Project Title: Crime Type Prediction and Pattern Analysis in Kigali

Objective

To predict crime types in Kigali and analyze patterns based on the following factors:

- **Location** and **Month**: Identify where and when each crime type occurs most frequently.
- **Gender and Age Group**: Compare crime patterns among males, females, and others, across:
  - Younger: 15–30
  - Adult: 31–60
  - Elderly: 61–80
- **Role Comparison**: Understand differences between **victims** and **suspects**.

Project Requirements

1. **Dataset**
   Use the provided dataset (`kigali_crime_data.csv`) containing 1000 rows. Focus only on Kigali locations and exclude the `weather` column.
2. **Preprocessing**
   - Extract `month` and `hour` from datetime fields.
   - Encode categorical variables (`location`, `gender`, `crime_type`, `role`).
   - Fix the previous issue with age scaling.
3. **Modeling**
   - Implement **Random Forest** and **Logistic Regression** models to predict `crime_type`.
   - Conduct pattern analysis based on location, month, gender, age group, and role.
4. **Evaluation**
   - Evaluate models using accuracy and classification reports.
   - Interpret demographic and location-based patterns.
5. **Visualizations**
   - Create simple and clear charts and graphs to communicate findings to all audiences.
6. **Dashboard**
   - Develop an **interactive Power BI dashboard** for dynamic exploration of crime trends and patterns.
7. **Final Report/Presentation**
   - To be developed later as per presentation requirements.

Key Concepts and Terminology

- **Normalization**: Scaling numeric features for model input (e.g., age), while keeping raw values for analysis and visualization.
- **Classification**: Predicting categories (e.g., crime types like Theft, Assault).
- **Pattern Analysis**: Uncovering trends such as seasonal spikes or demographic-based crime patterns.
- **Dashboard**: An interactive tool (in Power BI) to filter and visualize data insights.

Step 2: Dataset Description and Adjustments

Dataset Summary

- File: `kigali_crime_data.csv`
- Fields include: `crime_id`, `location`, `date`, `time`, `crime_type`, `gender`, `role`, `age`, `weather`, `severity`, `latitude`, `longitude`.
- The column `weather` will be excluded.
- `severity` will be used to predict high-severity crimes.

Age Scaling Fix

- Keep original `age` values (15–80) for all analysis and visualizations.
- Use `StandardScaler` only within the modeling phase.

Key Python Libraries Used

Library	Use Case
`pandas`	Tabular data manipulation (like Excel)
`numpy`	Numerical computation
`datetime`	Date and time management
`random`	Random sampling
`scikit-learn`	Machine learning models and preprocessing
`folium`	Interactive mapping
`matplotlib`/`seaborn`	Data visualization

Step 3: Analysis & Modeling

Preprocessing Summary

- Extract `month` from `date` and `hour` from `time`.
- Encode categorical variables.

- Create age groups.
- Normalize features only during modeling.

Modeling Techniques

- **Random Forest Classifier**
- **Logistic Regression**
- Additional task: Predict high-severity crimes.

Features Used for Modeling

- Encoded `location`, `month`, `hour`, `gender`, `role`, `age`, `severity`

Evaluation Metrics

- Accuracy
- Classification reports
- Feature importance

Visualization Outputs

- Crime by location (bar chart)
- Crime by month (bar chart)
- Crime hotspots map (interactive, with `folium`)
- Gender vs. location (stacked bar chart)
- Age group vs. location (stacked bar chart)
- Role vs. location (stacked bar chart)

Here's a **combined, detailed explanation** of the code with a focus on the **machine learning models, particularly Random Forest**, and all related concepts in the context of the Kigali crime dataset analysis:

🔍 1. Data Loading and Preprocessing

The dataset `kigali_crime_data.csv` is read into a Pandas DataFrame, and several preprocessing steps are applied:

Categorical Encoding:

Categorical variables (`location`, `crime_type`, `gender`, `role`) are encoded using **LabelEncoder**, transforming them into integers for use in ML models.

```
le_location = LabelEncoder()
df['location_encoded'] = le_location.fit_transform(df['location'])
```

Temporal Features:

- **Month** is extracted from the `date`.
- **Hour** is extracted from the `time`.

