CAS ACTIVITY REPORT

Name: Ganesh Laxman Phadtare

PRN No.: 2267571242131

Std: T.Y. B.Tech **Div: B**

Roll No.: 61

TASK 1
METHODS TO FIND AUTHENTIC DATA FOR MACHINE LEARNING

Finding reliable and research-backed datasets is a critical step in building accurate machine learning models. For the **Crime Classification using Logistic Regression project**, the following methods were used:

1. Academic Research Platforms

- Platforms like DELNET, Google Scholar, and IEEE Xplore were used to explore peer-reviewed journals and academic papers.
- Keywords such as "Crime prediction using machine learning" and "District-level IPC crime data India" helped identify studies that reference real datasets.
- Methodology sections of papers often revealed sources like data.gov.in or NCRB (National Crime Records Bureau) reports.

2. Public Dataset Repositories

- Platforms like Kaggle, OpenML, and India's Open Government Data (OGD) portal were used.
- The final dataset was downloaded from data.gov.in, which hosts official district-wise IPC crime data.
- Dataset included information on States/UTs, District, Year, and total IPC crimes — ideal for binary classification.

Step 1: Go to the DELNET Website

- I visited the official DELNET portal: https://delnet.in.
- DELNET (Developing Library Network) is a trusted academic platform that provides access to a wide range of scholarly resources.
- I logged in using the credentials provided by my educational institution, which gave me full access to the e-resources.

Step 2: Search for the Topic

- In the DELNET search bar, I entered the query:
  "Loan Approval prediction using machine learning"
- This helped me find studies where machine learning algorithms were applied to predict or diagnose diabetes.
- I applied filters to limit results to peer-reviewed journals, conference proceedings, and academic books, which are more likely to contain experimental research with datasets.

Step 3: Select and Read a Relevant Research Paper

- From the search results, I chose a research paper titled:
  " BANK LOAN PREDICTION USING MACHINE LEARNING TECHNIQUES "
- The title clearly indicated the use of supervised learning techniques for medical prediction tasks, making it relevant to my objective.
- I selected this paper because research papers with such titles often include detailed information about the dataset and methodology used.

Step 4: Access the Full Text of the Paper

- I clicked on the “Full Text” or “View PDF” link to access the complete research paper.
- Reading the full paper is essential because abstracts rarely contain dataset sources or technical details.
- The full text provided insight into the experimental process, the model used, and the dataset characteristics.

Step 5: Locate the Dataset Information

- While reading the Methodology or Experimental Setup section of the paper, I found a detailed reference to the dataset used in the study.
- The paper described key aspects of the dataset, such as:
  - Data source (e.g., loan approval records or online repository)
  - Number of records and features
  - Type of variables (e.g., age, monthly income , debt ratio)

