SF Crime Forecasting

CS 133 - Data Visualization
Member: Tang, Ho Lam
Ruiyuan Li

2.Project Questions

Project Questions

Key Questions We Explore:

1.Can we accurately predict the category of crime based on time, day of week, and police district?

2.What are the peak hours during which specific crimes occur more frequently?

3.Do different neighborhoods show consistent patterns for crime types like theft, assault, or vandalism?

4.Is there a seasonal pattern in violent vs. non-violent crimes?

5.What are the spatial crime hotspots in San Francisco over the last 5 years?

3.Dataset Overview

Source: San Francisco Police Department Incident Reports (2018–Present)

Columns used:9 key column

incident_category, incident_datetime, analysis_neighborhood, police_district, incident_day_of_week, incident_time, incident_description, longitude, latitude

Total rows: 952,286

```
crime_data_path = "/content/drive/Shareddrives/SF_Crime_Forecasting/Police_Department_Incident_Reports__2018_to_Present_20
crime_df = pd.read_csv(crime_data_path)
crime_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 952287 entries, 0 to 952286
Data columns (total 35 columns):
 #   Column                                         Non-Null Count   Dtype
---  ------                                         --------------   -----
 0   Incident Datetime                              952287 non-null  object
 1   Incident Date                                  952287 non-null  object
 2   Incident Time                                  952287 non-null  object
 3   Incident Year                                  952287 non-null  int64
 4   Incident Day of Week                           952287 non-null  object
 5   Report Datetime                                952287 non-null  object
 6   Row ID                                         952287 non-null  int64
 7   Incident ID                                    952287 non-null  int64
 8   Incident Number                                952287 non-null  int64
 9   CAD Number                                     741646 non-null  float64
 10  Report Type Code                               952287 non-null  object
 11  Report Type Description                        952287 non-null  object
 12  Filed Online                                   185557 non-null  object
 13  Incident Code                                  952287 non-null  int64
 14  Incident Category                              951133 non-null  object
 15  Incident Subcategory                           951133 non-null  object
 16  Incident Description                           952287 non-null  object
 17  Resolution                                     952287 non-null  object
 18  Intersection                                   900362 non-null  object
 19  CNN                                            900362 non-null  float64
 20  Police District                                952287 non-null  object
 21  Analysis Neighborhood                          900119 non-null  object
 22  Supervisor District                            899893 non-null  float64
 23  Supervisor District 2012                       900228 non-null  float64
 24  Latitude                                       900362 non-null  float64
 25  Longitude                                      900362 non-null  float64
 26  Point                                          900362 non-null  object
 27  Neighborhoods                                  888210 non-null  float64
 28  ESNCAG - Boundary File                         10984 non-null   float64
 29  Central Market/Tenderloin Boundary Polygon - Updated  127671 non-null  float64
 30  Civic Center Harm Reduction Project Boundary   124884 non-null  float64
 31  HSOC Zones as of 2018-06-05                    200793 non-null  float64
 32  Invest In Neighborhoods (IIN) Areas            0 non-null       float64
 33  Current Supervisor Districts                   900228 non-null  float64
 34  Current Police Districts                       899378 non-null  float64
dtypes: float64(14), int64(5), object(16)
memory usage: 254.3+ MB
```

4.Data Cleaning & Preparation

Raw sample of selected columns before cleaning

Shows missing values in Analysis Neighborhood and Police District

Format still includes string-type dates and times

Further cleaning needed for nulls and column transformations

```
selected_df.isna().sum()
```

	0
Incident Category	1154
Analysis Neighborhood	52168
Police District	0
Incident Date	0
Incident Day of Week	0
Incident Time	0

dtype: int64

```
# Make a copy of the original DataFrame
selected_df = crime_df.copy()

# Select only the desired columns (excluding Incident Year now)
selected_columns = [
        "Incident Category",
        "Analysis Neighborhood",
        "Police District",
        "Incident Date",
        "Incident Day of Week",
        "Incident Time"
]
selected_df = selected_df[selected_columns]

selected_df.head(10)
```

