AI CRIME PREDICITION SYSTEM

CRIME PREDICTION MODEL USING SAN FRANCISCO CRIME DATASET

ASIF ULLAH (TEAM-LEADER), ALISHBA FATIMA, MUHAMMAD HUZAIFA IFTIKHAR, MUHAMMAD ATIF LAGHARI, AHSAN ALI, NAVEERA FATIMA

AI-Augmented Crime Prediction and Explainability System

Project Name: AI Crime Predictor

Model Used: XGBoost Classifier (Optimized for Multi-Class Classification)

Explainability Layer: Llama 3.3 (via Groq API) **Goal:** To accurately predict the category of a crime incident based on spatio-temporal features and unstructured text, and to provide a real-time, human-readable explanation and validation of the prediction.

1. Project Aim:

The primary aims of this project are twofold, combining predictive power with transparent reasoning:

- **Prediction Accuracy:** To accurately predict the category of a crime incident based on spatio-temporal features (location, time) and unstructured text (Address, Description).
- **Explainability & Validation:** To provide a real-time, human-readable explanation and critical validation of the predictive model's output using a Large Language Model (LLM, Llama 3.3). This unique feature acts as an "AI Analyst" to detect and critique potential misclassifications by the core XGBoost model.

2. Project Overview and Architecture

This system is built as a hybrid AI solution designed to overcome the limitations of traditional machine learning models by integrating a powerful, fast predictive model with a Large Language Model (LLM) for reasoning and explainability.

Architecture Components:

- **Data Ingestion & Feature Engineering:** Raw San Francisco crime data is pre-processed into numerical features.
- **Predictive Engine (XGBoost):** A highly efficient gradient boosting model trained for multi-class classification.
- **Explainability Layer (Llama 3.3):** Provides contextual validation, explanation, and critique of the XGBoost model's output based on user input and prediction confidence.
- **Presentation Layer (Streamlit):** A user-friendly interface for inputting incident details and displaying both the prediction and the AI-generated explanation.

3. Data Preprocessing and Feature Engineering: The model relies on a dense set of features derived from time, location, and unstructured text. The total feature space consists of **~1291 features** per incident.

Feature Type	Source Column	Engineering Technique	Purpose

Spatio-Temporal	X, Y, Year, Month, Day, Hour, Minute	Direct Numeric Mapping	Captures location and time seasonality (e.g., crime is higher at night).
Cyclical Time	Hour	Sine/Cosine Transformation	Encodes time of day as a continuous, cyclical feature to preserve the relationship between 11:59 PM and 12:00 AM.
Categorical	PdDistrict, DayOfWeek	Label Encoding	Converts nominal categories into numerical values for XGBoost.
Text Feature 1 (Sparse)	Address	Feature Hashing (1024 Bins)	Efficiently maps location text into a sparse vector representation.
Text Feature 2 (Sparse)	Descript	Feature Hashing (256 Bins)	Maps unstructured description text into a vector; crucial for semantic interpretation.

Vector Construction: All engineered features are combined into a single sparse matrix (csr_matrix) using horizontal stacking (hstack) for efficient training and prediction.

4. XGBoost Predictive Model Details

The model is optimized for high accuracy and fast inference, crucial for the real-time Streamlit application.

Parameter/Setting	Value/Method	Rationale
Model	xgb.XGBClassifier	High performance on structured data and mixed feature types.
Objective	multi:softprob	Outputs probabilities for all 39 crime categories.

n_estimators	700+ (Tuned)	Sufficient trees for convergence.
learning_rate	0.05	Conservative rate to prevent overshooting the minimum.
max_depth	8	Prevents deep, overfit trees while allowing complex interaction capture.
Training Method	gpu_hist (fallback to hist)	Prioritizes GPU acceleration for fast training.
Imbalance Handling	compute_class_weight('balanced')	Dynamically weights rare crime categories during training, improving recall for minority classes.
Stop Condition	Early Stopping (50 rounds)	Prevents overfitting to the validation set.

5. Explainability and Validation Layer (Llama 3.3)

The LLM component is the defining feature of this project, mitigating the "black box" nature and potential errors of the core XGBoost model.

5.1. The Role of the LLM

- **Model:** Llama 3.3 (accessed via Groq).

- **Function:** After XGBoost provides the predicted category and confidence, the LLM takes the original text inputs (Address and Description) and the final prediction.
- **System Prompt:** The LLM is given a system prompt instructing it to act as an "AI Analyst" who must provide a concise explanation, and critically, **validate the prediction against the semantic meaning of the description text.**

5.2. Handling Model Discrepancy (The Key Feature]

As demonstrated in testing, the XGBoost model can sometimes misclassify an incident due to feature hashing collision or feature dominance (e.g., location weight overriding text weight).

- **Example Scenario:** If a user inputs a description like "Laptop bag stolen from car" but the model predicts **DISORDERLY CONDUCT**, the LLM's role is to:
    1. Identify the keyword "stolen."
    2. Reason that "stolen" belongs to the **LARCENY/THEFT** category.
    3. Generate an explanation that *critiques* the XGBoost output, stating, "The model might have classified this as DISORDERLY CONDUCT, but the description clearly indicates a stolen item. The predicted category should logically be THEFT."

This hybrid approach ensures high throughput for prediction while adding a robust layer of **contextual verification and human-like reasoning.**

6. Deployment and User Experience:

The application is deployed using **Streamlit**, which provides a rapid, reactive front-end interface.

- **Input Interface:** Users provide the date, time, coordinates, district, and a detailed description of the incident.
- **Output Panel:** Displays the XGBoost prediction, the confidence score, and the **Top 3 Predicted Categories** for transparency.
- **Chat Interface:** Allows users to interact with the LLM to ask follow-up questions about the prediction, historical crime trends, or the model's structure.