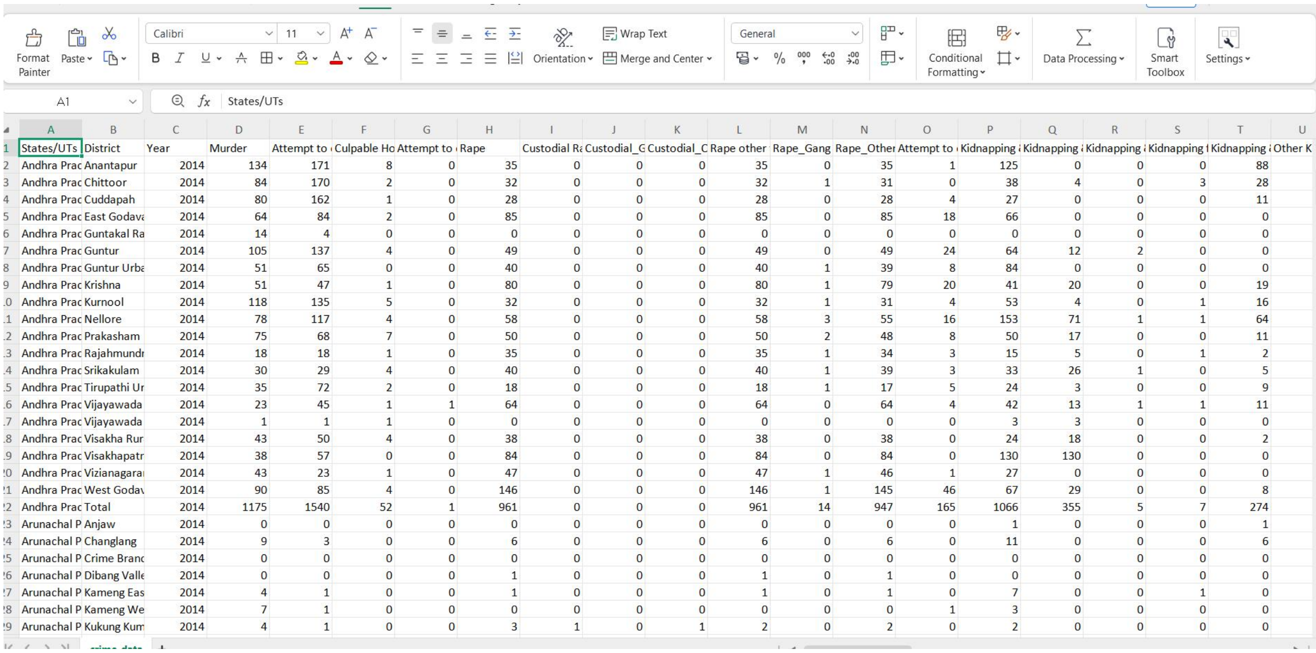
TASK 2

DATASET OVERVIEW

Dataset Name: District-wise Cognizable IPC Crimes
Source: https://data.gov.in
Total Records: ~720 rows (district-wise, year-wise data)
Total Features: 10+ columns

Feature Descriptions:

- States/UTs: Name of the state or union territory
- District: District name
- Year: Year of record
- Total Cognizable IPC crimes: Number of crimes recorded
- Additional columns: Various IPC sections (e.g., Murder, Robbery, Theft)



States/UTs

States/UTs	District	Year	Murder	Attempt to	Culpable Ho	Attempt to	Rape	Custodial Ra	Custodial_G	Custodial_C	Rape other	Rape_Gang	Rape_Other	Attempt to	Kidnapping	Kidnapping	Kidnapping	Kidnapping	Kidnapping	Other K
Andhra Prac	Anantapur	2014	134	171	8	0	35	0	0	0	35	0	35	1	125	0	0	0	88	
Andhra Prac	Chittoor	2014	84	170	2	0	32	0	0	0	32	1	31	0	38	4	0	3	28	
Andhra Prac	Cuddapah	2014	80	162	1	0	28	0	0	0	28	0	28	4	27	0	0	0	11	
Andhra Prac	East Godava	2014	64	84	2	0	85	0	0	0	85	0	85	18	66	0	0	0	0	
Andhra Prac	Guntakal Ra	2014	14	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Andhra Prac	Guntur	2014	105	137	4	0	49	0	0	0	49	0	49	24	64	12	2	0	0	
Andhra Prac	Guntur Urba	2014	51	65	0	0	40	0	0	0	40	1	39	8	84	0	0	0	0	
Andhra Prac	Krishna	2014	51	47	1	0	80	0	0	0	80	1	79	20	41	20	0	0	19	
Andhra Prac	Kurnool	2014	118	135	5	0	32	0	0	0	32	1	31	4	53	4	0	1	16	
Andhra Prac	Nellore	2014	78	117	4	0	58	0	0	0	58	3	55	16	153	71	1	1	64	
Andhra Prac	Prakasham	2014	75	68	7	0	50	0	0	0	50	2	48	8	50	17	0	0	11	
Andhra Prac	Rajahmundr	2014	18	18	1	0	35	0	0	0	35	1	34	3	15	5	0	1	2	
Andhra Prac	Srikakulam	2014	30	29	4	0	40	0	0	0	40	1	39	3	33	26	1	0	5	
Andhra Prac	Tirupathi Ur	2014	35	72	2	0	18	0	0	0	18	1	17	5	24	3	0	0	9	
Andhra Prac	Vijayawada	2014	23	45	1	1	64	0	0	0	64	0	64	4	42	13	1	1	11	
Andhra Prac	Vijayawada	2014	1	1	1	0	0	0	0	0	0	0	0	0	3	3	0	0	0	
Andhra Prac	Visakha Rur	2014	43	50	4	0	38	0	0	0	38	0	38	0	24	18	0	0	2	
Andhra Prac	Visakhapatr	2014	38	57	0	0	84	0	0	0	84	0	84	0	130	130	0	0	0	
Andhra Prac	Vizianagara	2014	43	23	1	0	47	0	0	0	47	1	46	1	27	0	0	0	0	
Andhra Prac	West Godav	2014	90	85	4	0	146	0	0	0	146	1	145	46	67	29	0	0	8	
Andhra Prac	Total	2014	1175	1540	52	1	961	0	0	0	961	14	947	165	1066	355	5	7	274	
Arunachal P	Anjaw	2014	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	
Arunachal P	Changlang	2014	9	3	0	0	6	0	0	0	6	0	6	0	11	0	0	0	6	
Arunachal P	Crime Branc	2014	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Arunachal P	Dibang Valle	2014	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0	
Arunachal P	Kameng Eas	2014	4	1	0	0	1	0	0	0	1	0	1	0	7	0	0	1	0	
Arunachal P	Kameng We	2014	7	1	0	0	0	0	0	0	0	0	0	1	3	0	0	0	0	
Arunachal P	Kukung Kum	2014	4	1	0	0	3	1	0	1	2	0	2	0	2	0	0	0	0	

