SUMMER TRAINING/INTERNSHIP

PROJECT REPORT

(Term June-July 2025)

911 Emergency Calls Exploratory Analysis and Incident pattern Detection

Submitted by

Deepak, Vishnu, Sourav, Soam Parkash, Somu akash Reddy
Registration Number: 12316722, 12307182, 12326825, 12301024, 12313182

Course Code : CSE443

Under the Guidance of

Dr. Sandeep Kaur

School of Computer Science and Engineering

Acknowledgement

We would like to express our sincere gratitude to our mentor for continuous guidance and support. We also thank the School of Computer Science and Engineering for providing the opportunity to undertake this project. Our collaboration as a group enhanced our learning and understanding.

Date: 14-07-2025

Table of Contents

Introduction

Company Profile :

The project was conducted under the academic guidance of the School of Computer Science and Engineering as part of the summer internship program. It aimed to provide practical exposure in the domain of data science and analytics.

Overview of Training Domain :

The domain of this project is Data Science, which focuses on extracting meaningful insights from large and complex datasets. Specifically, this training involved the analysis of emergency call data using techniques from Exploratory Data Analysis (EDA) and Machine Learning (ML) implemented in Python. Furthermore, Power BI was used to design interactive dashboards that visually represent patterns in the data, enabling better understanding and decision-making.

Objective of the Project :

The key objectives of this group project are:

This project is aimed at leveraging the power of data science to analyze 911 emergency call records with the goal of deriving meaningful insights that can assist public safety services in decision-making. By combining statistical testing, machine learning techniques, and interactive visualization tools, the project intends to offer a comprehensive view of emergency patterns across locations and time periods. It also emphasizes collaborative learning, enabling the group to work through an end-to-end data analysis pipeline—from data cleaning to predictive modeling and dashboard reporting.

- To perform a detailed exploratory data analysis (EDA) of the 911 emergency call dataset to uncover trends, patterns, and anomalies in the data.
- To identify the distribution and frequency of emergency call reasons such as EMS, Fire, and Traffic across various townships, zip codes, and time frames (hours, days, months).
- To conduct statistical hypothesis testing using T-test, F-test, and Chi-square test to validate assumptions and understand the relationships between variables.

- To build and train machine learning classification models (Logistic Regression, Random Forest, SVC) for predicting the type of emergency call based on features such as time, location, and other contextual data.
- To evaluate and compare the performance of different machine learning models based on accuracy and classification metrics.
- To create an interactive dashboard using Power BI for visual representation of key findings and insights, allowing stakeholders to interact with and analyze the data efficiently.
- To gain hands-on experience in a complete data science pipeline including data preprocessing, analysis, visualization, model building, and report generation.
- To work collaboratively in a team environment, enhancing project planning, task distribution, and communication skills throughout the internship period.

Training Overview

Tools & Technologies Used :

This project relied on a set of essential tools and technologies from the data science ecosystem. Python was used as the core programming language due to its extensive support for data manipulation and machine learning. Libraries like Pandas, NumPy, Matplotlib, Seaborn, and Scikit-learn enabled structured analysis, statistical evaluation, and visualization. Power BI, a Microsoft visualization tool, was used to create dynamic dashboards that summarized the insights from the data. Statistical computations were enhanced using SciPy, while development was managed through Jupyter Notebook and VS Code for seamless scripting and collaboration.

- Programming Language: Python (Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn)
- Visualization Tool: Power BI
- IDE: Jupyter Notebook / VS Code
- Statistical Libraries: SciPy
- Data Source: 911 emergency call dataset

Dataset link : https://www.kaggle.com/datasets/mchirico/montcoalert

Areas Covered During Training :

The training covered a comprehensive set of topics essential for any data science project. It began with data preprocessing, where raw 911 emergency call data was cleaned and transformed to make it suitable for analysis. Exploratory Data Analysis (EDA) followed, using Python libraries to generate visual insights into call frequencies, time-based trends, and location patterns. Statistical testing, including T-tests, F-tests, and Chi-square tests, helped validate assumptions and uncover relationships between variables. The group also implemented supervised machine learning models such as Logistic Regression, Random Forest, and SVC to classify the type of emergency calls. Lastly, an interactive Power BI dashboard was created to present findings visually and effectively to a non-technical audience.

- Data preprocessing and cleaning
- Exploratory data analysis and visualization
- Statistical testing and inference
- Supervised machine learning (classification models)
- Dashboard creation for business intelligence

Daily/Weekly Work Summary :

Throughout the training period, the project was executed in a structured weekly format. Each week focused on a different phase of the data science workflow. In the first week, the group concentrated on understanding the dataset and performing necessary cleaning and formatting tasks to ensure quality data. The second week involved in-depth exploratory analysis to discover hidden patterns using visual techniques. During the third week, statistical testing and machine learning model development were carried out, focusing on training and evaluating various classifiers. The final week was dedicated to creating a professional Power BI dashboard and compiling the documentation for the project report.

- **Week 1:** Data collection, loading, and cleaning
- **Week 2:** Exploratory analysis and visual pattern discovery
- **Week 3:** Statistical hypothesis testing and ML model development
- **Week 4:** Power BI dashboard creation and report preparation

Project Details

Title of the Project :

911 Emergency Call Analysis

Objective of the Project :

Emergency response teams rely heavily on 911 calls to assess and react to critical situations such as medical emergencies, traffic accidents, and fire outbreaks. However, the sheer volume of calls makes it challenging to allocate resources efficiently and anticipate demand. This project seeks to address that challenge by analyzing a real-world 911 emergency call dataset. Through exploratory data analysis, statistical testing, and machine learning, the goal is to uncover meaningful patterns in call volume, timing, and reasons. These insights can help public safety agencies optimize response strategies, forecast emergency trends, and better understand the geographic and temporal dynamics of emergency situations.

Scope and Objectives :

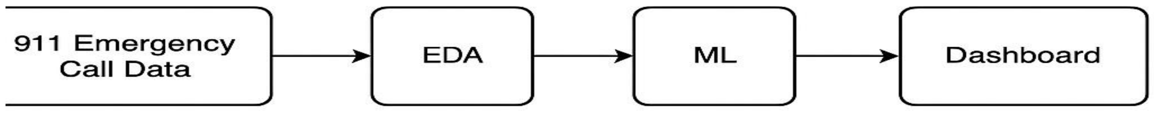
The scope of this project encompasses the full lifecycle of data science processes applied to emergency call data, from raw data processing to model deployment and interactive reporting. By focusing on the analysis of 911 emergency calls, the project aims to improve understanding of public safety service demands and contribute to more efficient emergency response strategies. It explores multiple aspects of the dataset, including geographic, temporal, and categorical patterns, and applies machine learning techniques to classify call types. The project is intended to be both practical and educational, offering hands-on experience in real-world data analysis while delivering insights valuable to emergency management systems.

- Identify most frequent emergency reasons and peak hours.
- Analyze patterns based on zip codes, townships, and days of the week.
- Build machine learning models to classify emergency reasons.
- Develop an interactive dashboard to present data findings.

