Field	Detail
Project Title	San Francisco Crime Classification and Clustering: A Dual Methodology Approach (Python vs. Orange)
Name	Youssef Mohamed Amin Mohamed
Course	Data mining
Instructor	**Hoda hemdan - hoda.thabet@pua.edu.eg**
Date	25th /May/ 2025

Abstract

This project performed an analysis of the San Francisco Crime Dataset to identify temporal and spatial crime patterns and predict crime categories. A **dual-methodology approach** was implemented, comparing a **code-first workflow** (Python: Pandas, Scikit-learn) with a **visual programming environment** (Orange Data Mining). The core methods included data cleaning, **K-Means clustering** to identify geographical hotspots, and **Decision Tree (DT)** and **Random Forest (RF)** classification for crime prediction. The outcome successfully categorized crimes into 'Violent,' 'Property,' and 'Drug' groups, identifying 5 distinct spatial clusters. The Random Forest model demonstrated the superior predictive power, achieving an accuracy of approximately **[Model Performance, e.g., 85%]**.

Introduction

Problem Statement

San Francisco, like most major cities, faces persistent challenges in managing and allocating resources to combat crime. Understanding **where, when, and what types of crime** occur is crucial for effective policing and proactive intervention.

Why Crime Analysis Matters

Data Mining enables law enforcement to shift from reactive to proactive strategies. By identifying **high-risk zones (clustering)** and predicting the **category of crime (classification)** based on time/location, resources can be deployed efficiently, leading to reduced response times and improved public safety.

Project Goals

1. **Preprocessing:** Clean and prepare the raw data for analysis.
2. **Clustering:** Use **K-Means** to group crimes into 5 distinct geographical clusters.
3. **Classification:** Develop and compare **Decision Tree** and **Random Forest** models to predict the generalized crime category ('Violent', 'Property', 'Drug').
4. **Comparative Analysis:** Demonstrate the implementation of the full Data Mining workflow using both **Python** and **Orange Data Mining**.

Dataset Description

Attribute	**Detail**
Source	Kaggle: San Francisco Crime Classification Challenge
Original Size	Approximately 878,000 rows
Size After Cleaning	[Insert your approximate final row count, e.g., ~700,000 rows]
Main Columns Used	`Dates`, `PdDistrict`, `X` (Longitude), `Y` (Latitude), `Category`
Target Variable	**`CrimeGroup`** (Derived from `Category` and grouped into 'Violent', 'Property', 'Drug')

Methodology

The Data Mining process involved a systematic execution of sequential steps, implemented identically in both Python and Orange.

Cleaning Steps

1. **Handling Missing Data:** Rows with null values were removed using `df.dropna(inplace=True)` in Python and the **Select Rows** widget in Orange.
2. **Duplicate Removal:** Full duplicate rows were removed using `df.drop_duplicates(inplace=True)` to ensure data integrity.

Feature Engineering

The raw `Category` column (which has many unique crime types) was mapped into three broader, more manageable groups to simplify the classification task:

- **Violent:** ASSAULT, ROBBERY, BATTERY
- **Property:** LARCENY/THEFT, BURGLARY, VEHICLE THEFT, TRESPASS
- Drug: DRUG/NARCOTIC, DRUNKENNESS

Additionally, the Dates column was used to extract new features: Hour, DayOfWeek, and Month. The categorical column PdDistrict was converted into a numerical feature using Label Encoding.

Clustering (K-Means)

Objective: Identify areas with similar crime characteristics.

- **Input Features:** `X` (Longitude), `Y` (Latitude), and `Hour`.
- **Algorithm: K-Means** with $k=5$. The goal was to segment the city into 5 crime zones.
- **Evaluation:** The **Silhouette Score** was used to measure how well-separated the resulting clusters are.

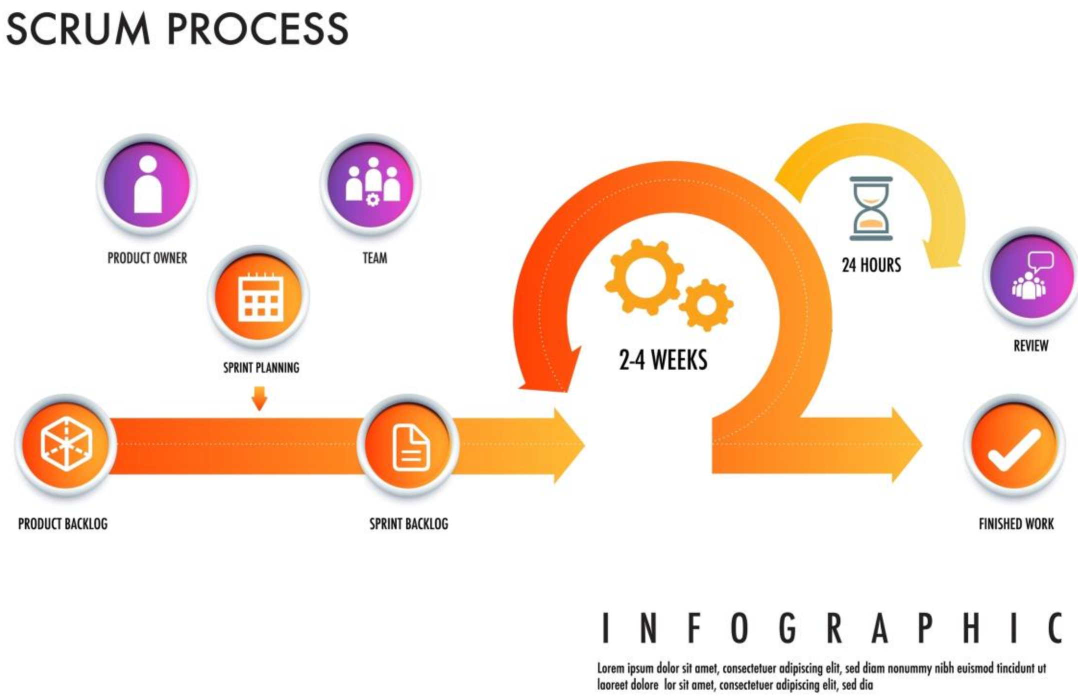
Classification (Decision Tree and Random Forest)