Target Variable:

- Binary Target: High Crime (1) or Low Crime (0)
- Threshold: Median of Total Cognizable IPC crimes

Preprocessing & Feature Engineering:

1. Dropped null values
2. Generated a Target column based on median threshold
3. Dropped identifiers like District, Year
4. Standardized features using StandardScaler

Algorithm Used:

- Logistic Regression: Suitable for binary classification of crime levels.

Modeling Steps:

Step 1: Import libraries

```
import pandas as pd
import numpy as np
```

Step 2: Load and clean dataset

```
df = pd.read_csv("crime_data.csv")
df.dropna(inplace=True)
```

Step 3: Create target variable

```
threshold = df['Total Cognizable IPC crimes'].median()
df['Target'] = (df['Total Cognizable IPC crimes'] > threshold).astype(int)
```

Step 4: Feature preparation

```
X = df.drop(columns=['States/UTs', 'District', 'Year', 'Total Cognizable IPC crimes',
'Target'])
y = df['Target']
```

Step 5: Model training

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2,
random_state=42)
```

```
model = LogisticRegression()
model.fit(X_train, y_train)
```

Step 6: Evaluation

```
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
predictions = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, predictions))
print("Confusion Matrix:\n", confusion_matrix(y_test, predictions))
print("Classification Report:\n", classification_report(y_test, predictions))
```

TASK 3

CRIME CLASSIFICATION WEB APP USING STREAMLIT

Introduction:

The Crime Classification App uses Logistic Regression to classify districts as "High" or "Low" crime based on IPC offense data. The application is built using Streamlit for an interactive, user-friendly experience.

How to Implement

Step 1: Installation and Setup

Install necessary libraries:

```
pip install streamlit pandas scikit-learn matplotlib seaborn plotly joblib
```

Step 2: Import Libraries

```
import streamlit as st
import pandas as pd
import joblib
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.graph_objects as go
import plotly.express as px
```

Step 3: Load the Dataset and Trained Model

```
df = pd.read_csv("Credit_Risk_Dataset_with_Loan_Status.csv")
```

```
model = joblib.load("model.pkl")
```

Step 4: Streamlit UI and Navigation

```
st.set_page_config(page_title="RiskLens: Loan Approval Analysis", layout="wide")
st.title("    RiskLens: Loan Approval Analysis")
st.sidebar.title("    Navigation")
page = st.sidebar.radio("Choose Page", ["Raw Data", "Summary", "Graphs & Charts",
"Loan Approval Predictor"])
```

Step 5: User Input Form