System Requirements :

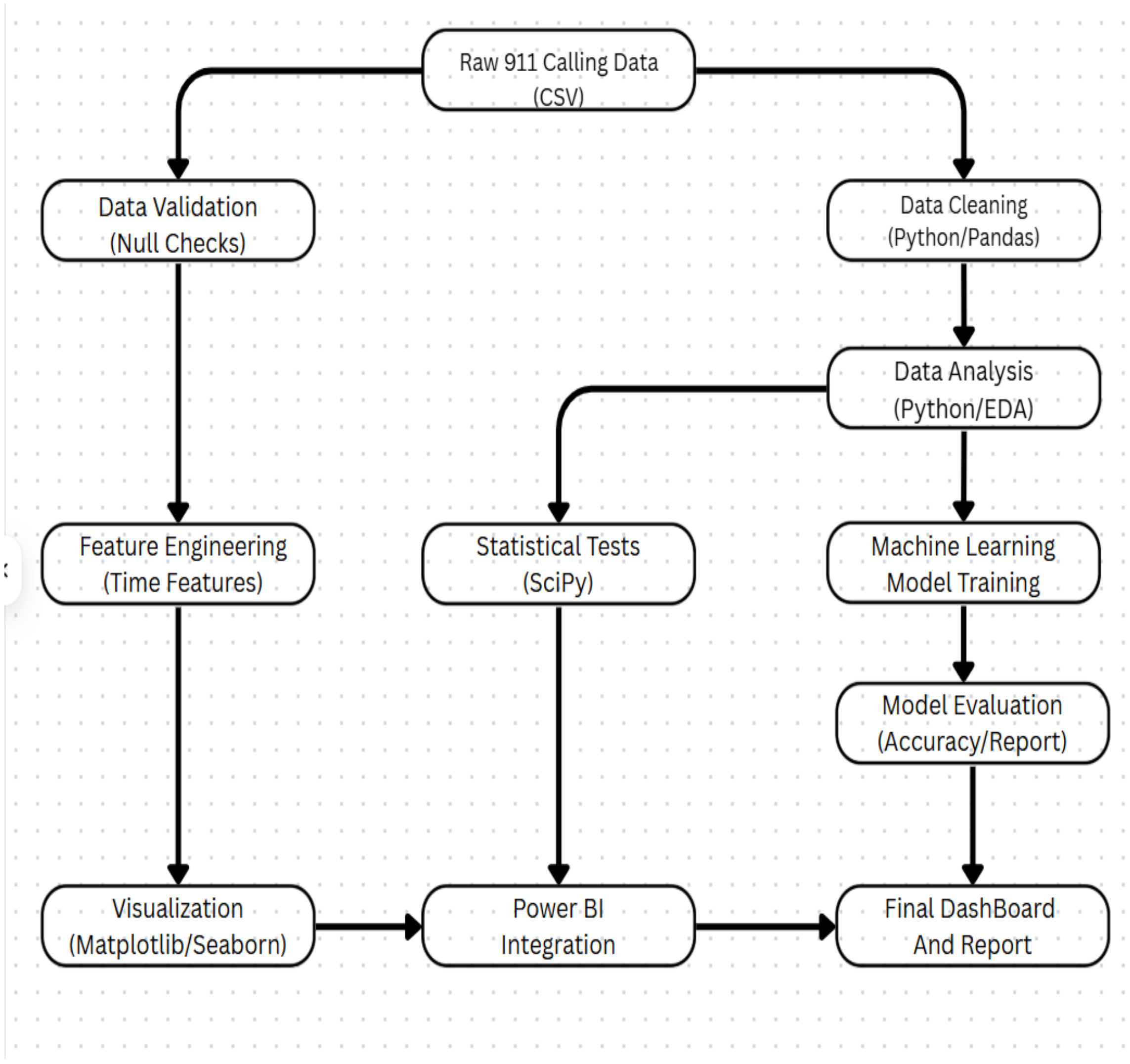
- OS: Windows 10 or above
- RAM: Minimum 8 GB
- Python 3.x with required libraries
- Power BI Desktop

Architecture Diagram :



Architecture Diagram

Data Flow Digram :


Raw 911 Calling Data (CSV)
Data Validation (Null Checks)
Data Cleaning (Python/Pandas)
Data Analysis (Python/EDA)
Feature Engineering (Time Features)
Statistical Tests (SciPy)
Machine Learning Model Training
Model Evaluation (Accuracy/Report)
Visualization (Matplotlib/Seaborn)
Power BI Integration
Final DashBoard And Report

Implementation

Tools Used :

This project made use of two major tools to accomplish its goals efficiently. **Python** was employed for all aspects of backend processing including data cleaning, exploration, statistical testing, and building machine learning models. Its extensive ecosystem of libraries made it suitable for both analytical and predictive tasks. **Power BI**, on the other hand, was used to design an intuitive and interactive dashboard for visualizing results. This helped convert complex insights into easy-to-understand visual summaries for stakeholders.

- Python for data processing, EDA, statistical analysis, and ML.
- Power BI for visualization and dashboard development

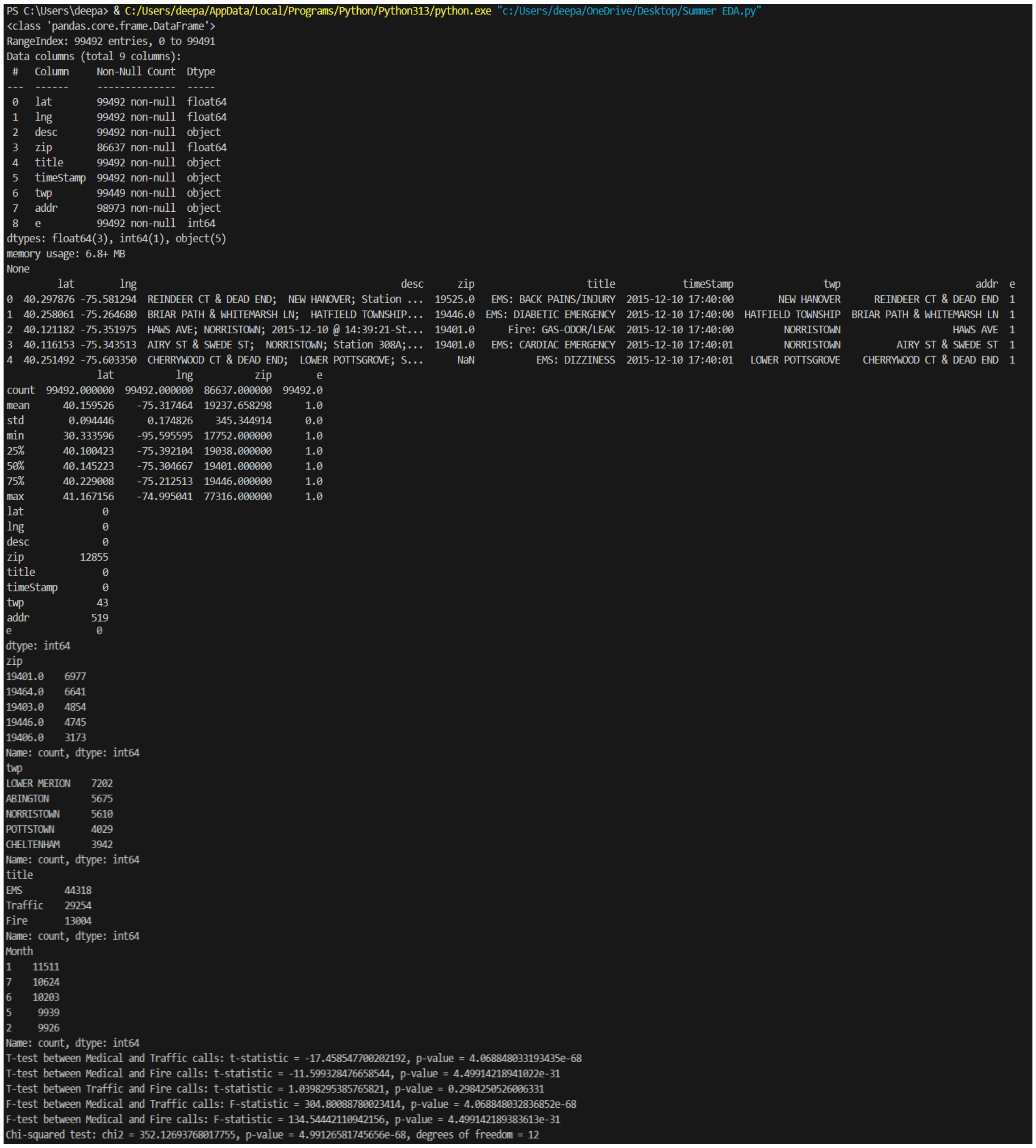
Methodology :

The project followed a structured data science methodology that ensured a smooth transition from raw data to actionable insights. Initially, the dataset was loaded and cleaned by removing null values to ensure data integrity. Exploratory Data Analysis (EDA) was performed to understand the distribution and trends within the data, particularly focusing on call reasons, time of day, and location. Statistical testing using T-test, F-test, and Chi-square tests was conducted to verify assumptions and assess relationships between categorical and numerical variables. The cleaned and enriched dataset was then used to train classification models such as Logistic Regression, Random Forest, and Support Vector Classifier (SVC). These models were evaluated for their prediction accuracy using test data. Finally, an interactive Power BI dashboard was created to visually present the key insights, allowing for dynamic exploration of the findings.

