

11-11-2022

CAT-3 PROJECT

AN ANALYSIS OF CRIME AGAINST WOMEN IN INDIA

TEAM MEMBERS:

2031015 GOKULA KRISHNAN S

2031056 YUVANIKA S V

PROBLEM DEFINITION:

An analysis of crimes against women in India from 2001 to 2012 is taken and visualized. Then the crime rate is predicted until 2022 by using ARIMA and SARIMA models. Then the result of these two models are compared and the model that gives better prediction is considered the best.

FUNCTIONAL REQUIREMENTS:

1. Data collection
2. Data processing
3. Training and Testing
4. Modeling
5. Predicting

SYSTEM COMPONENTS(MODULES):

1. Import Libraries:

In this module, different libraries like pandas, NumPy, matplotlib, and statsmodels are imported, which are useful for data processing, visualization, accuracy, and prediction.

2. Data Processing:

In this module, the data undergoes pre-processing like cleaning the dataset and removing null and unwanted values.

3. Visualization:

- Initially, the data undergoes visualization for the crime given by the user.
- Next for the entered year, visualization is done for all the crime counts for all the states in India.

- Next the crime rate is compared for the years 2001 and 2012 (i.e for the initial and final data in the data set).
- Then the graph is plotted against the crime count district-wise for the entered state.

4. Built ARIMA model

- An autoregressive integrated moving average model, a statistical analysis model that uses time series data to better understand the data set or predict future trends.
- Here the values in the data set range from 2001 to 2012.
- Here the data is fed from 2013 to 2022 manually.
- Then the model is built and the graph is displayed.

5. Built SARIMA model

- A seasonal autoregressive integrated moving average model which is like ARIMA but more powerful.
- We can use statsmodels implementation of SARIMA.
- Then the model is built and the graph is displayed using a seasonal pattern.

6. Streamlit:

- Here Streamlit is used as GUI.
- The user can select a particular state and crime and the graphs of ARIMA and SARIMA models are displayed with their predicted values.

SOURCE CODE:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import warnings
warning.filterwarnings('ignore')
df = pd.read_excel('crime.xlsx')
c=0
for i in range(len(df)):
    if df["STATE/UT"][i]=='All India':
        c=c+1
print("All India crimes : ",c)
```

All India crimes : 8

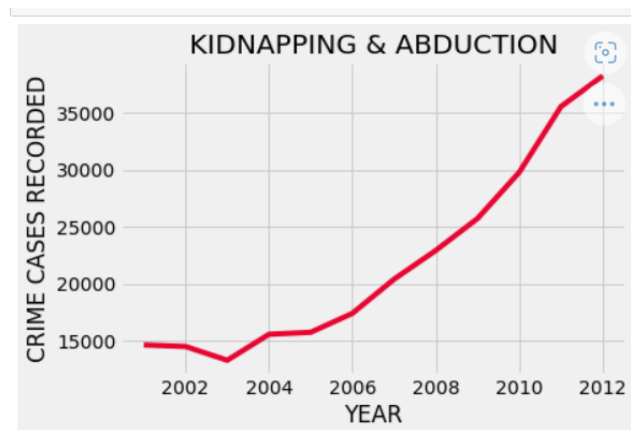
```