	Incident Category	Analysis Neighborhood	Police District	Incident Date	Incident Day of Week	Incident Time
0	Larceny Theft	NaN	Mission	2023/03/01	Wednesday	05:02
1	Recovered Vehicle	NaN	Out of SF	2023/03/14	Tuesday	18:44
2	Larceny Theft	NaN	Mission	2023/02/15	Wednesday	03:00
3	Larceny Theft	NaN	Central	2023/03/11	Saturday	15:00
4	Larceny Theft	NaN	Central	2023/03/13	Monday	07:30
5	Drug Violation	NaN	Out of SF	2023/03/16	Thursday	09:26
6	Assault	Potrero Hill	Bayview	2023/03/16	Thursday	17:30
7	Recovered Vehicle	NaN	Out of SF	2023/03/16	Thursday	13:49
8	Larceny Theft	NaN	Richmond	2023/03/16	Thursday	22:15
9	Larceny Theft	NaN	Central	2023/02/11	Saturday	14:00

5.Feature Engineering & Final Dataset

Parsed **`incident_datetime`** into **`hour`**, **`day`**, **`month`**, and **`year`** columns

Removed rare or missing categories and cleaned inconsistent values

Used one-hot encoding for categorical columns like **`neighborhood`**, **`district`**, and **`day_of_week`**

Aggregated incidents by hour using groupby to create the target variable: **`crime_count`**

Final dataset includes both time and location features for modeling crime trends

	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

6.Data Cleaning Confirmation

Confirm data cleanliness and structure.

No missing (null) values in any of the 12 columns — that means the dataset is fully cleaned.

Verifying that the dataset is clean and ready.

```
crime_count_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 951133 entries, 0 to 951132
Data columns (total 12 columns):
 #   Column                      Non-Null Count   Dtype
---  ------                      --------------   -----
 0   incident_category           951133 non-null  object
 1   analysis_neighborhood       951133 non-null  object
 2   police_district             951133 non-null  object
 3   incident_day_of_week        951133 non-null  object
 4   incident_datetime           951133 non-null  datetime64[ns]
 5   hour                        951133 non-null  int32
 6   day                         951133 non-null  int32
 7   month                       951133 non-null  int32
 8   year                        951133 non-null  int32
 9   count_crime_hourly          951133 non-null  int64
 10  count_crime_fulltime_frame  951133 non-null  int64
 11  count_crime_district        951133 non-null  int64
dtypes: datetime64[ns](1), int32(4), int64(3), object(4)
memory usage: 72.6+ MB
```

7.Data Exploration

Understand the diversity of each column's values — helpful before encoding or analysis.

Whether any column has too many or too few values.