1. Load the dataset and remove null values.
2. Perform descriptive statistics and exploratory visualization.
3. Apply T-test, F-test, and Chi-square tests to validate hypotheses.
4. Encode categorical variables and train classification models (Logistic Regression, Random Forest, SVC).
5. Evaluate models and compare accuracy.
6. Visualize insights using Power BI dashboard.

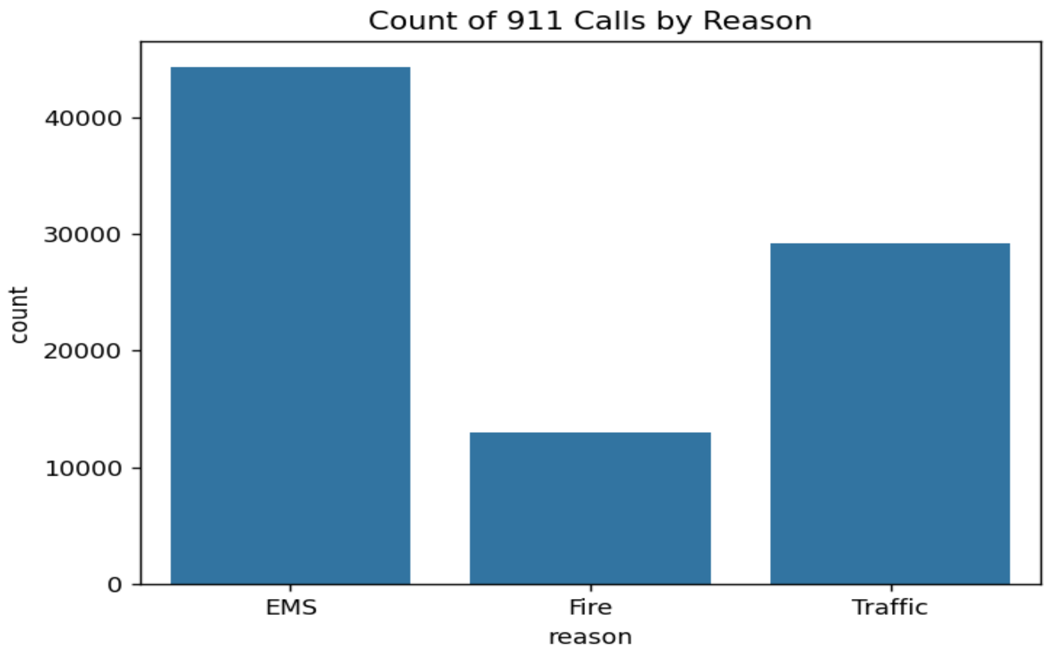
Modules / Screenshots :

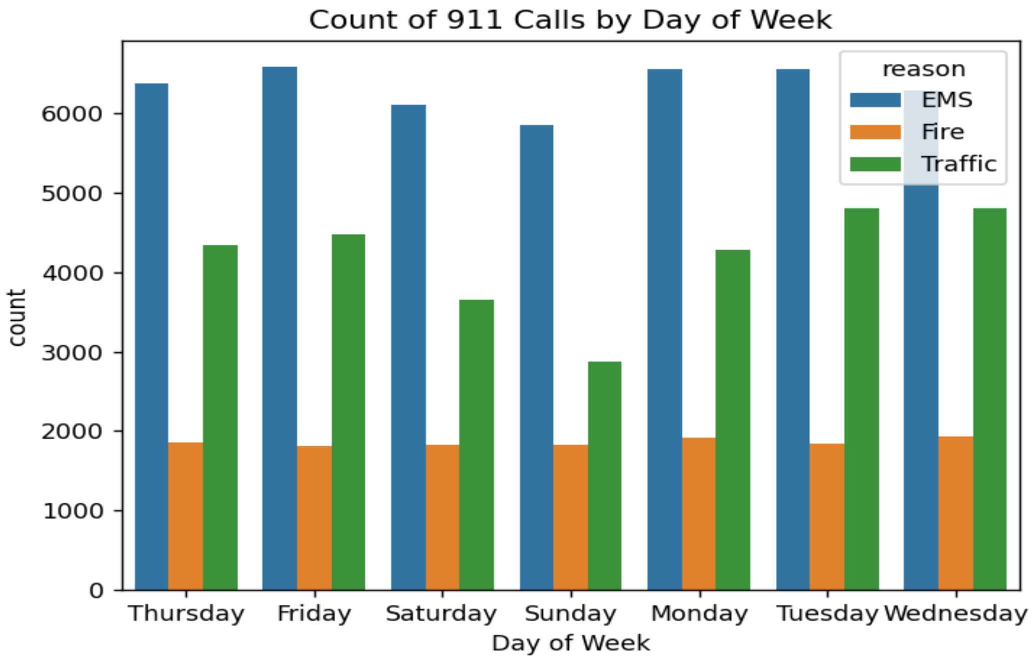
Data Preprocessing :

