SF Crime Forecasting

CS133 - Data Visualization

San Jose State University

Professor Jessica Huynh-Westfall

May 5, 2025

RuiYuanLi,Tang Ho Lam

1. Abstract

This project investigates the forecasting of crime patterns in San Francisco using historical police incident data from 2018 to the present. The primary goal is to develop machine learning models that can predict the number and types of crimes based on temporal and geographic indicators such as hour of day, day of week, and neighborhood. Crime in urban areas often follows temporal and spatial patterns, and identifying these trends can assist law enforcement in resource planning, community outreach, and prevention strategies.

The dataset used contains detailed incident reports including crime categories, timestamps, police districts, and neighborhood information. After data cleaning and feature engineering—including cyclic transformations of time-based data and one-hot encoding of categorical variables—a linear regression model was trained to predict hourly crime counts. The model achieved a reasonable R^2 score and low error metrics, confirming the presence of predictive signals in the dataset.

The future scope includes building classification models to predict crime types, constructing interactive geospatial visualizations using Folium, and incorporating real-time forecasting for public safety tools. This project demonstrates how data science can contribute to practical, socially relevant solutions in public policy and urban management.

2. Background & Research Questions

Urban safety remains a pressing issue. Crime trends vary by hour, day, and location. This project focuses on the following five questions:

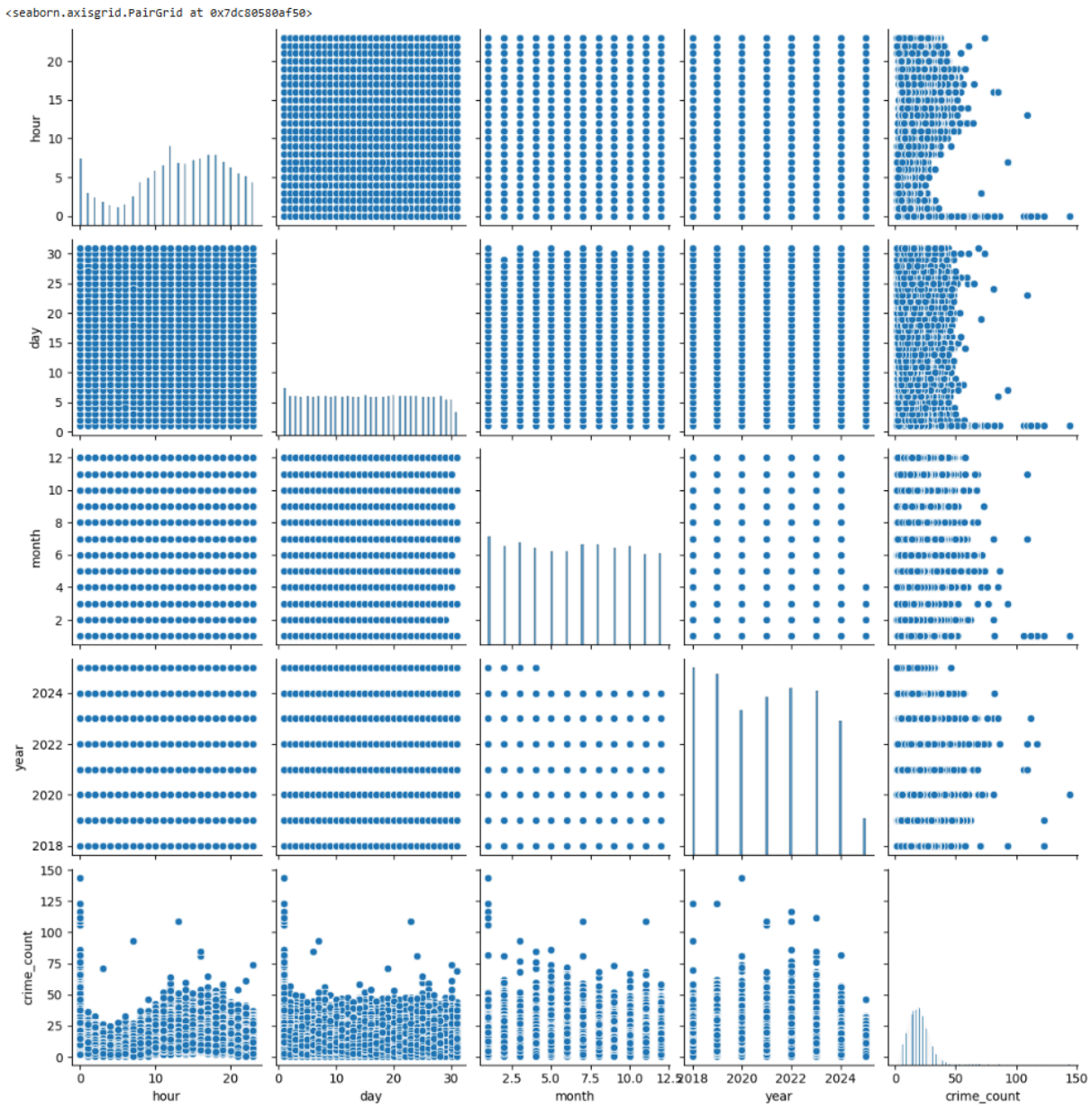
1. Can we predict the category of crime based on time and location?
2. What are the peak hours for different crime types?
3. Do neighborhoods show consistent patterns for specific crimes (e.g., theft vs. assault)?
4. Are there seasonal patterns in crime trends?
5. What are the geospatial hotspots for crime?