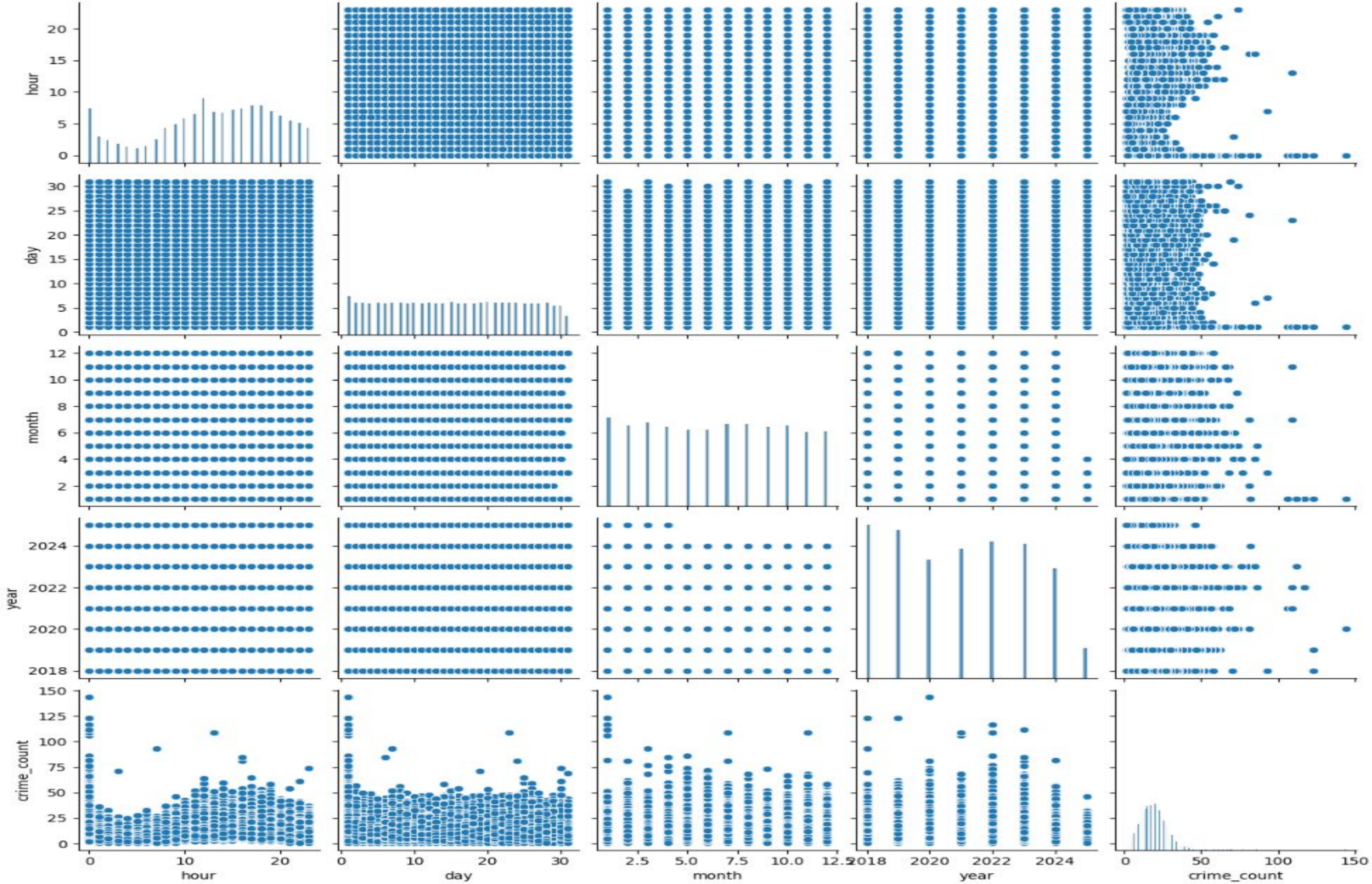
Next is Seaborn pairplot, and it's perfect for showing relationships between time-based features and the crime_count variable.

```
crime_count_df.nunique()
```

	0
incident_category	49
analysis_neighborhood	42
police_district	11
incident_day_of_week	7
incident_datetime	454889
hour	24
day	31
month	12
year	8
count_crime_hourly	24
count_crime_fulltime_frame	86
count_crime_district	194