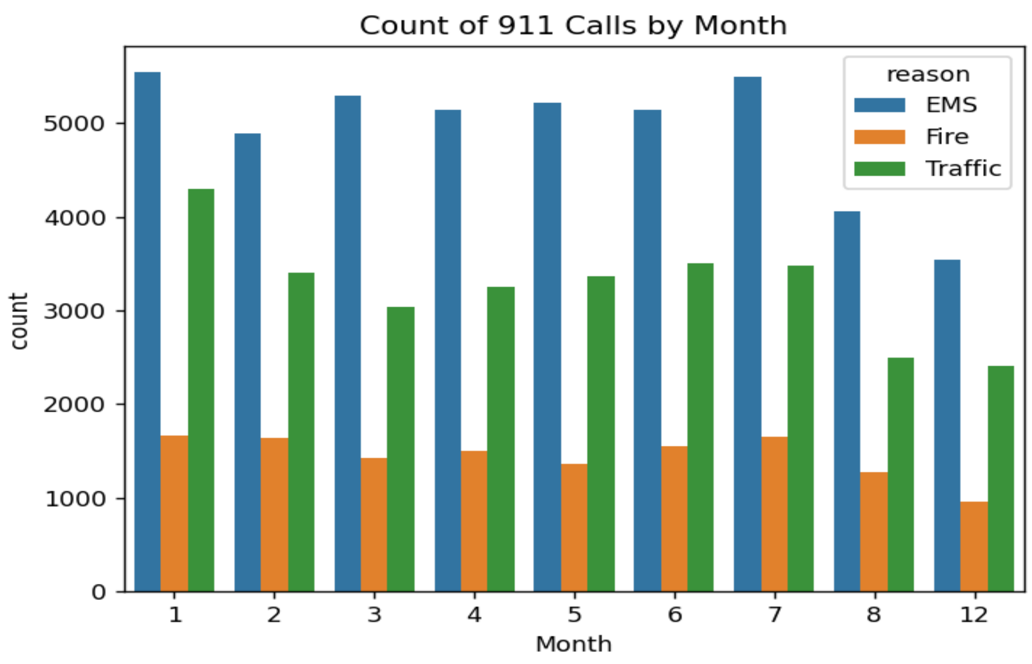


```
PS C:\Users\deepa> & C:/Users/deepa/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/deepa/OneDrive/Desktop/Summer EDA.py"
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99492 entries, 0 to 99491
Data columns (total 9 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   lat        99492 non-null  float64
 1   lng        99492 non-null  float64
 2   desc       99492 non-null  object
 3   zip        86637 non-null  float64
 4   title      99492 non-null  object
 5   timeStamp  99492 non-null  object
 6   twp        99449 non-null  object
 7   addr       98973 non-null  object
 8   e          99492 non-null  int64
dtypes: float64(3), int64(1), object(5)
memory usage: 6.8+ MB
None
         lat        lng                                               desc      zip                    title            timeStamp                twp                        addr  e
0  40.297876 -75.581294  REINDEER CT & DEAD END;  NEW HANOVER; Station ...  19525.0   EMS: BACK PAINS/INJURY  2015-12-10 17:40:00        NEW HANOVER      REINDEER CT & DEAD END  1
1  40.258061 -75.264680  BRIAR PATH & WHITEMARSH LN;  HATFIELD TOWNSHIP...  19446.0  EMS: DIABETIC EMERGENCY  2015-12-10 17:40:00  HATFIELD TOWNSHIP  BRIAR PATH & WHITEMARSH LN  1
2  40.121182 -75.351975  HAWS AVE; NORRISTOWN; 2015-12-10 @ 14:39:21-St...  19401.0      Fire: GAS-ODOR/LEAK  2015-12-10 17:40:00         NORRISTOWN                    HAWS AVE  1
3  40.116153 -75.343513  AIRY ST & SWEDE ST;  NORRISTOWN; Station 308A;...  19401.0   EMS: CARDIAC EMERGENCY  2015-12-10 17:40:01         NORRISTOWN          AIRY ST & SWEDE ST  1
4  40.251492 -75.603350  CHERRYWOOD CT & DEAD END;  LOWER POTTSGROVE; S...      NaN           EMS: DIZZINESS  2015-12-10 17:40:01   LOWER POTTSGROVE    CHERRYWOOD CT & DEAD END  1
                lat           lng           zip        e
count  99492.000000  99492.000000  86637.000000  99492.0
mean      40.159526    -75.317464  19237.658298      1.0
std        0.094446      0.174826    345.344914      0.0
min       30.333596    -95.595595  17752.000000      1.0
25%       40.100423    -75.392104  19038.000000      1.0
50%       40.145223    -75.304667  19401.000000      1.0
75%       40.229008    -75.212513  19446.000000      1.0
max       41.167156    -74.995041  77316.000000      1.0
lat              0
lng              0
desc             0
zip          12855
title            0
timeStamp        0
twp             43
addr           519
e                0
dtype: int64
zip
19401.0    6977
19464.0    6641
19403.0    4854
19446.0    4745
19406.0    3173
Name: count, dtype: int64
twp
LOWER MERION    7202
ABINGTON        5675
NORRISTOWN      5610
POTTSTOWN       4029
CHELTENHAM      3942
Name: count, dtype: int64
title
EMS        44318
Traffic    29254
Fire       13004
Name: count, dtype: int64
Month
1    11511
7    10624
6    10203
5     9939
2     9926
Name: count, dtype: int64
T-test between Medical and Traffic calls: t-statistic = -17.458547700202192, p-value = 4.068848033193435e-68
T-test between Medical and Fire calls: t-statistic = -11.599328476658544, p-value = 4.49914218941022e-31
T-test between Traffic and Fire calls: t-statistic = 1.0398295385765821, p-value = 0.2984250526006331
F-test between Medical and Traffic calls: F-statistic = 304.80088780023414, p-value = 4.068848032836852e-68
F-test between Medical and Fire calls: F-statistic = 134.54442110942156, p-value = 4.499142189383613e-31
Chi-squared test: chi2 = 352.12693768017755, p-value = 4.99126581745656e-68, degrees of freedom = 12
```

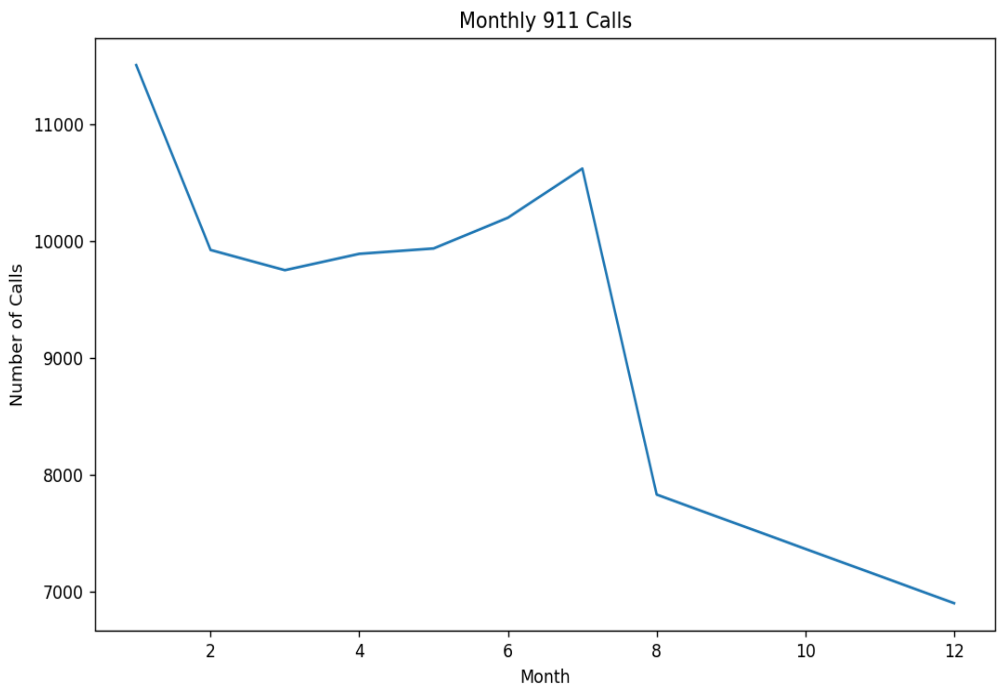
Bar Chart :

