Data Mining Final Project

Team Members:

- Farah Walid (23010036)
- Nour Essam (23010035)
- Basmala Hossam El-Din (23011052)
- Rodina Mohamed (23010014)
- Mariam Ahmed (23012058)

Team ID: 17

Introduction:

Understanding crime patterns in urban areas is essential for improving public safety and resource allocation. San Francisco, like many metropolitan cities, faces a diverse range of criminal activities across its neighborhoods. By analyzing historical crime data, we can uncover hidden patterns and group similar types of incidents geographically or behaviorally. This project aims to apply unsupervised and supervised mining techniques to explore, cluster, and classify crime data in San Francisco.

The primary focus of this project is to use clustering algorithms such as Hierarchical Clustering to identify natural groupings within the crime data. These clusters can represent geographical crime hotspots, types of incidents, or behavioral similarities. Once clusters are formed, we further explore the potential of supervised learning by applying K-Nearest Neighbors (KNN) to classify new crime records based on their similarity to previously identified clusters.

This project provides insights into how clustering combined with classification can be a powerful tool for crime pattern detection and forecasting in public safety.

Why we chose this data?

The San Francisco Crime dataset was chosen due to its rich, real-world relevance and diversity of features it offers for analysis. This diversity makes it ideal for both clustering and classification to discover natural groupings in data and to predict future crime patterns.

Dataset Description:

The dataset used in this project is originally made available through Kaggle. It contains detailed information on over 800,000 crime incidents reported in San Francisco between January 2003 and May 2015. Each record represents a single crime event and includes both categorical and numerical features.

Data Preprocessing:

Before applying any machine learning algorithms, several preprocessing steps were required to clean, transform, and prepare the San Francisco crime data for analysis.

Sampling:

Due to the large size of the original dataset (~800,000 records), we extracted a **random sample of 15,000 entries** to make the clustering and training processes computationally feasible. This sample retained a representative distribution of days, districts, and locations.

Feature Selection:

We selected a subset of features that were most relevant to our clustering and classification objectives:

- **X and Y**: Geographical coordinates were essential for detecting spatial crime patterns.
- **DayOfWeek**: Converted into numerical values to represent temporal variation in crimes.
- **PdDistrict**: Encoded as categorical data to preserve information about police jurisdictions.
- **Category**: The crime Category column provides crucial semantic context and is valuable for potential supervised learning tasks or for evaluating the quality of the unsupervised clusters.

Encoding Categorical Variables:

To convert non-numeric data into machine-readable format:

- **DayOfWeek** was label encoded (e.g., Monday = 0, Tuesday = 1, etc.).
- **PdDistrict** was label encoded to prevent introducing ordinal bias into the model.

Scaling of Coordinates:

Since clustering and KNN are distance-based algorithms, we **standardized only the 'X' and 'Y' features** using "StandardScaler". This ensured that the scale of latitude and longitude did not distort the distance calculations, while other features remained in their original or encoded form. We deliberately avoided scaling the entire feature set to preserve interpretability.

Train-Test Split:

After preprocessing, the data was split into:

- **Training Set**: 70% of the sample (used for clustering and training KNN)
- **Test Set**: 30% of the sample (used for evaluating KNN performance)

Special care was taken to fit the scaler only on the training data, and not leak information from the test set during preprocessing.

Hierarchical Clustering:

Hierarchical Clustering is an agglomerative clustering algorithm that builds a tree (or dendrogram) of clusters by iteratively merging the closest pairs of data points. It does not require specifying the number of clusters in advance and provides a visual representation of cluster relationships.

Implementation Steps:

- Applied **Clustering** with linkage='ward' and metric='euclidean'.
- Used **scaled X and Y coordinates** along with encoded categorical features for clustering.
- Visualized the clustering hierarchy using a **dendrogram**, which helped determine the optimal number of clusters.

Cluster Labeling: After inspecting the dendrogram, we selected a cutoff that resulted in **5 distinct clusters**. These cluster labels were then assigned to each data point in the training set.

Classification with K-Nearest Neighbors (KNN):

Following the clustering stage, we transitioned into supervised learning by treating the cluster labels obtained from Hierarchical Clustering as pseudo-class labels. Our goal was to use these labels to train a classifier capable of predicting the cluster for new or unseen data.

Motivation:

Using KNN allowed us to evaluate how well the clusters generalize and whether new crime data could be effectively categorized into one of the discovered clusters. This approach simulates real-world scenarios where new crime reports are assigned to known patterns or regions for analysis or intervention.

