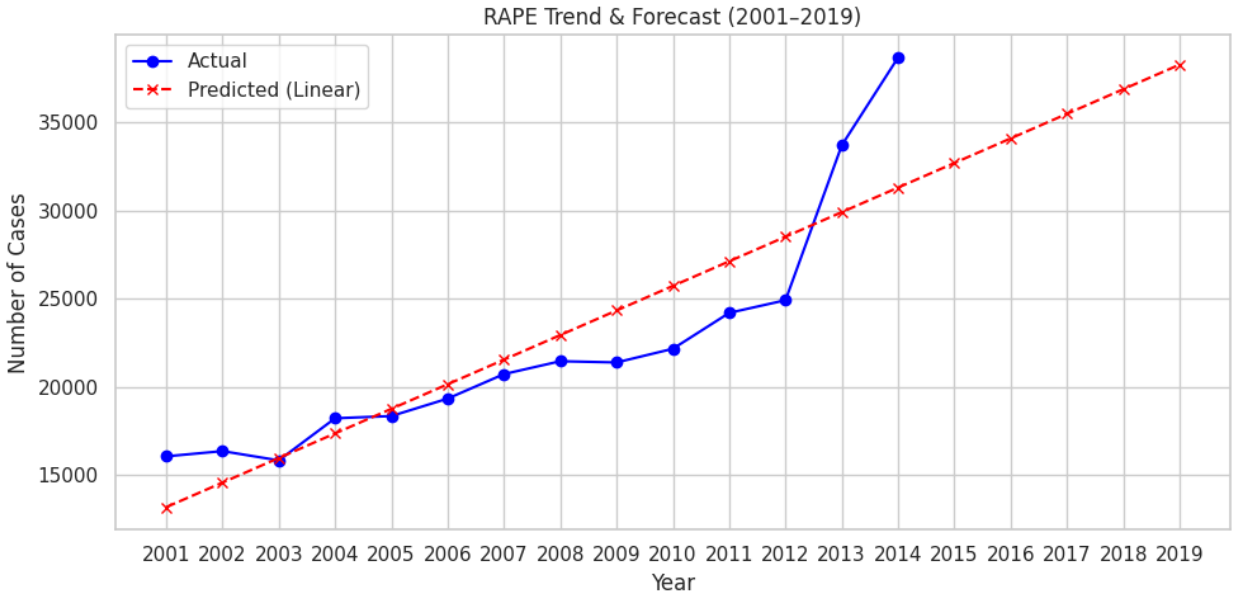

MURDER Trend & Forecast (2001–2019)
Actual
Predicted (Linear)
Number of Cases
Year
32500
33000
33500
34000
34500
35000
35500
36000
2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019