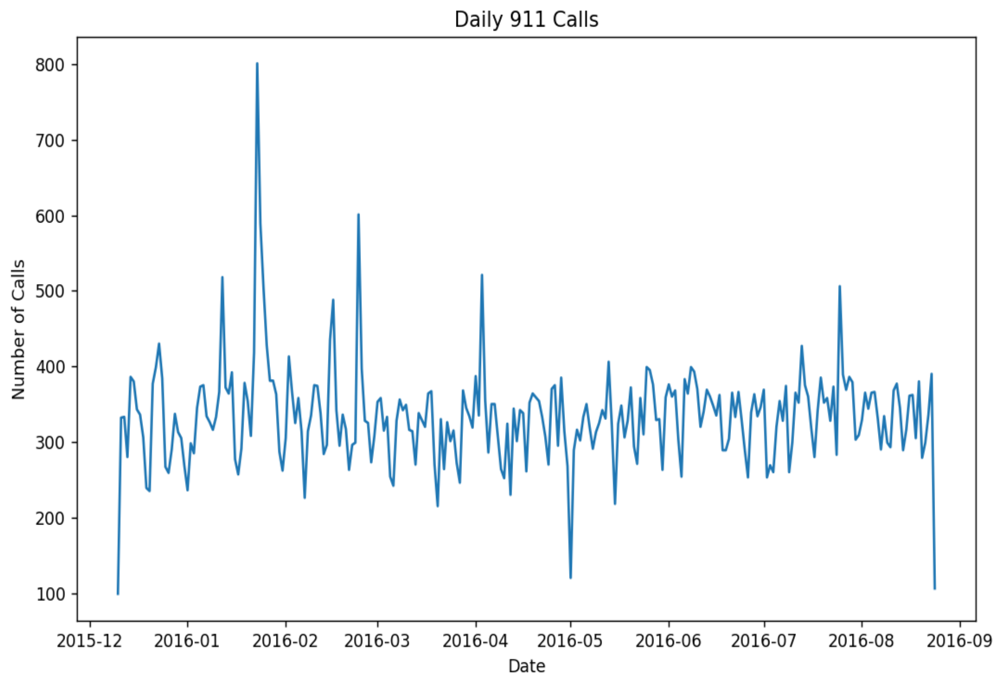

Count of 911 Calls by Reason
count
40000
30000
20000
10000
0
EMS
Fire
Traffic
reason

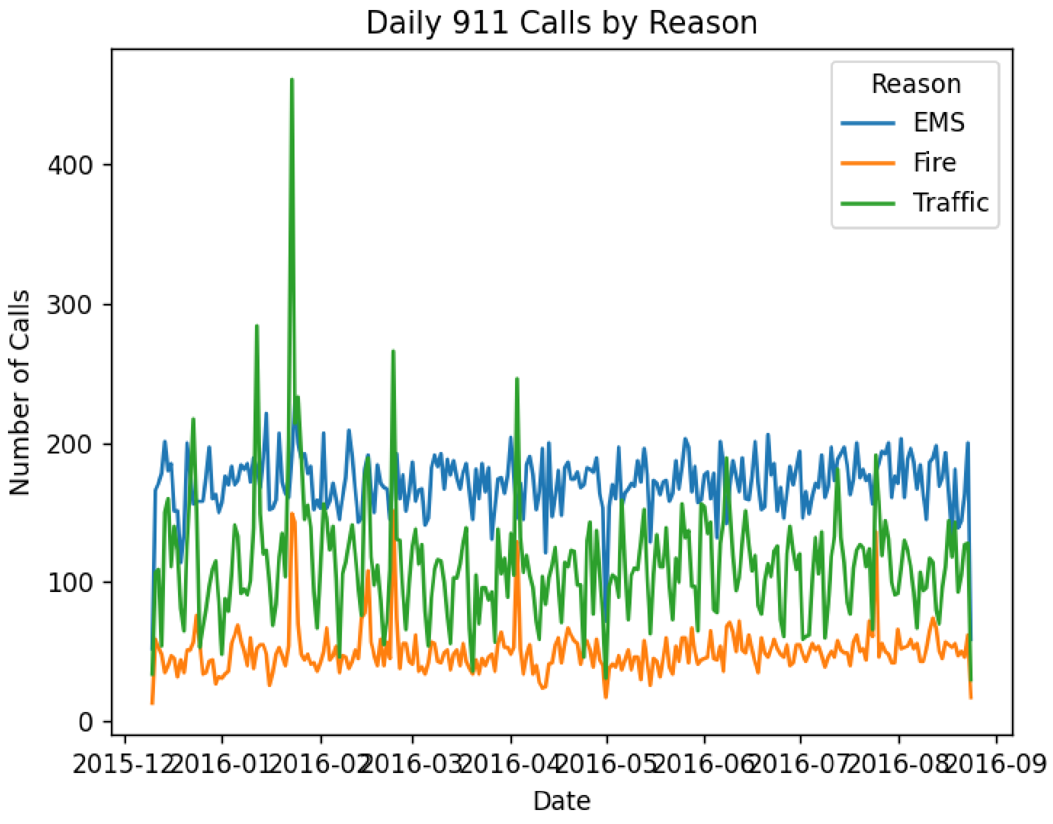

Count of 911 Calls by Day of Week
count
6000
5000
4000
3000
2000
1000
0
Thursday
Friday
Saturday
Sunday
Monday
Tuesday
Wednesday
Day of Week
reason
EMS
Fire
Traffic


Count of 911 Calls by Month
reason
EMS
Fire
Traffic
count
0
1000
2000
3000
4000
5000
1
2
3
4
5
6
7
8
12
Month

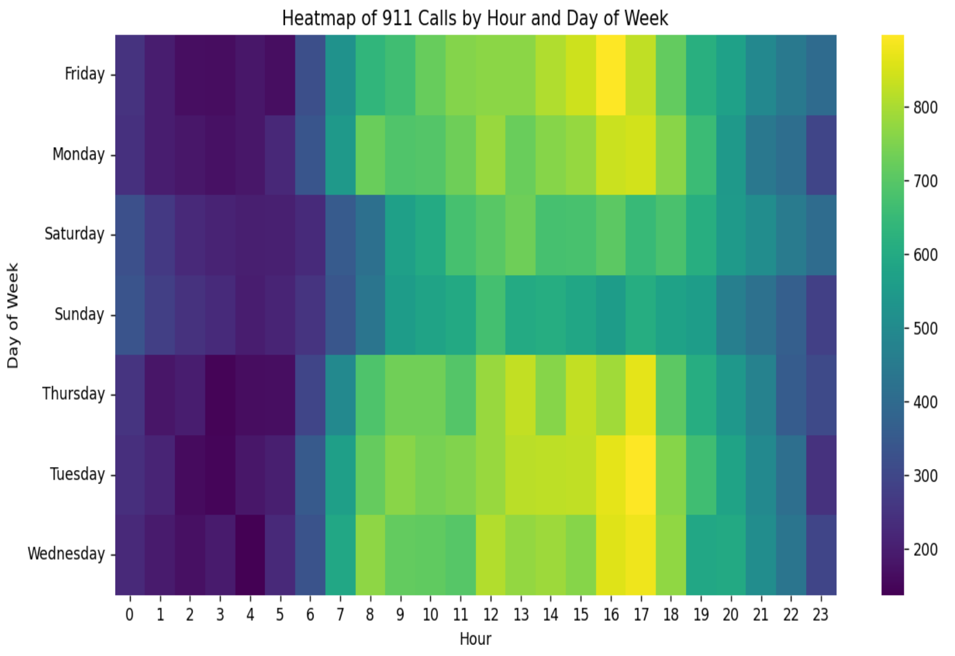
Line Chart :