df.set_index("STATE/UT", inplace = True)
df=df.drop("All India")
crime=[]
head=df["CRIME HEAD"].unique()
#print("Crime heads",head)
k=0
for i in head:
    crime.append(i)
    print(k,i)
    k+=1
#print(crime)
cr=int(input("\nEnter crime name from given menu(0-7) : "))
print("\nSelected Crime : ",crime[cr])
0 RAPE
1 KIDNAPPING & ABDUCTION
2 DOWRY DEATH
3 ASSAULT ON WOMEN WITH INTENT TO OUTRAGE HER MODESTY
4 INSULT TO THE MODESTY OF WOMEN
5 CRUELTY BY HUSBAND OR RELATIVES
6 IMMORAL TRAFFIC(PREVENTION)ACT
7 INDECENT REPRESENTATION OF WOMEN(PREVENTION)ACT
Enter crime name from given menu(0-7) : 1
Selected Crime : KIDNAPPING & ABDUCTION
cr1=[]
b=0
a=0
for z in range(2001,2013):
    for i in range(len(df)):
        if df["CRIME HEAD"][i]==crime[cr]:
            a=df[z][i]
            b+=a
    cr1.append(b)
    b=0
yr=[feat for feat in df]
yr.remove("CRIME HEAD")
print(f'List of Total " {crime[cr]} " case registered:\n')
print(yr)
print(cr1)
List of Total " KIDNAPPING & ABDUCTION " case registered:
[2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012]
[14645, 14506, 13296, 15578, 15750, 17414, 20416, 22939, 25741, 29795, 35565, 38262]
yeardf =pd.DataFrame(yr)
crimedf =pd.DataFrame(cr1)
CrimeTable = pd.concat([yeardf,crimedf],axis=1)
CrimeTable.columns=["Year", "Total Crime Recorded"]
CrimeTable

```

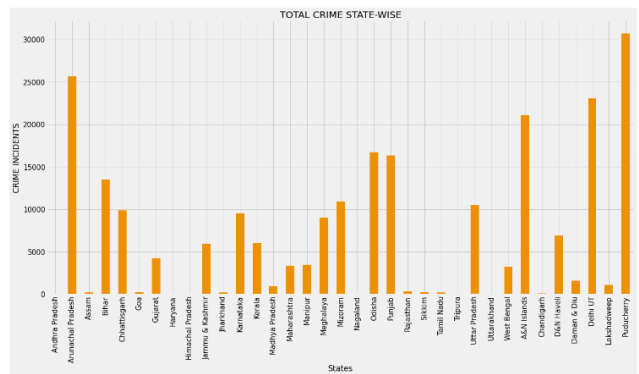
| | Year | Total Crime Recorded |
|----|------|----------------------|
| 0 | 2001 | 14645 |
| 1 | 2002 | 14506 |
| 2 | 2003 | 13296 |
| 3 | 2004 | 15578 |
| 4 | 2005 | 15750 |
| 5 | 2006 | 17414 |
| 6 | 2007 | 20416 |
| 7 | 2008 | 22939 |
| 8 | 2009 | 25741 |
| 9 | 2010 | 29795 |
| 10 | 2011 | 35565 |
| 11 | 2012 | 38262 |

```
plt.style.use('fivethirtyeight')
plt.plot(yr, cr1, color='#e70631')
plt.xlabel('YEAR')
plt.ylabel('CRIME CASES RECORDED')
plt.title(crime[cr])
plt.show()
```

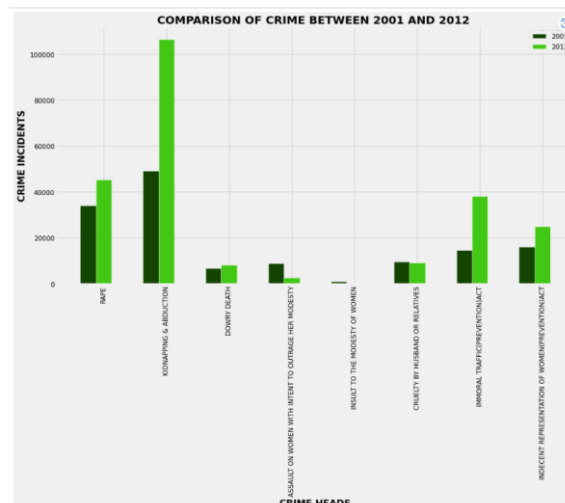


```
df=df.reset_index()
states=df["STATE/UT"].unique()
states=list(states)
stateG=df.groupby(["STATE/UT"])
states1 =pd.DataFrame(states)
totalC=[]
year=int(input("Enter year: "))
s1=stateG.agg({year:"sum"})
for i in range(len(s1)):
    val=s1[year][i]
    totalC.append(val)
Enter year: 2012
totalC1 =pd.DataFrame(totalC)
tc = pd.concat([states1,totalC1],axis=1)
tc.columns=["States","Total Crime"]
tc=tc.set_index("States")
plt.style.use('fivethirtyeight')
plt.title('TOTAL CRIME STATE-WISE')
```