```
murder = st.number_input("Murder Cases", 0, 500, 10)
theft = st.number_input("Theft Cases", 0, 1000, 50)
robbery = st.number_input("Robbery Cases", 0, 300, 20)
rioting = st.number_input("Rioting Cases", 0, 200, 15)
cheating = st.number_input("Cheating Cases", 0, 400, 25)
assault = st.number_input("Assault on Women", 0, 600, 30)
others = st.number_input("Other IPC Crimes", 0, 1500, 100)

user_data = pd.DataFrame([[murder, theft, robbery, rioting, cheating, assault, others]],
                         columns=['Murder', 'Theft', 'Robbery', 'Rioting', 'Cheating', 'Assault',
'Others'])
```

Step 6: Make Prediction

```
if st.button("Predict Crime Level"):
    prediction = model.predict(user_data)[0]
    proba = model.predict_proba(user_data)[0][1]

    if prediction == 1:
        st.success(f"✅ High Crime District (Confidence: {proba:.2%})")
    else:
        st.error(f"❌ Low Crime District (Confidence: {1 - proba:.2%})")
```

Step 7: Visual Insights

```
gauge = go.Figure(go.Indicator(
    mode="gauge+number",
    value=proba * 100,
    title={'text': "Crime Probability (%)"},
    gauge={
        'axis': {'range': [0, 100]},
        'bar': {'color': "red" if prediction == 1 else "green"},
        'steps': [
            {'range': [0, 50], 'color': "lightgreen"},
            {'range': [50, 100], 'color': "salmon"}
        ]
    }
))
```