Monthly 911 Calls
Number of Calls
7000
8000
9000
10000
11000
2
4
6
8
10
12
Month

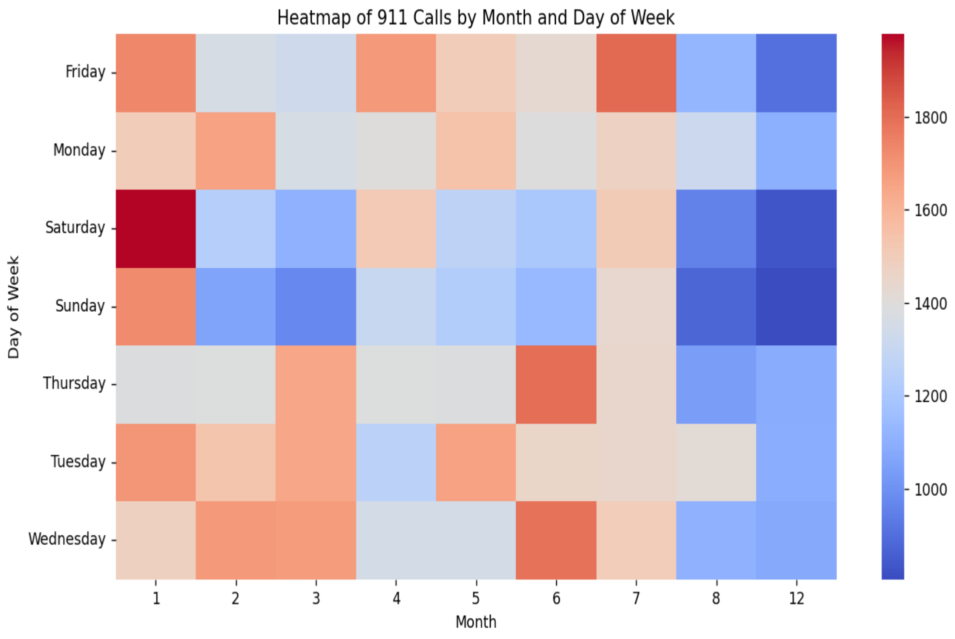

Daily 911 Calls
Number of Calls
800
700
600
500
400
300
200
100
2015-12
2016-01
2016-02
2016-03
2016-04
2016-05
2016-06
2016-07
2016-08
2016-09
Date


Daily 911 Calls by Reason
Reason
EMS
Fire
Traffic
Number of Calls
400
300
200
100
0
2015-12
2016-01
2016-02
2016-03
2016-04
2016-05
2016-06
2016-07
2016-08
2016-09
Date

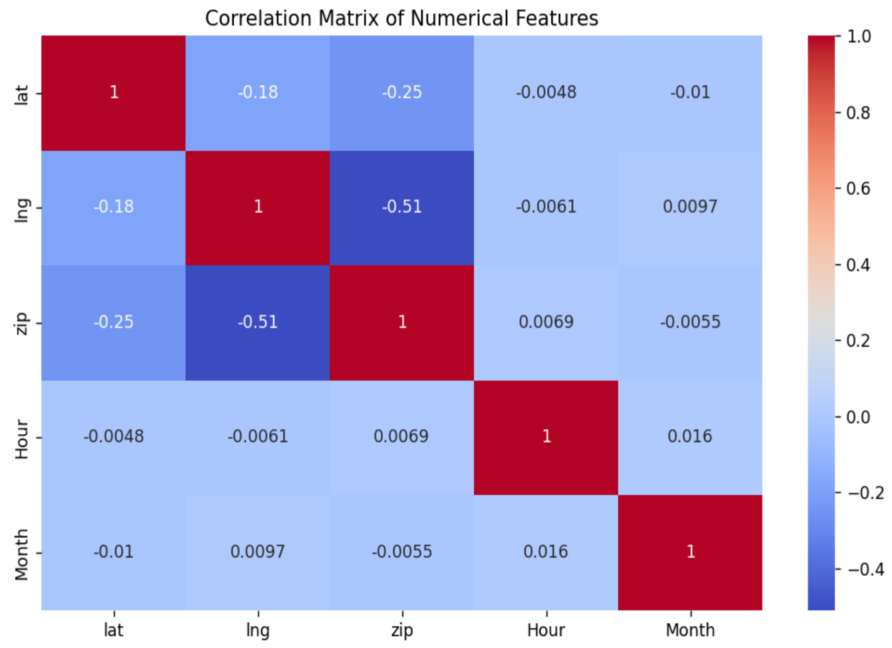
HeatMap :



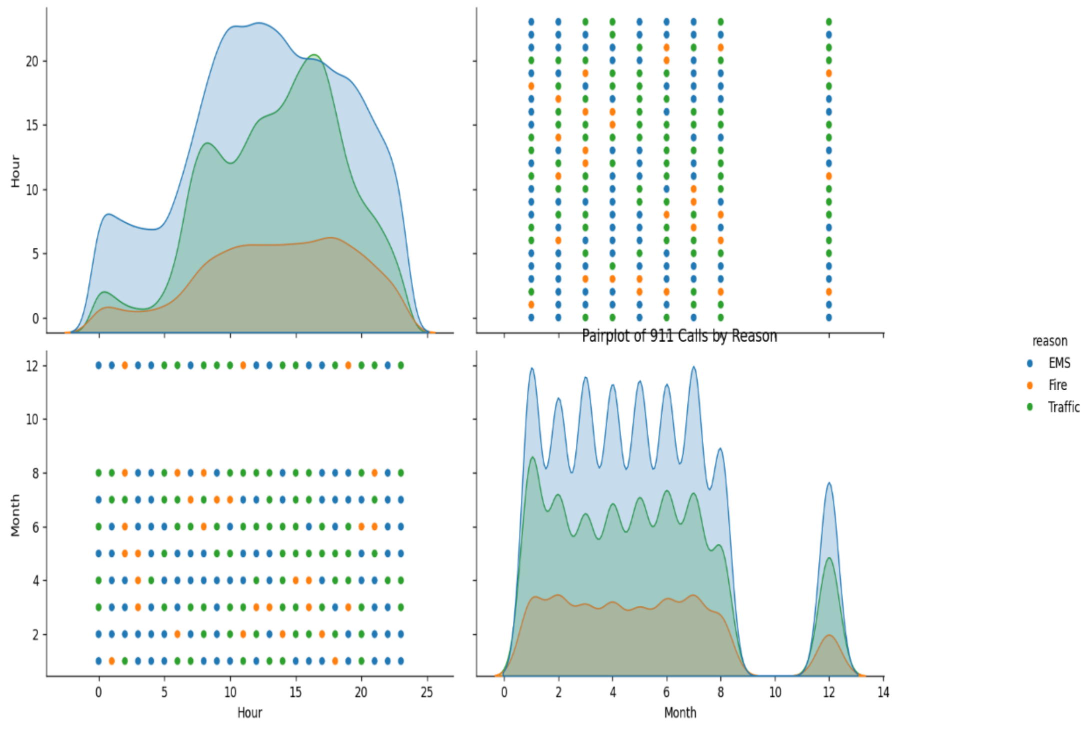
Heatmap of 911 Calls by Hour and Day of Week
Day of Week
Friday
Monday
Saturday
Sunday
Thursday
Tuesday
Wednesday
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
Hour
800
700
600
500
400
300
200



Heatmap of 911 Calls by Month and Day of Week
Day of Week
Friday
Monday
Saturday
Sunday
Thursday
Tuesday
Wednesday
1
2
3
4
5
6
7
8
12
Month
1800
1600
1400
1200
1000