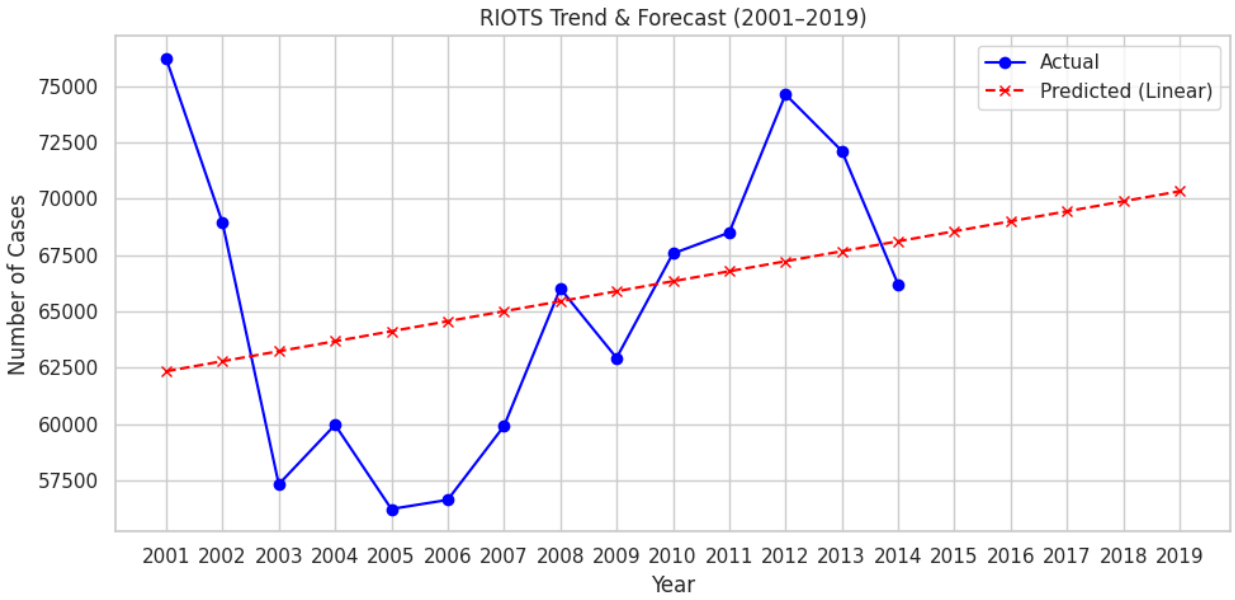

OTHER IPC CRIMES Trend & Forecast (2001–2019)
1e6
Actual
Predicted (Linear)
1.1
1.0
0.9
0.8
0.7
Number of Cases
2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019
Year

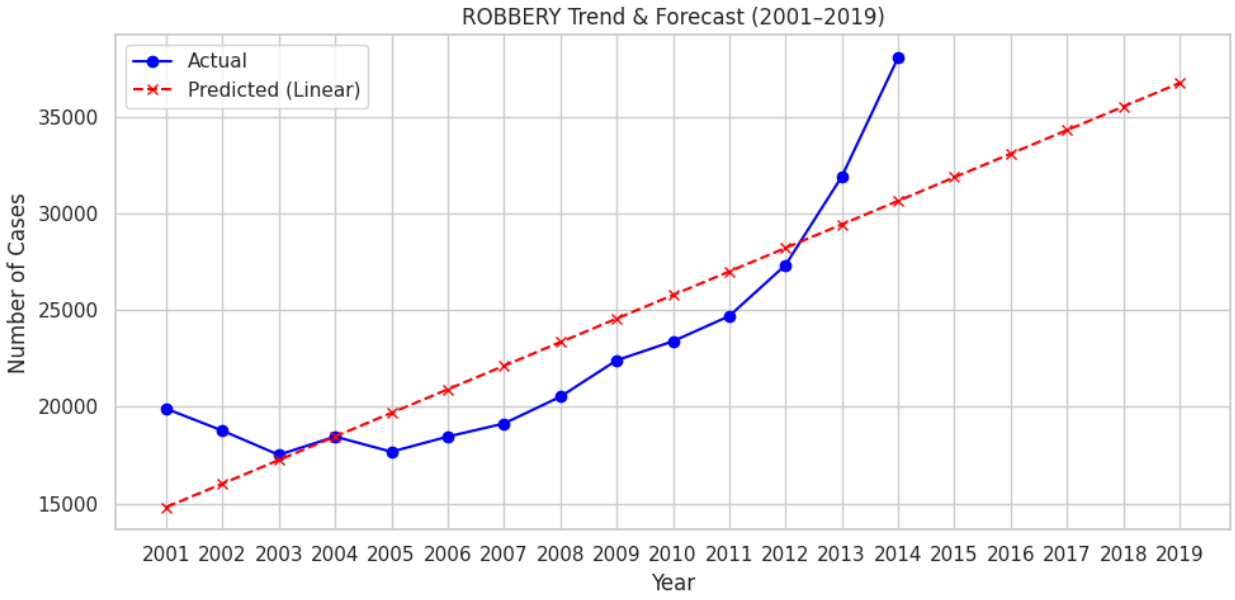

OTHER RAPE Trend & Forecast (2001–2019)
Actual
Predicted (Linear)
35000
30000
25000
20000
15000
Number of Cases
2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019
Year