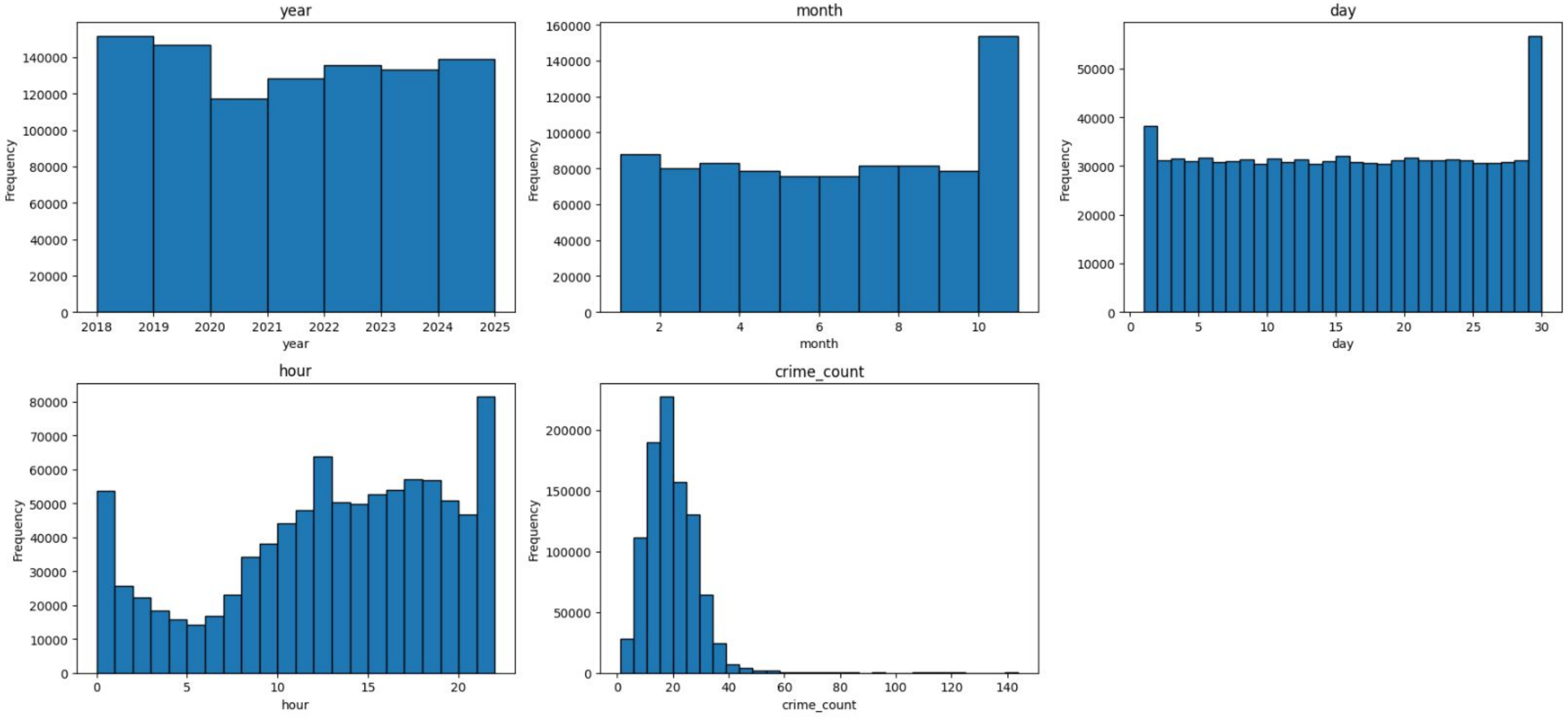
dtype: int64

<seaborn.axisgrid.PairGrid at 0x7dc7a1ef1e10>

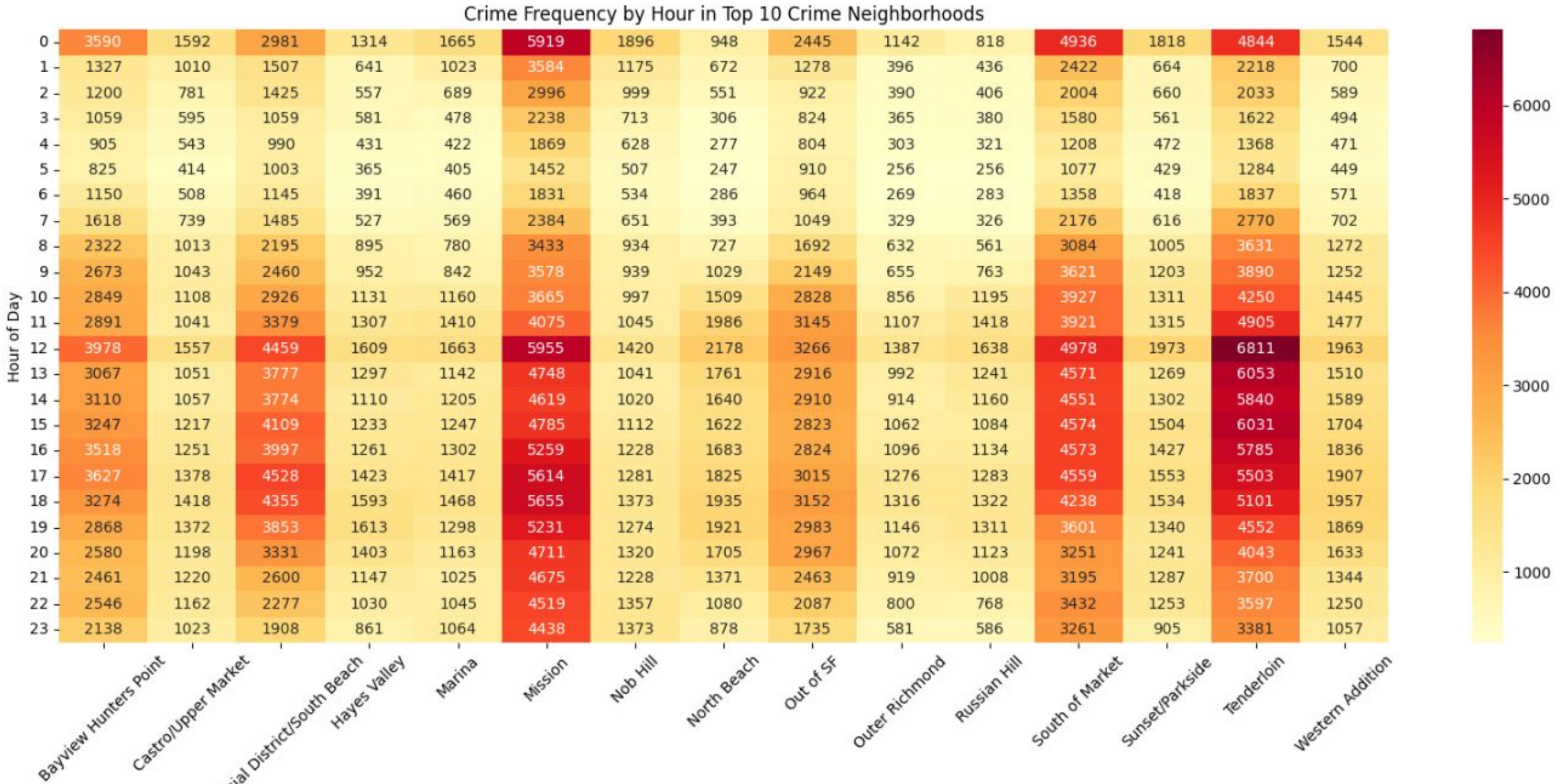


8.Hourly Crime Trends & Feature Distributions

This exploration helped validate hour, day, month, and year as meaningful features for modeling