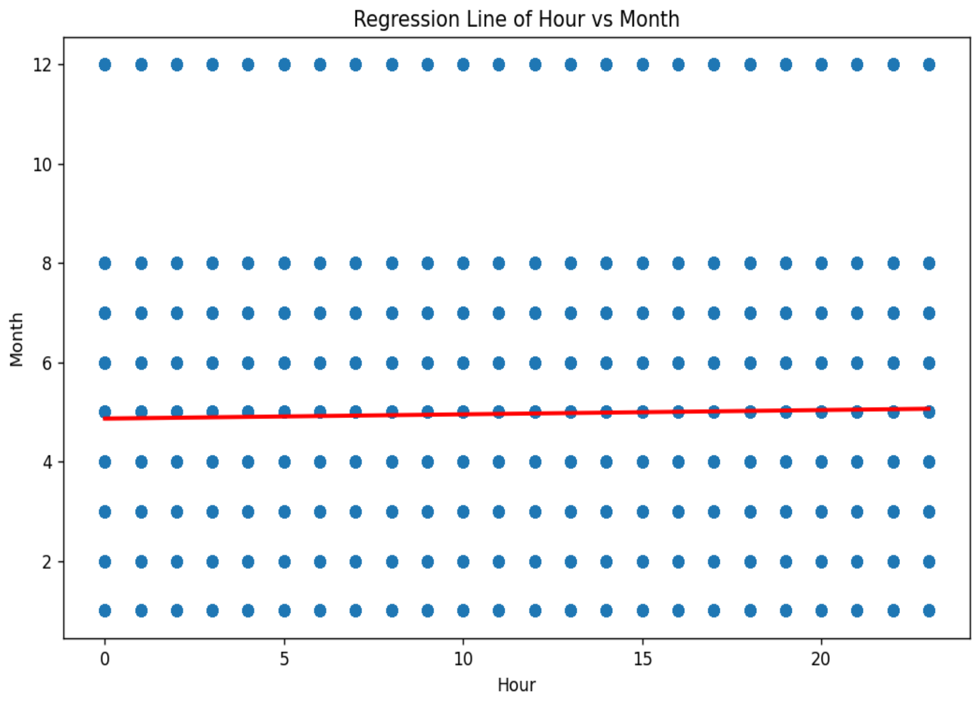

Correlation Matrix of Numerical Features
lat
lng
zip
Hour
Month
1
-0.18
-0.25
-0.0048
-0.01
-0.18
1
-0.51
-0.0061
0.0097
-0.25
-0.51
1
0.0069
-0.0055
-0.0048
-0.0061
0.0069
1
0.016
-0.01
0.0097
-0.0055
0.016
1
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

Pair Plot:

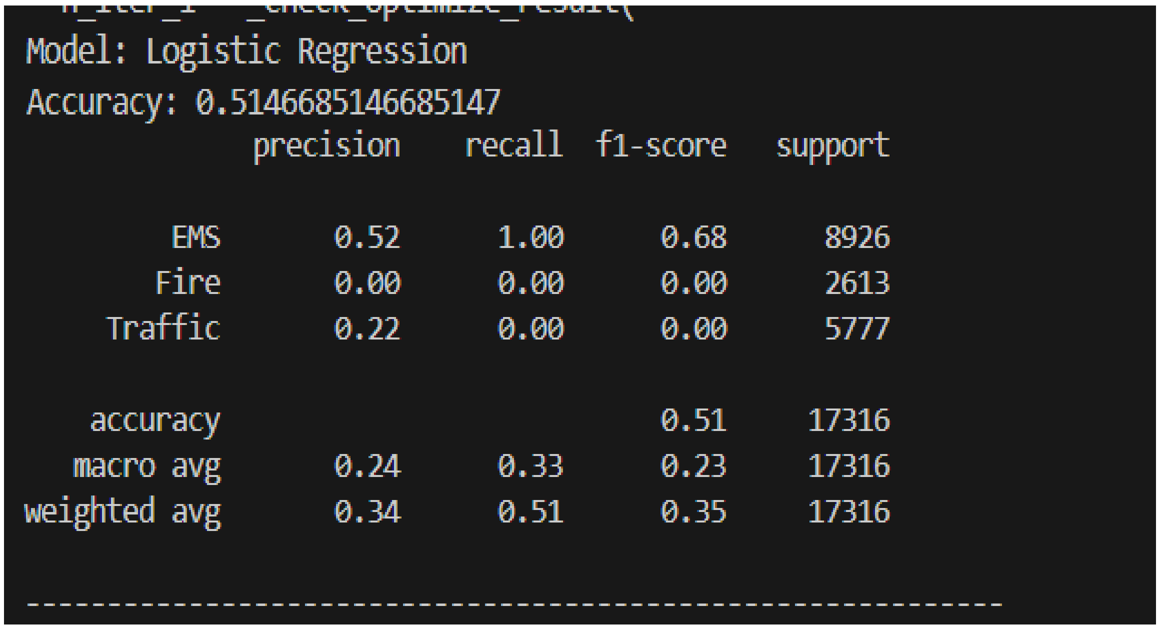

Pairplot of 911 Calls by Reason
Hour
Month
reason
EMS
Fire
Traffic

Machine-Learning Model :

- Regression Line :

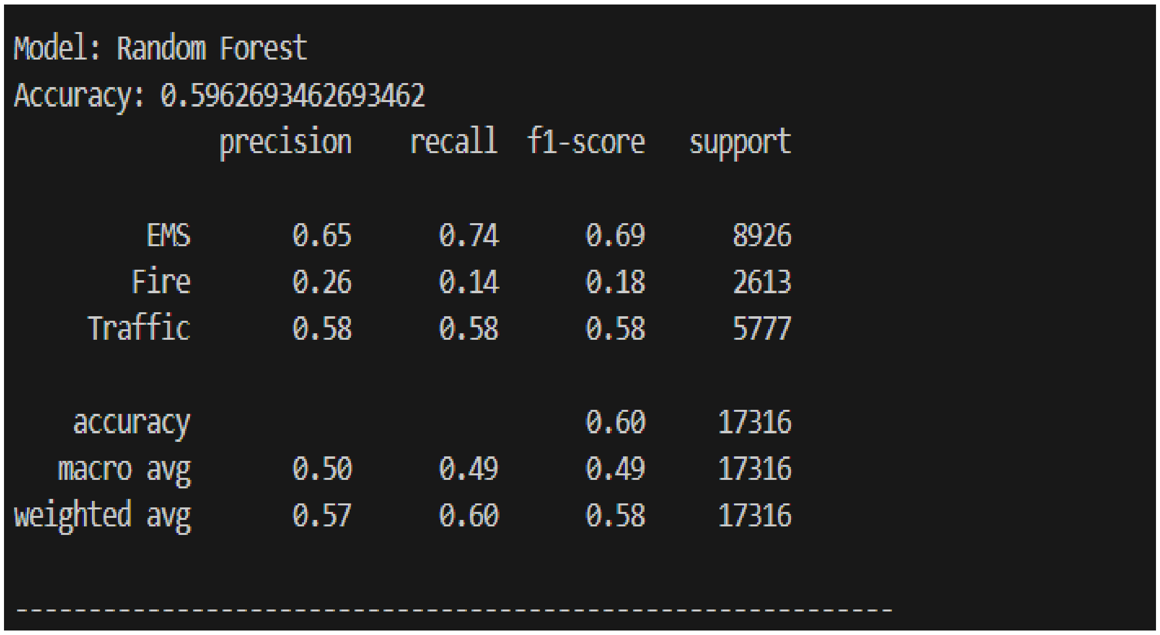


- Logistic Regression Model :