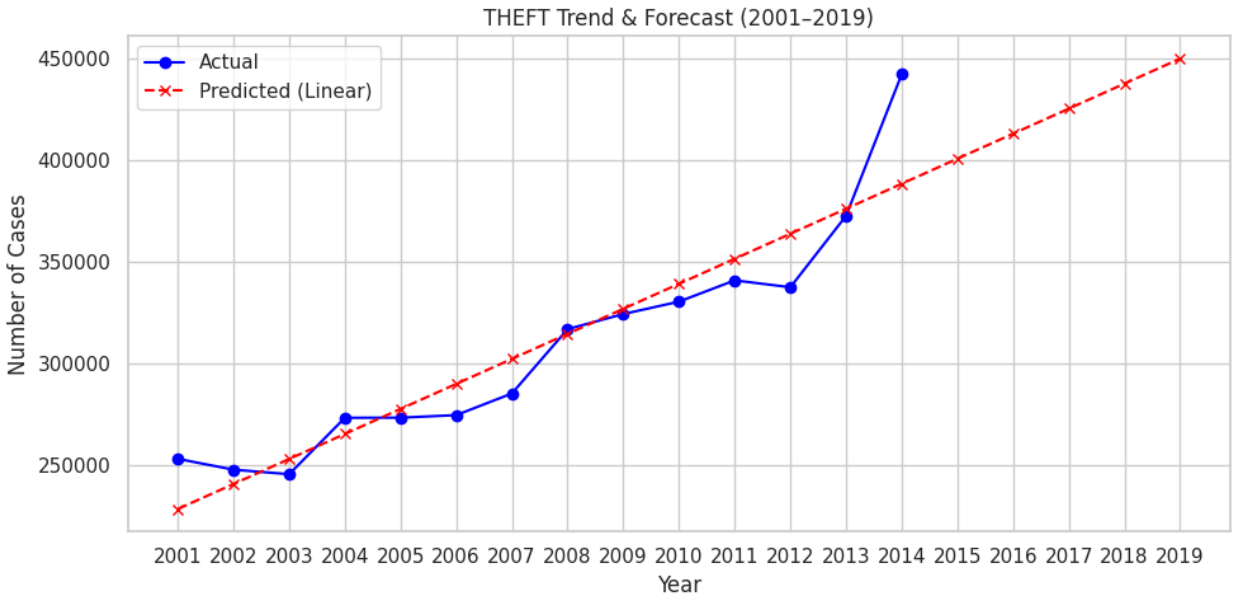

OTHER THEFT Trend & Forecast (2001–2019)
Actual
Predicted (Linear)
Number of Cases
Year
260000
250000
240000
230000
220000
210000
200000
190000
180000
2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019


PREPARATION AND ASSEMBLY FOR DACOITY Trend & Forecast (2001–2019)
Actual
Predicted (Linear)
Number of Cases
Year
3500
3000
2500
2000
2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019


RAPE Trend & Forecast (2001–2019)
Actual
Predicted (Linear)
Number of Cases
35000
30000
25000
20000
15000
2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019
Year