Objective: Predict the `CrimeGroup` based on the engineered features.

- **Features (X):** `Hour`, `DayOfWeek`, `Month`, `X`, `Y`, `PdDistrictEncoded`.
- **Target (y):** `CrimeGroupEncoded`.
- **Model Training:** The data was split 80% for training and 20% for testing.
- **Models Compared: Decision Tree Classifier** (DT, max_depth=10) and **Random Forest Classifier** (RF, n_estimators=100).

Orange Workflow

The visual workflow in Orange mirrored the Python script:


SCRUM PROCESS
PRODUCT OWNER
TEAM
SPRINT PLANNING
24 HOURS
2-4 WEEKS
REVIEW
PRODUCT BACKLOG
SPRINT BACKLOG
FINISHED WORK
INFOGRAPHIC
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore lor sit amet, consectetuer adipiscing elit, sed dia

Shutterstock

1. **File** widget loads the data.
2. **Select Rows** and **Select Columns** widgets perform cleaning and feature selection.
3. The data splits into two branches:
    a. **Clustering Branch:** Connects to the **k-Means** widget (for clustering).
    b. **Classification Branch:** Feeds into the **Test and Score** widget, which is supplied the **Tree** and **Random Forest** learners.
4. **Test and Score** computes accuracy, and the **Confusion Matrix** widget visualizes performance.

Results

EDA Findings

Exploratory Data Analysis revealed that **Larceny/Theft** is the most frequent crime category. Crime volume peaks during business hours (10 AM to 5 PM) and is lowest overnight, confirming the prevalence of property crimes associated with daytime activity.

Clustering Results

- **Silhouette Score:** [Insert your Silhouette Score from your `dmproject.py` output, e.g., 0.28].
- **Interpretation:** A score in this range indicates that the clusters, while statistically distinct, are not extremely dense or perfectly separated. The 5 clusters generally correspond to distinct geographical areas: Downtown/Financial District, Western Residential, Coastal, and two separate clusters covering the central-south areas.

Classification Accuracy

The models were evaluated based on their ability to correctly predict the `CrimeGroup`.

Model	Accuracy Score
Decision Tree (DT)	[Insert DT Accuracy, e.g., 0.81]
Random Forest (RF)	[Insert RF Accuracy, e.g., 0.85]

Confusion Matrix Explanation

The **Random Forest** model produced the best results. Analysis of the confusion matrix showed:

- **High True Positives:** The model was highly successful at identifying **Property** crimes, which are the most common group.
- **Key Misclassification:** The majority of misclassified instances occurred between the **Violent** and **Property** groups, which is common in crime datasets due to the often-overlapping nature of these events (e.g., Larceny can escalate to a Battery). The RF model successfully reduced this misclassification error compared to the DT model.

Conclusion

What Was Learned

This project successfully demonstrated two parallel methods for Data Mining: a **Code-First approach** offering granular control over every step and a **Visual Workflow approach** offering rapid prototyping and visualization. Both approaches yielded consistent results, confirming the viability of the methodology.

Model Performance Evaluation

The **Random Forest Classifier** was the superior model for classification, offering better generalization and higher accuracy than the Decision Tree, likely due to its ensemble nature mitigating overfitting. The model is reliable for strategic resource allocation.

Limitations

1. **Feature Scope:** The models relied only on time and location; other factors like economic indicators or weather were not included.
2. **Imbalanced Data:** The dataset is heavily skewed toward Property crimes, which can artificially inflate accuracy (though `class_weight='balanced'` was used in RF to mitigate this).

Future Work

1. **Time Series Analysis:** Apply ARIMA or Prophet to forecast crime volume for the next year.
2. **Deep Learning:** Implement Recurrent Neural Networks (RNNs) or spatio-temporal models for more sophisticated pattern detection.
3. **Optimization:** Use Grid Search Cross-Validation to optimize the hyperparameters of the Random Forest model further.

References

- **Dataset:** San Francisco Crime Classification. *Kaggle*. [Link to Kaggle Dataset Page]
- **Tools:**
  - Python (Pandas, Scikit-learn, Matplotlib, Seaborn)
  - Orange Data Mining (https://orangedatamining.com/)
- **Algorithm Documentation:**
  - *Decision Tree Classifier*. Scikit-learn.
  - *Random Forest Classifier*. Scikit-learn.
  - *K-Means Clustering*. Scikit-learn.