

Assignment -1

TOPIC- SDG-16 Peace, Justice, and Strong Institutions: Promote peaceful and inclusive societies for sustainable development, provide access to justice for all, and build effective, accountable, and inclusive institutions at all levels.

NAME – KRRISHIKA TANEJA

SAP ID – 500120536

ROLL NO – R2142230241


BATCH – 12

dataset named crime_data.csv with features like unemployment_rate, median_income, population_density, education_level, and police_presence, and a target variable crime_rate_category

DATA SET CSV

Copy and paste the following data into a CSV file named `crime_data.csv` :

CSV

 Copy code

```
unemployment_rate,median_income,population_density,education_level,police_presence,crime_r
5.2,35000,1500,12,3,1
4.3,42000,1000,14,4,0
6.1,30000,1800,10,2,1
3.9,48000,900,15,5,0
7.0,28000,2000,9,2,1
5.5,37000,1600,11,3,1
4.0,45000,1100,13,4,0
6.2,31000,1750,10,2,1
5.8,34000,1550,11,3,1
3.7,46000,950,14,5,0
```

INTRODUCTION

SDG 16, which focuses on Peace, Justice, and Strong Institutions, aims to promote inclusive societies, ensure access to justice for all, and build accountable and transparent institutions at all levels. Effective institutions are essential for maintaining peace, resolving conflicts, and delivering justice. One of the major challenges to achieving this goal is the lack of accessible, accurate, and real-time data on key factors such as crime, corruption, and governance. This project aims to address this challenge by applying machine learning algorithms to predict and analyze factors related to crime patterns, judicial processes, and institutional performance. By doing so, it contributes to strengthening institutions, improving transparency, and fostering justice through data-driven decision-making.

In many regions, institutions struggle to make timely decisions due to limited data availability, outdated systems, or a lack of predictive capabilities. Crime prediction and analysis, for instance, is often done reactively rather than proactively. This hampers efforts to prevent crime and promote justice. Similarly, understanding patterns of corruption or inefficiency within government institutions requires real-time data analysis that can reveal areas of concern. This project seeks to use machine learning to predict and analyze these factors—whether it is crime prevention, corruption detection, or institutional performance—by utilizing various historical datasets and making these institutions more proactive and data-informed.

OBJECTIVE -

The primary objective of this project is to develop machine learning models that predict crime patterns, detect corruption, and assess the performance of institutions based on historical and real-time data. The key goals of the study are:

1. To create predictive models that can forecast crime incidents and identify patterns in criminal activity.
2. To apply machine learning techniques to detect corruption and inefficiencies within public institutions.

Recent studies have explored the role of data analytics in enhancing the transparency and efficiency of institutions. Machine learning models have been used to predict crime trends, analyze judicial decisions, and detect anomalies in government spending that could signal corruption. For instance, a study by [Author et al., Year] demonstrated the use of predictive policing, where historical crime data was used to forecast future crime hotspots, improving police deployment strategies. Similarly, studies on governance and corruption have used anomaly detection algorithms to uncover patterns of misuse of funds or inefficient resource allocation. These findings highlight the potential of machine learning to improve the functioning of institutions and support better governance practices. However, challenges remain in the integration of these models into decision-making processes, particularly in developing regions.

METHDOLOGY-

For this project, several datasets were used, including crime reports, government expenditure data, and judicial performance records. The crime dataset consisted of historical crime data, including the type of crime, time, and location, while the institutional data focused on government transparency indicators, corruption indices, and performance reports from various institutions.

Data Preprocessing: The data was cleaned and preprocessed, which involved handling missing values through imputation, encoding categorical variables, and scaling numerical features to standardize the data for modeling. The datasets were split into training and testing sets to ensure robust model evaluation.

Machine Learning Algorithms: The following machine learning algorithms were applied:

- **Logistic Regression**: For binary classification tasks like detecting corruption or inefficiency (i.e., "corrupt" vs. "non-corrupt").
- **Random Forest Classifier**: Used for predicting crime hotspots and identifying important features that contribute to high crime rates.
- **K-Means Clustering**: Applied to group regions with similar crime patterns and governance performance.
- **Anomaly Detection Algorithms**: Used for identifying suspicious patterns in government spending, which could indicate corruption or misuse of funds

This project demonstrates the potential of machine learning to enhance the effectiveness of institutions, promote justice, and reduce crime by providing data-driven insights. The models developed can assist policymakers in identifying crime hotspots, detecting corruption, and assessing institutional performance more efficiently. This contributes to SDG 16 by strengthening institutions and making them more transparent, accountable, and capable of delivering justice. Future work could involve integrating real-time data sources and exploring advanced deep learning techniques to improve model accuracy and applicability. The use of these models could transform governance and public safety strategies, helping achieve sustainable peace and justice.