Data Preparation:

- **Features (X)**: Used the same scaled features as those used in clustering (specifically scaled X, Y, and encoded temporal/district data).
- **Target (y)**: Used the cluster labels from Hierarchical Clustering as the target classes.
- **Train-Test Split**:
    - 70% of the sample used to train the KNN model.
    - 30% used for testing its predictive performance.
    - **Important Note**: To better reflect real-world data input, we **did not scale the test data** (beyond coordinates), preserving their raw structure.

Results:

- KNN successfully learned the pseudo-labels from the training set.
- The accuracy indicated a strong correlation between spatial and categorical features and the identified clusters.
- The classification task validated that the clusters were meaningful and could generalize to new data.

Metric Evaluation and Results:

To assess the effectiveness of our clustering and classification efforts, we evaluated both the **quality of the clusters** generated by Hierarchical Clustering and the **accuracy** of the K-Nearest Neighbors classifier trained on those cluster labels.

Hierarchical Clustering Evaluation:

- **Visual Inspection via Dendrogram**: The dendrogram provided insights into the natural groupings within the dataset. We chose a cutoff point that generated **5 clusters**, balancing detail and interpretability.
- **Silhouette Score**: A silhouette score help quantifying the cohesion and separation of clusters. Scores closer to 1 indicate well-defined clusters, while scores near 0 suggest overlapping.

KNN Classification Evaluation:

- **Accuracy Score**: The classifier achieved an accuracy of approximately **84%** on the test data, using the cluster labels from Hierarchical Clustering as targets. This indicated that the model was able to generalize and recognize patterns in unseen data.

Visualization:

Visualizations played a critical role in understanding the spatial and structural patterns in the dataset. We utilized several types of visual tools throughout the analysis:

- **Bar Plot: "Number of Incidents by Crime Category"** This plot illustrated the frequency of each crime type, providing insight into which categories were most prevalent. It helped us understand the distribution of crime types and identify dominant classes like theft, assault, and drug offenses.
- **Scatter Plots (X vs. Y)**: After clustering, we visualized the crime data using scatter plots colored by cluster assignment. This helped in interpreting the spatial distribution of clusters and validating the geospatial coherence of our results.
- **Bar Plot: "Number of Incidents by Hour of the Day"** To capture temporal crime trends, we plotted the number of incidents for each hour. This visualization revealed peaks during certain times of the day, such as late evenings or early mornings, suggesting patterns that could be tied to human behavior and urban activity cycles.

Conclusion:

This project successfully combined clustering and classification techniques to analyze the San Francisco Crime dataset.

Hierarchical Clustering enabled us to identify natural groupings in the data, and visual tools like dendrograms and scatter plots validated these clusters. We then used K-Nearest Neighbors (KNN) to classify new data based on these learned clusters, effectively simulating a semi-supervised learning approach.

Visualizations including bar plots for crime category and hour, and scatter plots for spatial distribution enhanced our understanding of crime dynamics and guided data preprocessing choices.

Challenges Faced:

- **Data Imbalance and Noise**: The SF Crime dataset contains a large number of categories for crime types, some of which appear very infrequently. This imbalance, combined with potential inconsistencies or noise in location data (e.g., missing or outlier coordinates), required careful filtering and normalization.
- **Scaling Geographic Features**: Since the dataset contains geographic coordinates (X and Y), proper scaling was necessary to ensure these values did not dominate distance-based clustering algorithms. At the same time, we had to avoid distorting spatial relationships that are critical for meaningful clustering.
- **Choosing the Number of Clusters**: One of the fundamental challenges in clustering is determining the optimal number of clusters (5). Without ground truth labels, we relied on dendrograms, elbow methods, and silhouette scores, which are sometimes ambiguous or conflicting.
- **Unsupervised to Supervised Transition**: Using cluster labels as targets for KNN classification introduced a new layer of complexity. These labels are not "true" classes but groupings inferred from patterns, so their interpretability and consistency across algorithms had to be handled cautiously.
- **Computational Load**: Clustering large datasets (especially hierarchical clustering) is computationally expensive. To address this, we had to sample the dataset and optimize preprocessing steps to ensure reasonable runtimes without compromising too much accuracy.
- **Scaling Strategy for KNN Evaluation**:
  During KNN implementation, ensuring that only training data was used to fit the scaler was crucial to avoid data leakage. Special care was taken to scale only selected features like X and Y, and to avoid scaling test data in some experiments to maintain real-world interpretability.

Team Member's Roles:

Basmala: Data Collection & Preprocessing.

Nour: Hierarchical Clustering & Notebook Markdowns.

Mariam: Classification with K-Nearest Neighbors (KNN).

Rodina: Evaluation Metrics.

Farah: Visualizations & Report Documentation.

The End