```
st.plotly_chart(gauge)
```

Step 8: Model Accuracy Evaluation

```
X = df.drop(columns=['States/UTs', 'District', 'Year', 'Total Cognizable IPC crimes', 'Target'])
y = df['Target']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)
accuracy = accuracy_score(y_test, model.predict(X_test))
st.write(f"Model Accuracy: {accuracy * 100:.2f}%")
```

Step 9: Run the App

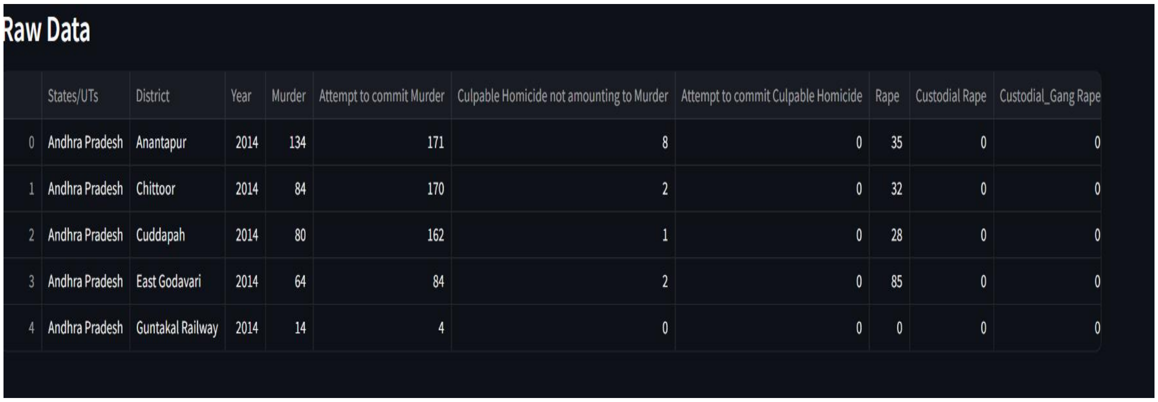
```
streamlit run app.py
```

Step 10: Deploy the App (Streamlit Cloud)

1. Push your project to GitHub
   Repo: GitHub - https://github.com/ganu4533/Crime-dataset-App
2. Visit:: https://crime-dataset-app-ganesh.streamlit.app/
3. Click **"Deploy an App"**
4. Connect your GitHub repo and choose app.py

STREAMLIT APPLICATION INTERFACE OVERVIEW

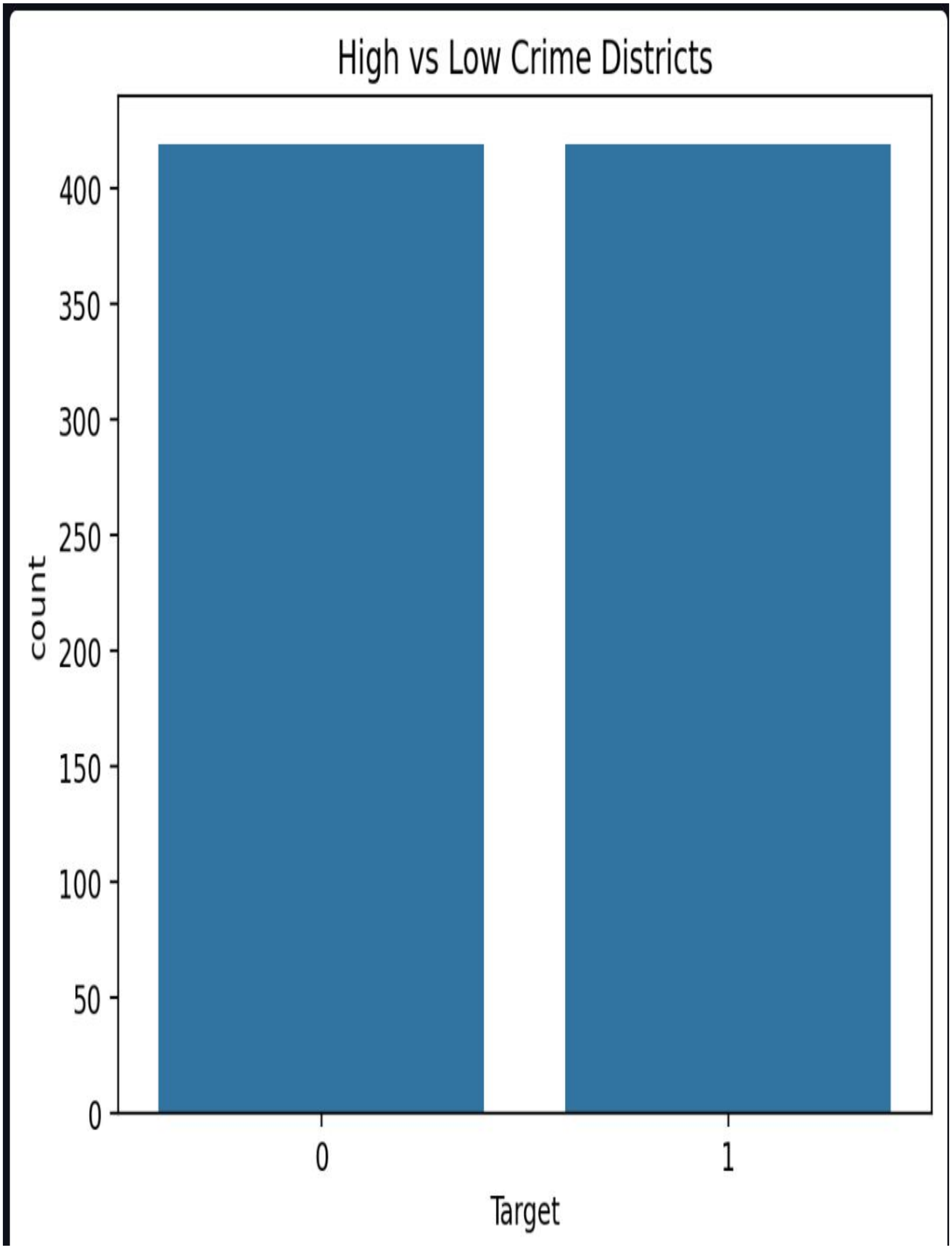