3. Organization

This project was organized using a structured workflow within a Jupyter Notebook environment on Google Colab. Python was chosen as the primary programming language due to its flexibility and powerful data science libraries. Data analysis and preprocessing tasks were performed using Pandas and NumPy, while visualizations were created with Matplotlib and Seaborn. For machine learning tasks, the Scikit-learn library was utilized to build preprocessing pipelines and train models such as Linear Regression.

The overall process was divided into several key stages: data loading, data cleaning, feature engineering, model training, and evaluation. The notebook was structured to follow this logical progression, ensuring clarity and reproducibility. Exploratory Data Analysis (EDA) helped uncover important patterns, while feature engineering introduced cyclic time-based variables to better model temporal crime behavior. The modeling section focused on building a regression model to predict hourly crime counts, and performance was assessed using metrics like MAE, RMSE, and R^2.

4. Initial Data Exploration



Pairplot showing distributions and pairwise relationships between time-based features (hour, day, month, year) and the aggregated crime_count for San Francisco incident data.

This figure visualizes the relationships between temporal variables and hourly crime occurrences. From the histograms along the diagonal, we observe that:

Crimes are most frequent between 12 PM and 8 PM, with fewer incidents in early morning hours.

Crime counts are distributed throughout the days of the month, but weekends (not shown directly) correlate with spikes.

Data is well distributed across years from 2018 to 2024, indicating long-term consistency.

The crime_count distribution is right-skewed, meaning a few hours have very high crime volumes, but most are lower.

These insights confirm temporal patterns in criminal activity and support using time features for prediction.

5. Data Preprocessing

```
# # Replace spaces with underscores in all column names
# # Make all column names lowercase and replace spaces/special characters
numerify_crime_df_cleaned.columns = (
        numerify_crime_df_cleaned.columns
        .str.strip()
        .str.lower()
        .str.replace(' ', '_')
        .str.replace(r'[^a-zA-Z0-9_]', '', regex=True)
)
numerify_crime_df_cleaned = numerify_crime_df_cleaned.drop("incident_date", axis=1)
numerify_crime_df_cleaned['year'].value_counts().sort_index()
numerify_crime_df_cleaned.tail()
```

	incident_category	analysis_neighborhood	police_district	incident_day_of_week	incident_datetime	hour	day	month	year
11633	Larceny Theft	Bernal Heights	Ingleside	Monday	2025-04-28 20:59:00	20	28	4	2025
11816	Larceny Theft	Bernal Heights	Ingleside	Monday	2025-04-28 21:05:00	21	28	4	2025
11805	Larceny Theft	Bernal Heights	Ingleside	Monday	2025-04-28 21:06:00	21	28	4	2025
11930	Larceny Theft	South of Market	Tenderloin	Monday	2025-04-28 21:56:00	21	28	4	2025
12103	Non-Criminal	Mission	Mission	Monday	2025-04-28 22:34:00	22	28	4	2025