“Crime Frequency by Hour in Top 10 Crime Neighborhoods”



9.Testing & Result

We have three models

Linear Regression,Decision Tree,and Random Forest

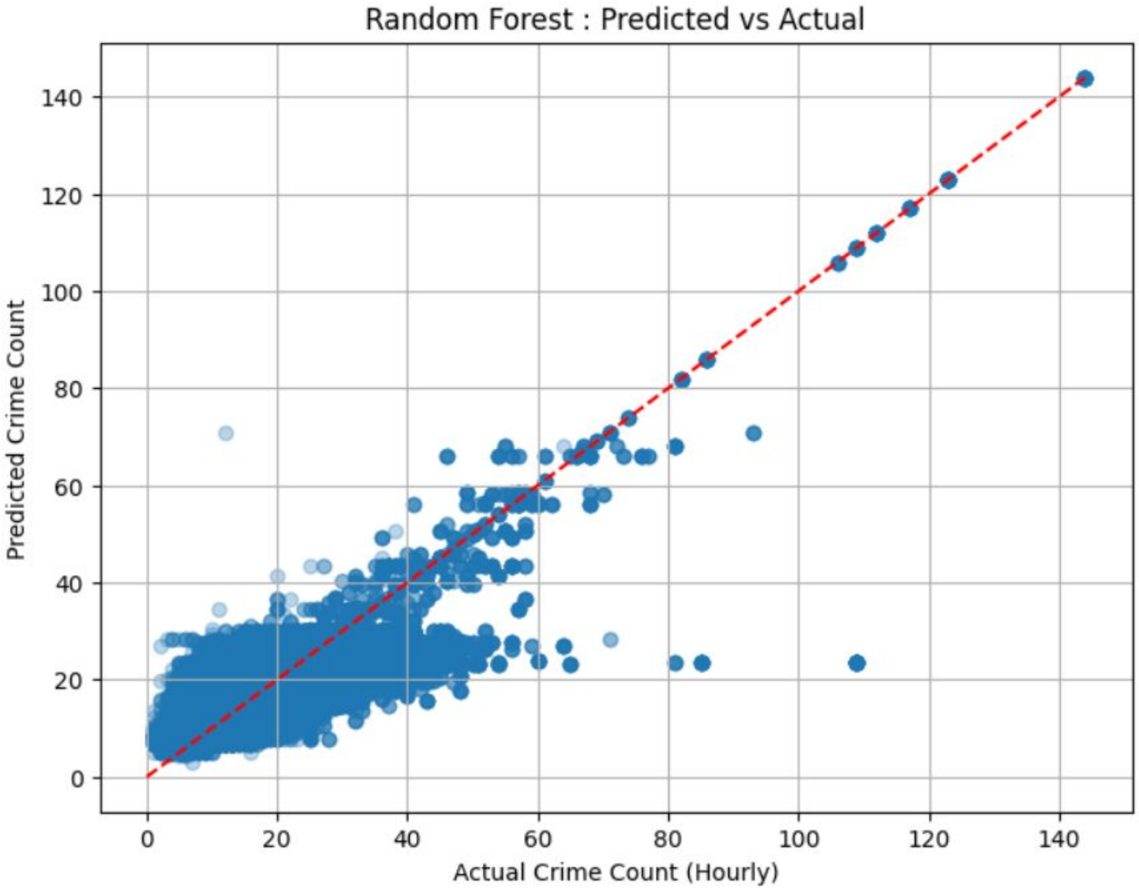
```
RF MAE: 6.43
RF RMSE: 8.95
RF R²: 0.075
```

```
RF MAE: 4.58
RF RMSE: 6.10
RF R²: 0.573
```