a)Showcase of Overall Raw Data:-



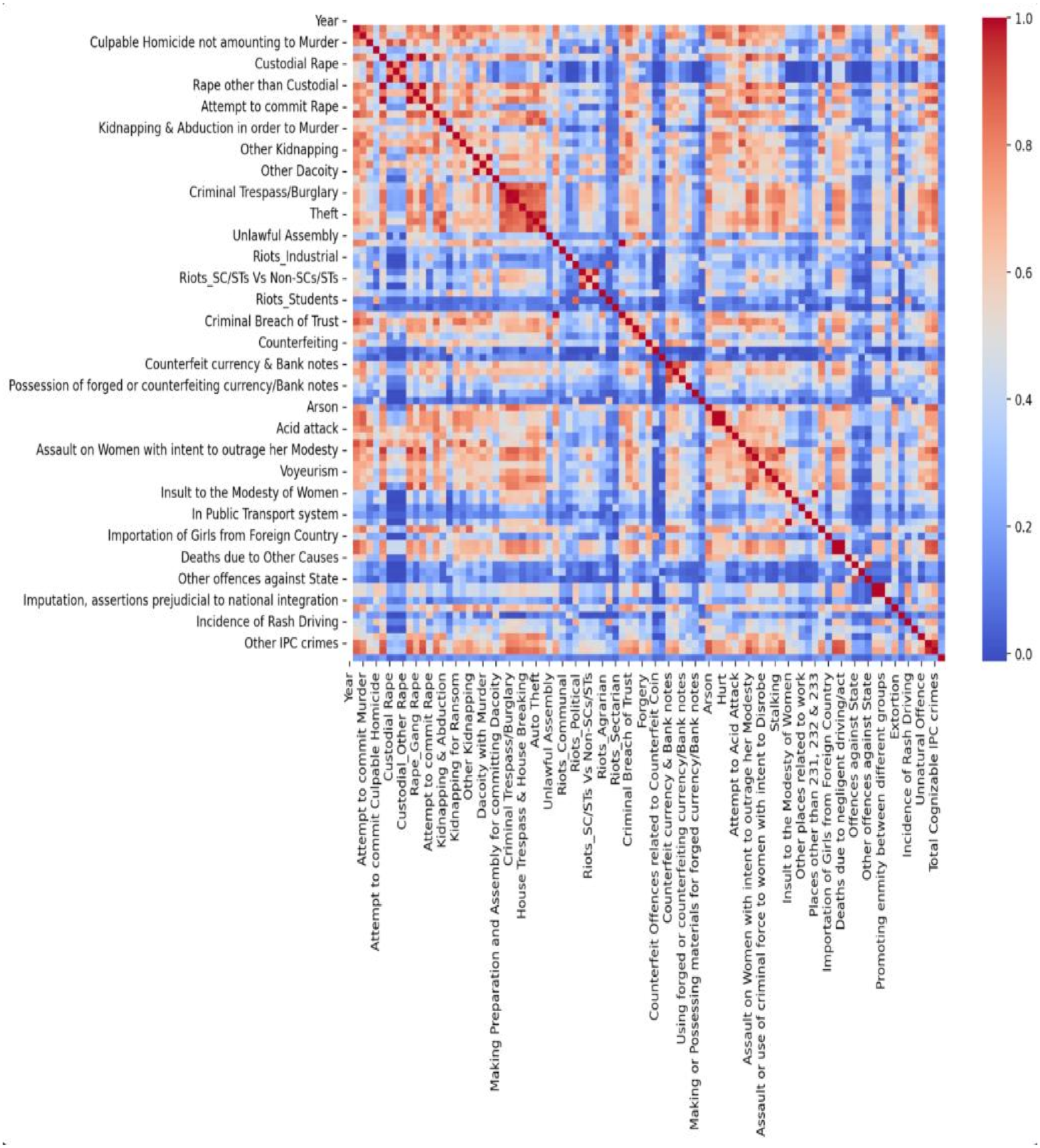
Raw Data

	States/UTs	District	Year	Murder	Attempt to commit Murder	Culpable Homicide not amounting to Murder	Attempt to commit Culpable Homicide	Rape	Custodial Rape	Custodial_Gang Rape
0	Andhra Pradesh	Anantapur	2014	134	171	8	0	35	0	0
1	Andhra Pradesh	Chittoor	2014	84	170	2	0	32	0	0
2	Andhra Pradesh	Cuddapah	2014	80	162	1	0	28	0	0
3	Andhra Pradesh	East Godavari	2014	64	84	2	0	85	0	0
4	Andhra Pradesh	Guntakal Railway	2014	14	4	0	0	0	0	0

b)Graphical represntation of data:



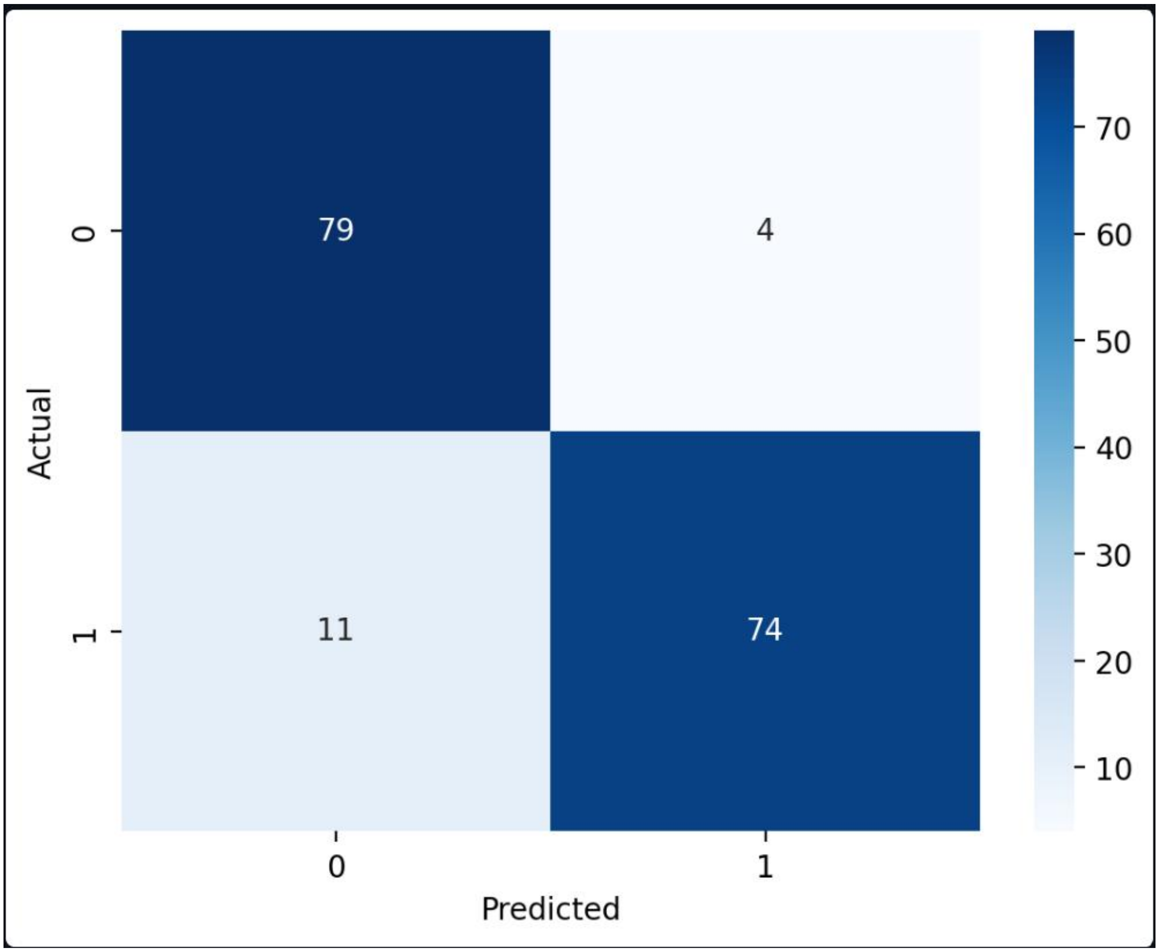
C)Correlation of heatmap



Model Performance

Accuracy: 0.9107142857142857

```
Classification Report
precision recall f1-score support
0 0.88 0.95 0.91 83
1 0.95 0.87 0.91 85 accuracy 0.91 168
 macro avg 0.91 0.91 0.91 168
 weighted avg 0.91 0.91 0.91 168
```



Conclusion:

A fully functional crime classification system was built using Logistic Regression and deployed with Streamlit. The project demonstrates how ML can aid law enforcement by identifying high-crime regions. Data was sourced from data.gov.in, ensuring authenticity and public relevance.

Future Scope:

- Add more algorithms like Random Forest, XGBoost for comparison
- Expand dataset to include socio-economic factors
- Deploy app to Streamlit Cloud or Heroku
- Integrate interactive maps for geo-visualization

References:

- https://data.gov.in
- https://delnet.in
- NCRB Crime Statistics Reports
- Python Libraries: pandas, numpy, scikit-learn, streamlit, matplotlib, seaborn