```
plt.xlabel('STATE')
plt.ylabel('CRIME INCIDENTS')
tc["Total Crime"].plot(kind='bar',figsize=(20,10),fontsize=15, color='#ec9104')
```



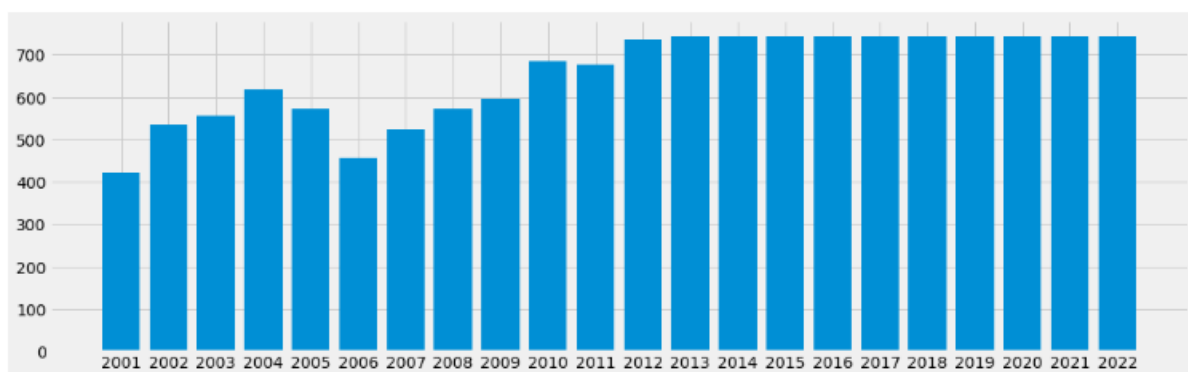
```
crime=list(crime)
Cgroup=df.groupby("CRIME HEAD")
Cgroup.sum()
Cgroup1=Cgroup.agg({'2001':'sum','2012':'sum'})
plt.style.use('fivethirtyeight')
barWidth = 0.25
bars1 = []
bars2 = []
for i in range(len(Cgroup1)):
    bars1.append(Cgroup1[2001][i])
for i in range(len(Cgroup1)):
    bars2.append(Cgroup1[2012][i])
plt.figure(figsize=(15,8),)
r1 = np.arange(len(bars1))
r2 = [x + barWidth for x in r1]
plt.title('COMPARISON OF CRIME BETWEEN 2001 AND 2012 ',fontweight='bold')
plt.bar(r1, bars1, color='#134402', width=barWidth, edgecolor='white', label='2001')
plt.bar(r2, bars2, color='#42c714', width=barWidth, edgecolor='white', label='2012')
plt.ylabel('CRIME INCIDENTS',fontweight='bold')
plt.yticks(fontsize=12)
plt.xlabel('CRIME HEADS', fontweight='bold')
plt.xticks([r + barWidth for r in range(len(bars1))],crime,rotation="vertical",fontsize=12)
plt.legend(fontsize=12)
plt.show()
```



```

import statsmodels.api as sm
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.tsa.arima.model import ARIMA
names = ['2001', '2002', '2003', '2004', '2005', '2006', '2007', '2008', '2009', '2010', '2011', '2012', '2013', '2014', '2015', '2016', '2017', '2018', '2019', '2020', '2021', '2022']
def state_case(state, case):
    for i in range(0, len(df)):
        if df.iloc[i,0] == state and df.iloc[i,1]==case:
            temp = df.iloc[i, 2:]
            train = np.array(temp)
            train = train.astype(np.int64)
            train = np.reshape(train, (-1, 1))
    temp = pd.DataFrame(train)
    sm.graphics.tsa.plot_acf(temp.values.squeeze())
    sm.graphics.tsa.plot_pacf(temp.values.squeeze(),lags=5)
    model = ARIMA(train, order=(1,1,1))
    model_fit = model.fit()
    pred = model_fit.predict(start=13, end=22)
    new_data = np.append(train, pred)
    plt.figure(figsize=(16,5))
    plt.bar(names, new_data)
    year = [2013, 2014, 2015, 2016,2017,2018,2019,2020,2021,2022]
    for w in range(0, 10):
        print(year[w], " ",pred[w].round(0))
    return pred
print("Enter the state:")
s=input()
print("Enter the crime")
c=input()
pred = state_case(s, c)
Enter the state:
Tamil Nadu
Enter the crime
RAPE

```



```

from statsmodels.tsa.statespace.sarimax import SARIMAX

names = ['2001', '2002', '2003', '2004', '2005', '2006', '2007', '2008', '2009', '2010', '2011', '2012', '2013', '2014', '2015', '2016', '2017', '2018', '2019', '2020', '2021', '2022']