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

All models predicted the same target (hourly crime count) using identical time and location features. Among them, Random Forest is the best with the lowest error."

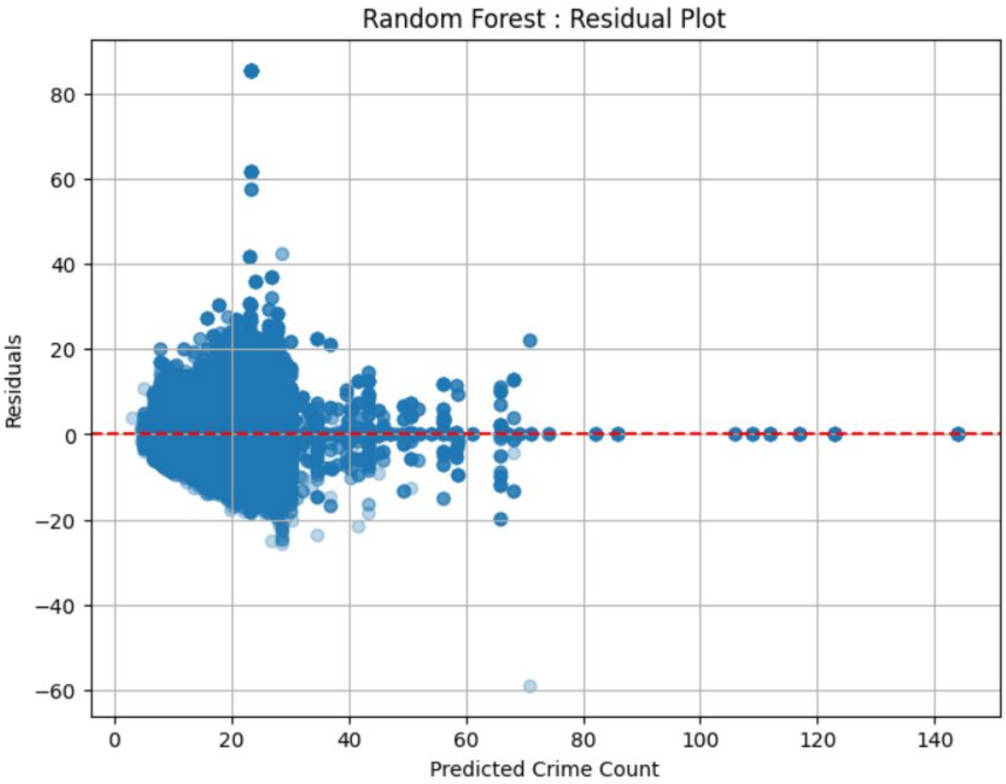
Testing & Result



Key Point

- Most hourly crime predictions are **very close to the real numbers** (on the diagonal line)
- When crime is **low**, the model sometimes guesses a little too high
- When crime is **high**, the model usually guesses a bit too low
- Only a few points are **far off**, meaning big mistakes are rare

Testing & Result



Key Point

- Most prediction errors are close to **0**, which means the model is mostly accurate
- When the predicted crime count is **low**, the errors are more spread out
- When the predicted crime count is **high**, the errors are smaller and more steady
- There are a few unusual points (outliers), but the model does **not make big mistakes often**

10.Challenges

At first, we were confused about how to clean the data and tried different ways before finding the right method

Some columns like neighborhood and district had missing values, so we filled them with “Out of SF”

If we understand the data better at the start, we can save time and avoid doing things over