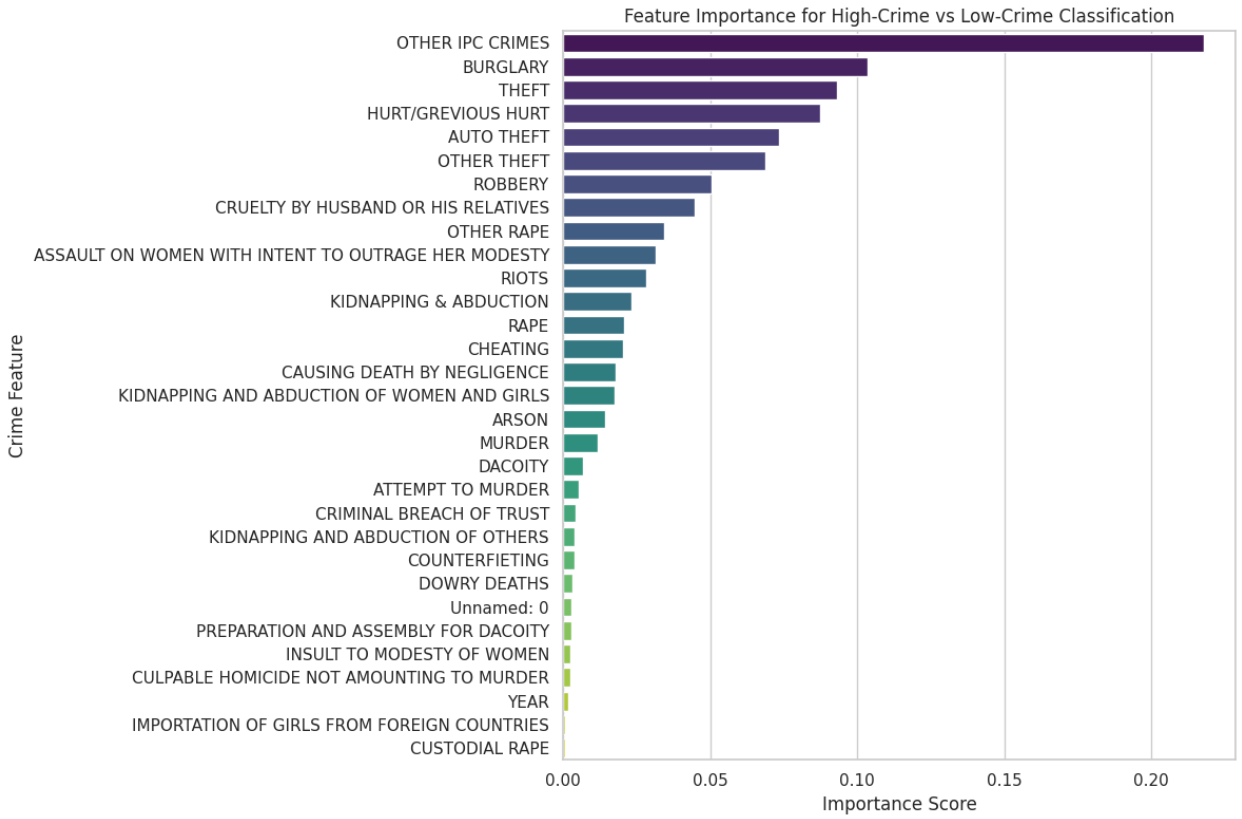

RIOTS Trend & Forecast (2001–2019)
Actual
Predicted (Linear)
Number of Cases
75000
72500
70000
67500
65000
62500
60000
57500
2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019
Year





- Use a machine learning model to classify high-crime and low-crime districts.

  We used Random Forest Classifier to classify districts Into either high crime or low crime, with the help of total IPC crimes. As shown below.



	DISTRICT	TOTAL IPC CRIMES	PREDICTED_CRIME_LABEL
0	24 PARGANAS NORTH	136204	High-Crime
1	24 PARGANAS SOUTH	136479	High-Crime
2	A and N ISLANDS	807	Low-Crime
3	ADILABAD	74376	High-Crime
4	AGAR	2861	Low-Crime
5	AGRA	96803	High-Crime
6	AHMEDABAD CITY	15286	Low-Crime
7	AHMEDABAD COMMR.	239263	High-Crime
8	AHMEDABAD RURAL	46015	High-Crime
9	AHMEDNAGAR	86653	High-Crime

- Develop a crime risk index for districts based on historical data.

  Below is a new calculated field where we calculate CRIME_RISK_INDEX using MinMaxScalar.

```
                    CRIME_RISK_INDEX
DISTRICT
BANGALORE COMMR.          16.236243
MUMBAI COMMR.             14.287922
PATNA                     11.606812
MURSHIDABAD               10.909849
HYDERABAD CITY            10.709770
24 PARGANAS NORTH          9.930747
LUCKNOW                    9.680983
KOLKATA                    9.554579
CYBERABAD                  9.457047
AHMEDABAD COMMR.           8.941472
PUNE COMMR.                8.818419
INDORE                     8.720915
24 PARGANAS SOUTH          8.503972
WEST                       8.347027
AGRA                       8.208015
NADIA                      8.082083
KANPUR NAGAR               7.926434
SOUTH                      7.886104
THANE COMMR.               7.805595
MUZAFFARPUR                7.599625
<ipython-input-37-8a881a361ce9>:8: FutureWarning:

errors='ignore' is deprecated and will raise in a future version. Use to_numeric without passing `errors` and catch exceptions explicitly instead
```

Bonus Questions

- What percentage of crimes are committed against women?