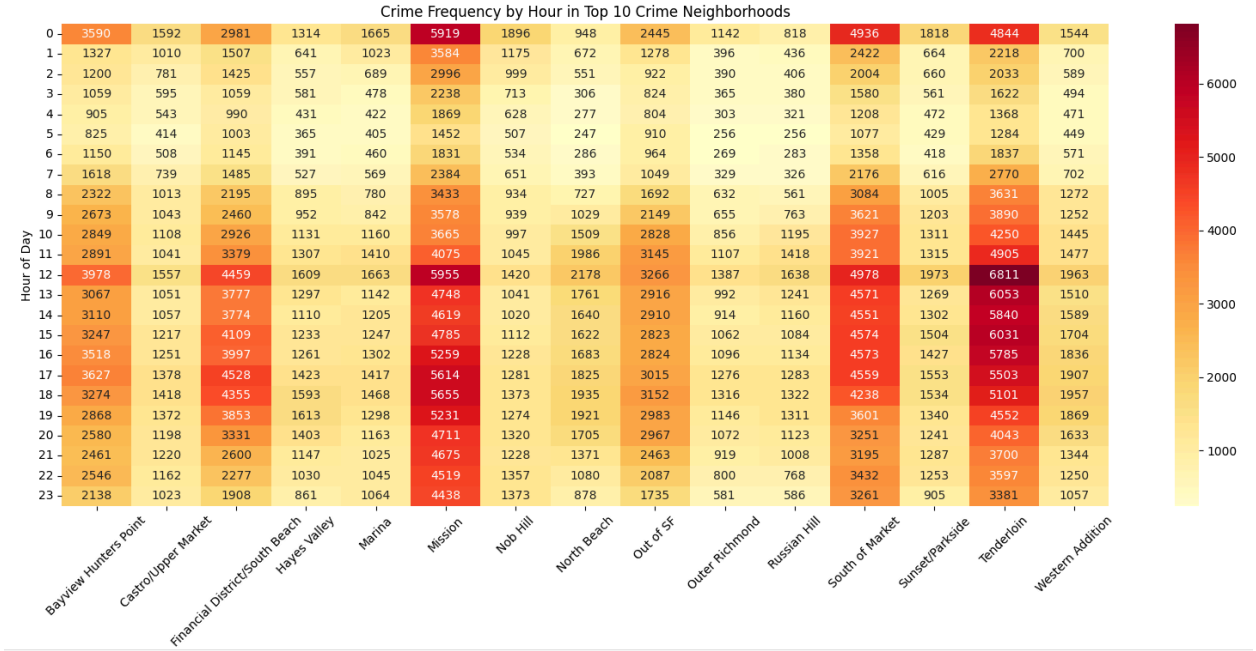
```
[37] # Create proper aggregation for regression task
crime_count_df = numerify_crime_df_cleaned.copy()
crime_counts = numerify_crime_df_cleaned.groupby(['year', 'month', 'day', 'hour']).size().reset_index(name='crime_count')
crime_count_df = crime_count_df.merge(crime_counts, on=['year', 'month', 'day', 'hour'], how='left')
crime_count_df.tail()
```

	incident_category	analysis_neighborhood	police_district	incident_day_of_week	incident_datetime	hour	day	month	year	crime_count
951128	Larceny Theft	Bernal Heights	Ingleside	Monday	2025-04-28 20:59:00	20	28	4	2025	5
951129	Larceny Theft	Bernal Heights	Ingleside	Monday	2025-04-28 21:05:00	21	28	4	2025	3
951130	Larceny Theft	Bernal Heights	Ingleside	Monday	2025-04-28 21:06:00	21	28	4	2025	3
951131	Larceny Theft	South of Market	Tenderloin	Monday	2025-04-28 21:56:00	21	28	4	2025	3
951132	Non-Criminal	Mission	Mission	Monday	2025-04-28 22:34:00	22	28	4	2025	1

The dataset was first cleaned by selecting relevant columns like Incident Category, Incident Date, Incident Time, Police District, and Analysis Neighborhood. Missing values were filled or dropped appropriately. A new column, incident_datetime, was created by combining date and time fields, from which hour, day, month, and year were extracted.

Categorical features such as neighborhood and day of the week were converted using One-Hot Encoding. To handle cyclical patterns in time (like hours repeating daily), hour and day_of_week were encoded with sine and cosine transformations.

Finally, the data was grouped by hour to compute crime_count, the target for the regression model. These preprocessing steps made the data ready for effective machine learning analysis.



6.heatmap is valuable:

- It shows crime frequency patterns by hour across the top 10 neighborhoods
- You can clearly see peak hours (e.g., 12 PM–6 PM), especially in high-crime areas like Tenderloin and Mission

7. Model Evaluation

```
RF MAE: 4.53
RF RMSE: 6.03
RF R²: 0.583
```

We trained three models: Linear Regression, Decision Tree, and Random Forest, all using the same features. Random Forest achieved the best performance with the lowest MAE (4.53), RMSE (6.03), and highest R^2 score (0.583). This shows it was the most accurate at predicting hourly crime counts.

8. Challenges

In the early days, we encountered many difficulties in data cleaning and had to try different methods before finding the right one. We also realized that we had to clearly understand the data first. Later, we successfully designed useful features and applied appropriate encoding to improve the performance of the model. However, we still cannot make very accurate predictions.