Age Grouping:

- Converts `age` into bins: Younger, Adult, Elderly for better demographic analysis.

Feature Scaling:

`age` and `severity` are scaled using `StandardScaler` to normalize values for better model performance.

🤖 2. Machine Learning Models

✨ Objective 1: Predict Crime Type (Multiclass Classification)

🏗 Features Used:

```
['location_encoded', 'month', 'hour', 'gender_encoded', 'role_encoded',
'age_scaled', 'severity_scaled']
```

🎯 Target:

```
crime_type_encoded (represents different types of crimes)
```

🌲 Random Forest Classifier (RFC)

🔎 What is it?

Random Forest is an ensemble machine learning algorithm based on **decision trees**. It combines multiple trees to improve accuracy and avoid overfitting.

🧠 How it works:

- Trains multiple **decision trees** on random subsets of data and features.
- Uses **majority voting** for classification.

- Offers **feature importance** scores, showing which features most influence the model.

✔️ Why use Random Forest?

- Handles both numerical and categorical features.
- Robust to noise and outliers.
- Offers interpretability through `feature_importances_`.

📊 Evaluation:

```
accuracy_score(y_test, rf_pred)
classification_report(y_test, rf_pred)
```

Reports **precision, recall, F1-score** for each crime type.

🔥 Feature Importance Output:

Ranks features like `location`, `role`, `hour`, etc., based on how much they influence the predictions.

➕ Logistic Regression (Baseline Model)

What it does:

A linear model that estimates probabilities using a logistic function. Used here as a **benchmark**.

```
lr_clf = LogisticRegression()
```

It's generally less powerful than Random Forest for non-linear problems but good for interpretable and fast training.

🎯 Objective 2: Predict High Severity (Binary Classification)

Target:

```
df['high_severity'] = (df['severity'] > 5).astype(int)
```

This turns severity into a **binary** label: `1 = high`, `0 = low`.

Model:

Another **Random Forest Classifier** is trained and evaluated similarly.

📈 3. Pattern Analysis Using Grouping

Grouping and counting crimes based on:

- `location` and `month`
- `location` and `gender`
- `location` and `age_group`
- `location` and `role`

These help understand **crime distribution trends**.

🎨 4. Visualizations

Seaborn and Matplotlib

Used to generate:

- Crime types by location
- Crime types by month
- Crimes by gender, age group, and location

Saved as PNGs.

🗺️ 5. Crime Hotspot Mapping with KMeans and Folium

KMeans Clustering:

Used to find **5 clusters** (crime hotspots) based on GPS coordinates.

```
kmeans = KMeans(n_clusters=5)
```

KMeans groups similar points (latitude and longitude) into clusters.

Folium Map:

Plots clustered crime locations with colored markers based on `kmeans` cluster. Popup info includes `location`, `crime_type`, and `severity`.

Saved as an **interactive HTML map**: `kigali_crime_hotspots.html`

🧠 Summary of Machine Learning Concepts

Concept	Explanation
Label Encoding	Converts text labels into integers.
StandardScaler	Normalizes numerical data (mean=0, std=1).
Train-Test Split	Splits data into training (80%) and test (20%) for model evaluation.
Random Forest	Ensemble of decision trees, good for both classification and regression, robust and interpretable.
Logistic Regression	A simple linear model for classification problems.
Accuracy	Proportion of correct predictions.
Classification Report	Includes precision, recall, and F1-score.
Feature Importance	Measures how useful each feature was in the model.
KMeans Clustering	Groups similar data points into clusters.
Folium	Visualizes data on maps (great for geospatial data).