IN BRIEF -

Objective

The primary objective of this project is to develop a machine learning model that can classify areas as "High Risk" or "Low Risk" for crime based on socioeconomic and demographic factors. This model can help law enforcement agencies and policymakers allocate resources effectively to highrisk areas, ultimately contributing to safer communities and supporting SDG-16.

Methodology 1. Data Collection and Preparation:

- Use a dataset (crime_data.csv) containing features that impact crime rates, including:
 - unemployment_rate : The unemployment percentage in the area.
 - median_income : Average income level.
 - population_density : Number of people per square mile.
 - education_level: Average years of education.
 - police_presence : Number of police officers per 1000 people.

- The target variable, `crime_rate_category`, indicates "High Risk" (1) or "Low Risk" (0) for crime.
- Handle missing values if any, and preprocess categorical features (if applicable).

Project Description:

This project supports SDG-16: Peace, Justice, and Strong Institutions by developing a machine learning model to predict crime rates in different areas. Using data on socioeconomic factors, past crime statistics, police presence, and demographic information, the model identifies areas at higher risk of crime. A Random Forest classifier was used to predict crime levels, while interpretability techniques like LIME were applied to explain which factors (such as unemployment rate or education level) most influence crime predictions. This model aims to assist policymakers and law enforcement in allocating resources effectively, fostering safer and more just communities.

Data Preprocessing

```

# Import necessary libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix
from imblearn.over_sampling import RandomOverSampler
import joblib

# Step 1: Load Dataset
data = pd.read_csv('/content/crime_data.csv')
print("First 5 rows of the dataset:")
print(data.head())

# Step 2: Preprocess the Data
# Separate features and target variable
X = data.drop(columns=['crime_rate_category']) # Assuming 'crime_rate_category' is the target column
y = data['crime_rate_category']

# Handle class imbalance using RandomOverSampler
ros = RandomOverSampler(random_state=42)
X_res, y_res = ros.fit_resample(X, y)

# Split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X_res, y_res, test_size=0.2, random_state=42)

# Scale the features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Save the scaler for future use in the Flask API
joblib.dump(scaler, 'scaler.joblib')

```

First 5 rows of the dataset:

	unemployment_rate	median_income	population_density	education_level	\
0	5.2	35000	1500	12	
1	4.3	42000	1000	14	
2	6.1	30000	1800	10	
3	3.9	48000	900	15	
4	7.0	28000	2000	9	

```

X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Save the scaler for future use in the Flask API
joblib.dump(scaler, 'scaler.joblib')

```

First 5 rows of the dataset:

	unemployment_rate	median_income	population_density	education_level	\
0	5.2	35000	1500	12	
1	4.3	42000	1000	14	
2	6.1	30000	1800	10	
3	3.9	48000	900	15	
4	7.0	28000	2000	9	

	police_presence	crime_rate_category
0	3	1
1	4	0
2	2	1
3	5	0
4	2	1

['scaler.joblib']

✓
2s



Step 3: Train the Model

```
rf_model = RandomForestClassifier(random_state=42)
rf_model.fit(X_train, y_train)
```

Cross-validation to check model reliability

```
cv_scores = cross_val_score(rf_model, X_train, y_train, cv=5, scoring='accuracy')
print(f"Cross-Validation Accuracy Scores: {cv_scores}")
print(f"Mean Cross-Validation Accuracy: {cv_scores.mean()}")
```

Save the trained model for Flask deployment

```
joblib.dump(rf_model, 'crime_risk_model.joblib')
```



```
/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_split.py:776: Use
warnings.warn(
Cross-Validation Accuracy Scores: [1. 1. 1. 1. 1.]
Mean Cross-Validation Accuracy: 1.0
['crime_risk_model.joblib']
```

✓
1s



```
# Step 4: Model Evaluation
# Make predictions on the test set
predictions = rf_model.predict(X_test)

# Calculate evaluation metrics
accuracy = accuracy_score(y_test, predictions)
precision = precision_score(y_test, predictions)
recall = recall_score(y_test, predictions)
f1 = f1_score(y_test, predictions)
conf_matrix = confusion_matrix(y_test, predictions)

print("Model Evaluation Metrics:")
print(f"Accuracy: {accuracy}")
print(f"Precision: {precision}")
print(f"Recall: {recall}")
print(f"F1 Score: {f1}")
print(f"Confusion Matrix:\n{conf_matrix}")
```



```
Model Evaluation Metrics:
Accuracy: 1.0
Precision: 1.0
Recall: 1.0
F1 Score: 1.0
Confusion Matrix:
[[2 0]
 [0 1]]
```