  Crimes committed against women is 9.96%

```python
import pandas as pd

# Load dataset
df = pd.read_csv("Districtwise_Crime_of_India_2001_to_2014 - Sheet1.csv")
df.columns = df.columns.str.strip()
df = df[~df['DISTRICT'].str.upper().str.contains('TOTAL')]
df = df.apply(pd.to_numeric, errors='ignore')

# List of crimes against women
crimes_against_women = [
    'RAPE',
    'CUSTODIAL RAPE',
    'OTHER RAPE',
    'DOWRY DEATHS',
    'ASSAULT ON WOMEN WITH INTENT TO OUTRAGE HER MODESTY',
    'INSULT TO MODESTY OF WOMEN',
    'CRUELTY BY HUSBAND OR HIS RELATIVES',
    'KIDNAPPING AND ABDUCTION OF WOMEN AND GIRLS'
]

# Ensure numeric conversion
df[crimes_against_women] = df[crimes_against_women].apply(pd.to_numeric, errors='coerce')
df['TOTAL IPC CRIMES'] = pd.to_numeric(df['TOTAL IPC CRIMES'], errors='coerce')

# Total crimes against women
total_women_crimes = df[crimes_against_women].sum().sum()
total_ipc_crimes = df['TOTAL IPC CRIMES'].sum()

# Percentage calculation
percentage_women_crimes = (total_women_crimes / total_ipc_crimes) * 100

print(f"Percentage of crimes committed against women: {percentage_women_crimes:.2f}%")
```

```
Percentage of crimes committed against women: 9.96%
```

- Identify the state with the highest number of dowry deaths.

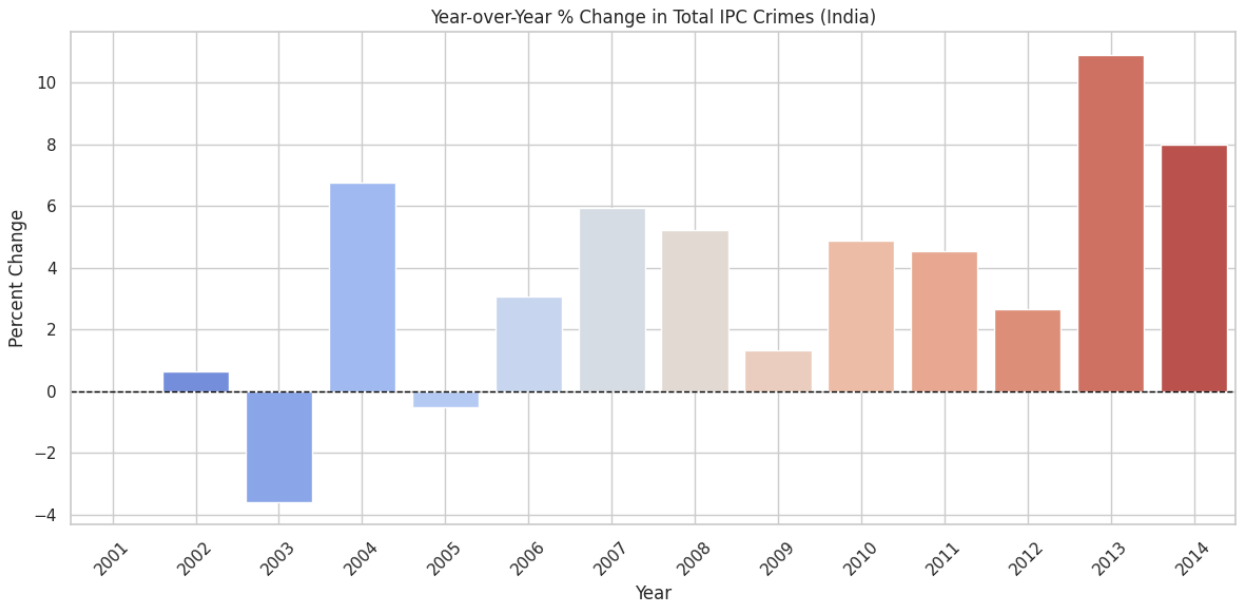
  Highest number of dowry deaths was in UTTAR PRADESH.

```python
import pandas as pd

# Load dataset
df = pd.read_csv("Districtwise_Crime_of_India_2001_to_2014 - Sheet1.csv")
df.columns = df.columns.str.strip()

# Filter out TOTAL rows
df = df[~df['DISTRICT'].str.upper().str.contains('TOTAL')]

# Ensure 'DOWRY DEATHS' is numeric
df['DOWRY DEATHS'] = pd.to_numeric(df['DOWRY DEATHS'], errors='coerce')

# Group by state and sum dowry deaths
dowry_deaths_by_state = df.groupby('STATE/UT')['DOWRY DEATHS'].sum()

# Find state with highest number
max_state = dowry_deaths_by_state.idxmax()
max_value = dowry_deaths_by_state.max()

print(f"State with highest dowry deaths: {max_state} ({max_value} cases)")
```

```
State with highest dowry deaths: UTTAR PRADESH (28628 cases)
```