def state_case(state, case):

```

```

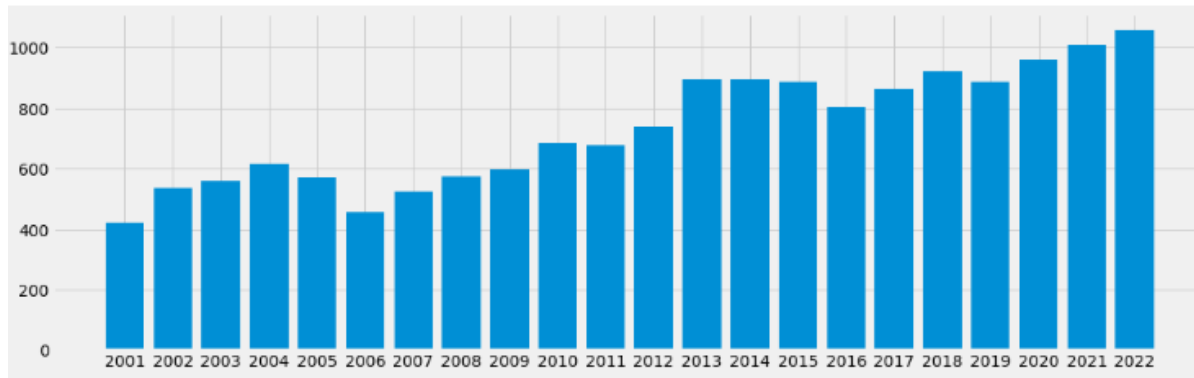
for i in range(0, len(df)):
    if df.iloc[i,0] == state and df.iloc[i,1]==case:
        temp = df.iloc[i, 2:]
        train = np.array(temp)
        train = train.astype(np.int64)
        train = np.reshape(train, (-1, 1))
    temp = pd.DataFrame(train)
    sm.graphics.tsa.plot_acf(temp.values.squeeze(),lags=10)
    sm.graphics.tsa.plot_pacf(temp.values.squeeze(),lags=5)
    model = SARIMAX(train, order=(12,1,1))
    model_fit = model.fit()
    pred = model_fit.predict(start=13, end=22)
    model_fit.summary()
    new_data = np.append(train, pred)
    plt.figure(figsize=(16,5))
    plt.bar(names, new_data)
    year = [2013, 2014, 2015, 2016,2017,2018,2019,2020,2021,2022]
    for w in range(0, 10):
        print(year[w]," ",pred[w].round(0))
    return pred
print("Enter the state:")
s=input()
print("Enter the crime")
c=input()
pred = state_case(s, c)

```

```

Enter the state:
Tamil Nadu
Enter the crime
RAPE

```



OUTPUT:

Time Series

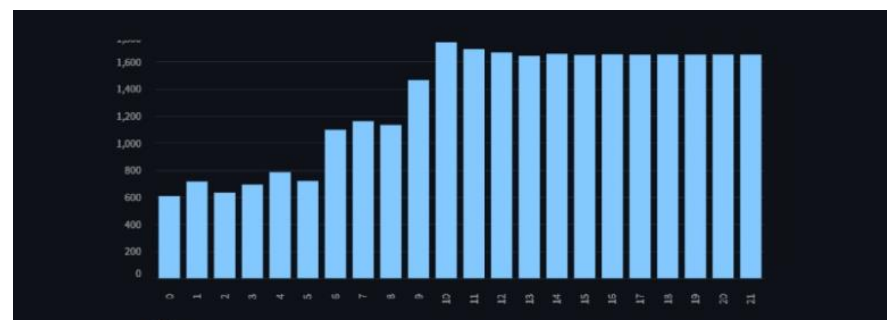
Select State Name

Tamil Nadu

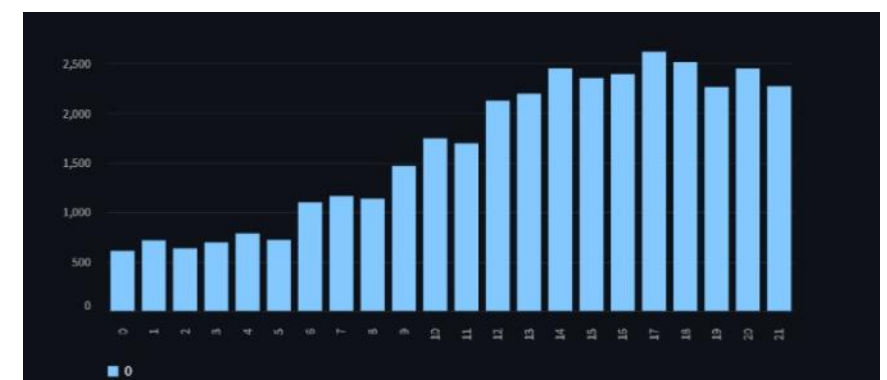
Select Crime Name

KIDNAPPING & ABDUCTION

ARIMA TIME SERIES



SARIMAX TIME SERIES



INFERENCE:

The SARIMA model best predicts the data than the ARIMA model.