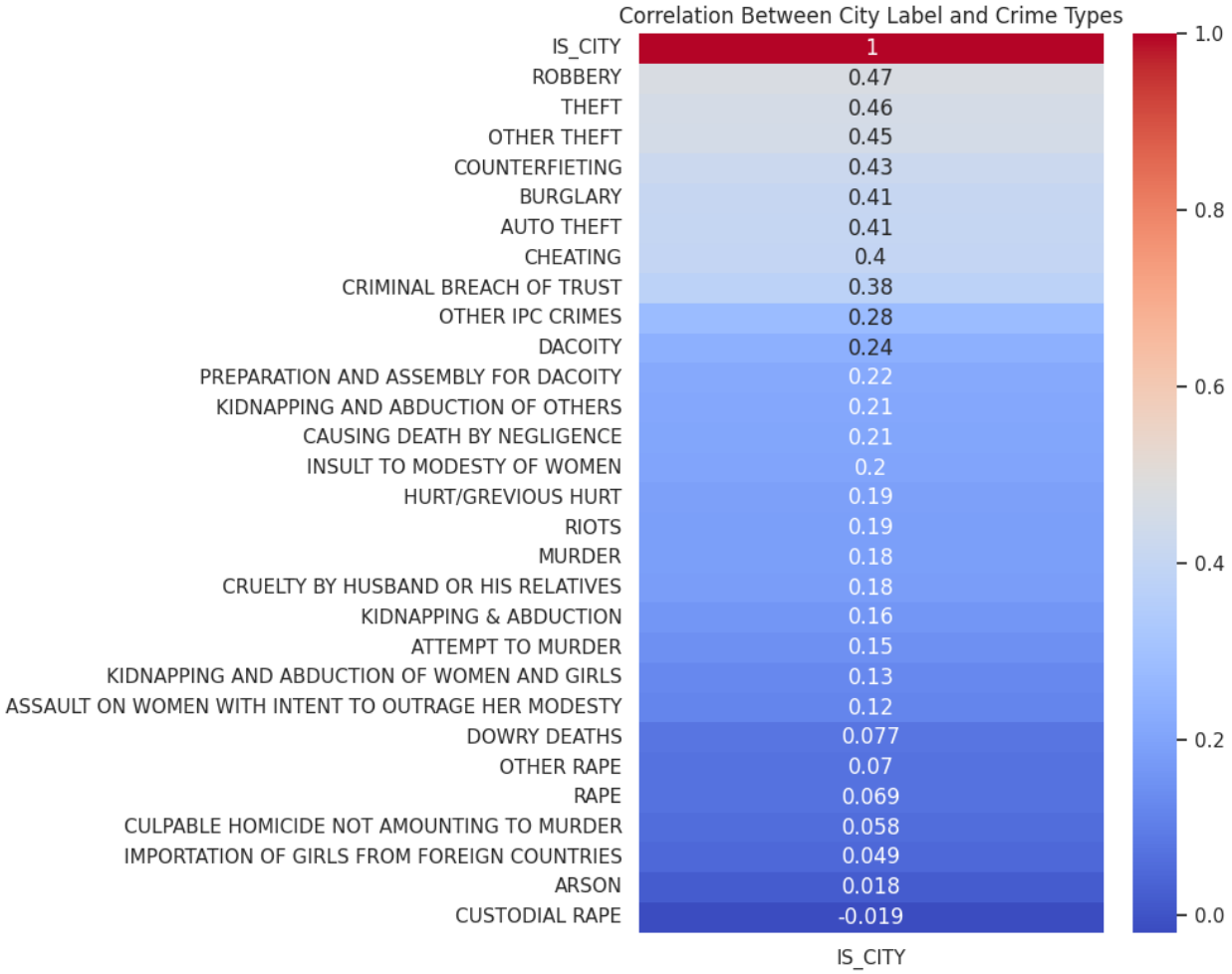
- Analyze seasonal variations in crime trends (e.g., do crimes increase during certain months?).

  This shows the year over year seasonality as we did not have months data in dataset.



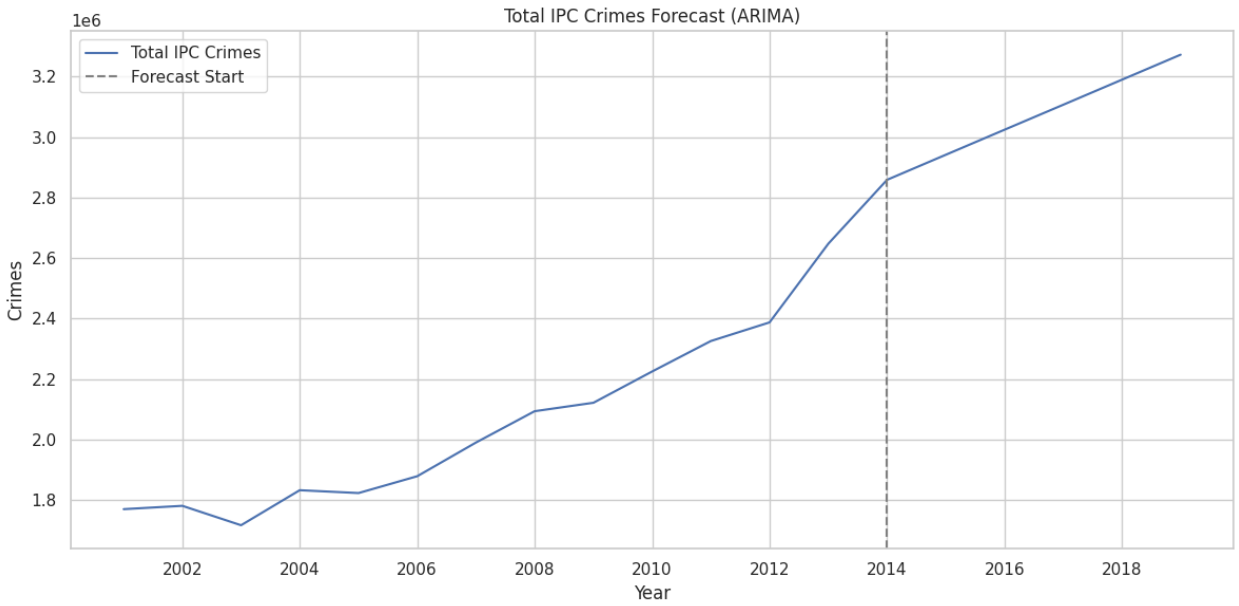
- Examine if there is a link between cities and crime rates.

  Below is correlation matrix which shows the link between cities and crime.



- Build a time-series model to forecast crime rates for the next five years.

  We have used ARIMA to forecast 5 years data.




```
2015    2941755
2016    3024569
2017    3107382
2018    3190193
2019    3273002
Freq: Y-DEC, Name: predicted_mean, dtype: int64
```