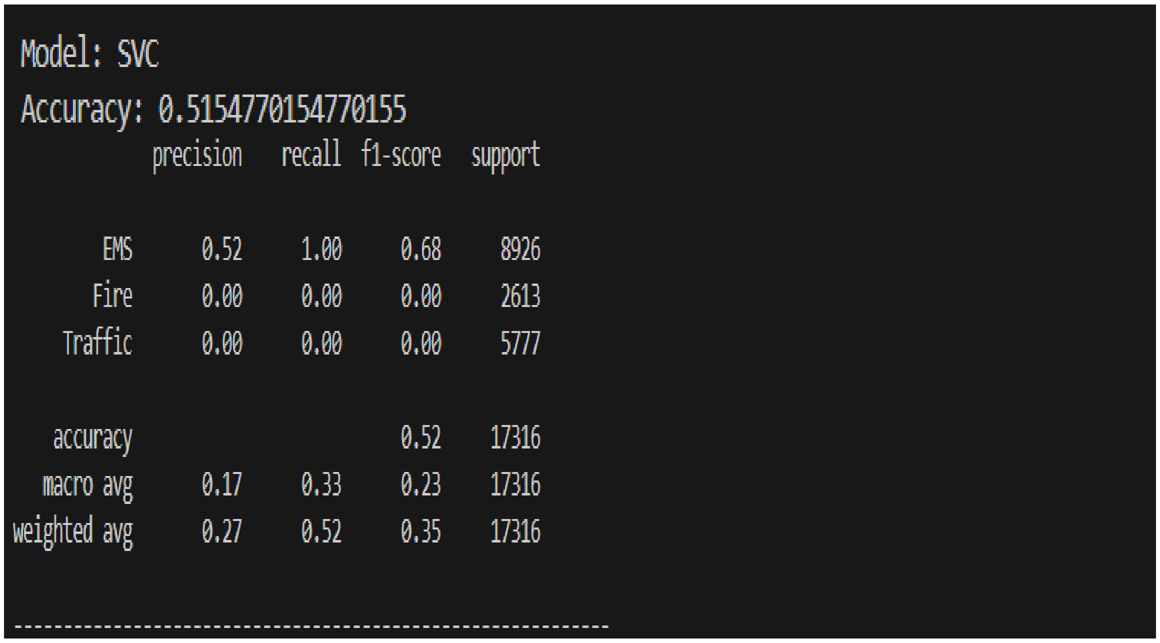
```
Model: Logistic Regression
Accuracy: 0.5146685146685147
              precision    recall  f1-score   support

         EMS       0.52      1.00      0.68      8926
        Fire       0.00      0.00      0.00      2613
     Traffic       0.22      0.00      0.00      5777

    accuracy                           0.51     17316
   macro avg       0.24      0.33      0.23     17316
weighted avg       0.34      0.51      0.35     17316


------------------------------------------------------------
```

- Random Forest Model :



```
Model: Random Forest
Accuracy: 0.5962693462693462
              precision    recall  f1-score   support

         EMS       0.65      0.74      0.69      8926
        Fire       0.26      0.14      0.18      2613
     Traffic       0.58      0.58      0.58      5777

    accuracy                           0.60     17316
   macro avg       0.50      0.49      0.49     17316
weighted avg       0.57      0.60      0.58     17316


------------------------------------------------------------
```

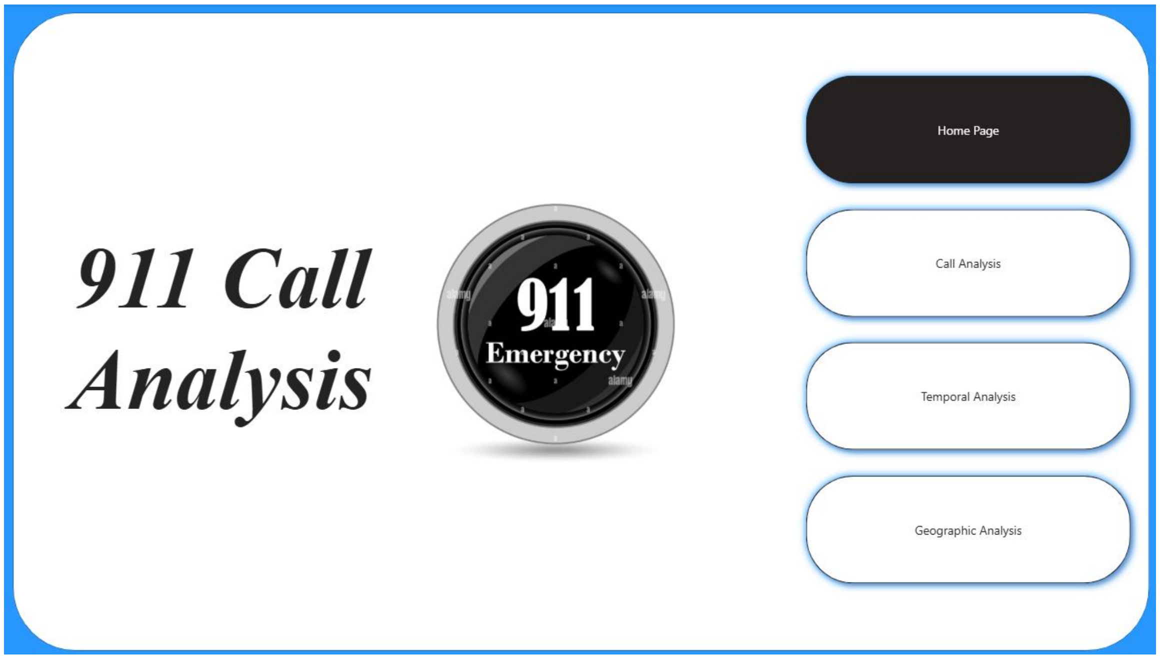
- SVC Model :



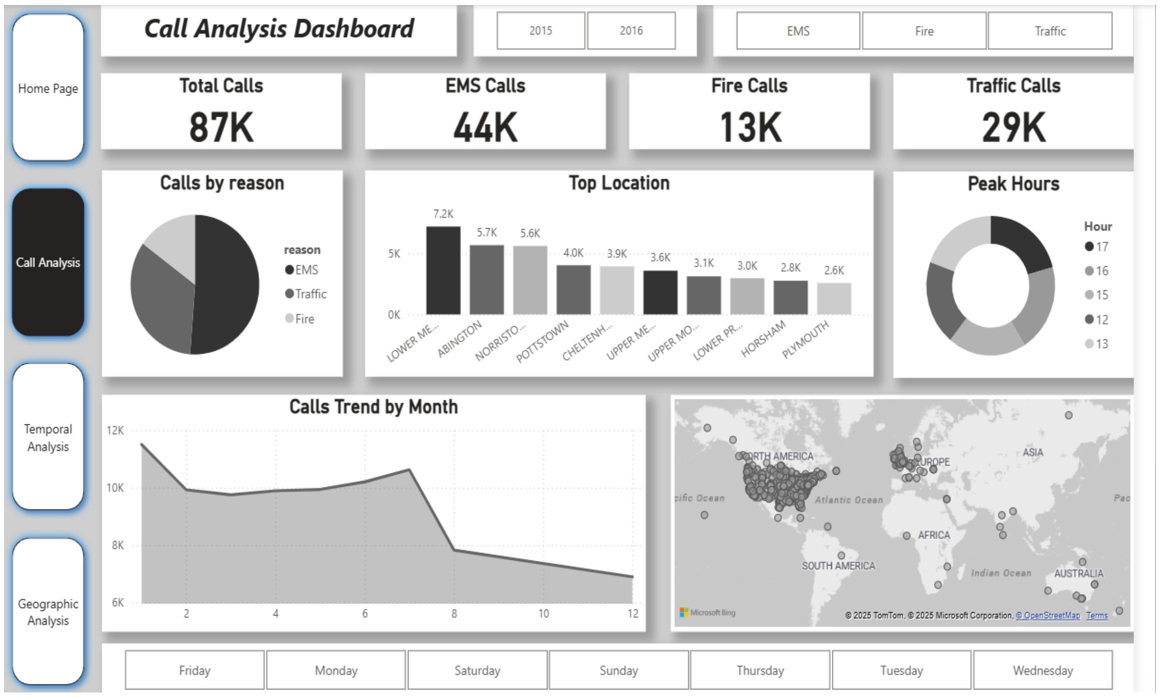
```
Model: SVC
Accuracy: 0.5154770154770155
              precision    recall  f1-score   support

         EMS       0.52      1.00      0.68      8926
        Fire       0.00      0.00      0.00      2613
     Traffic       0.00      0.00      0.00      5777

    accuracy                           0.52     17316
   macro avg       0.17      0.33      0.23     17316
weighted avg       0.27      0.52      0.35     17316


------------------------------------------------------------
```

Power Bi :

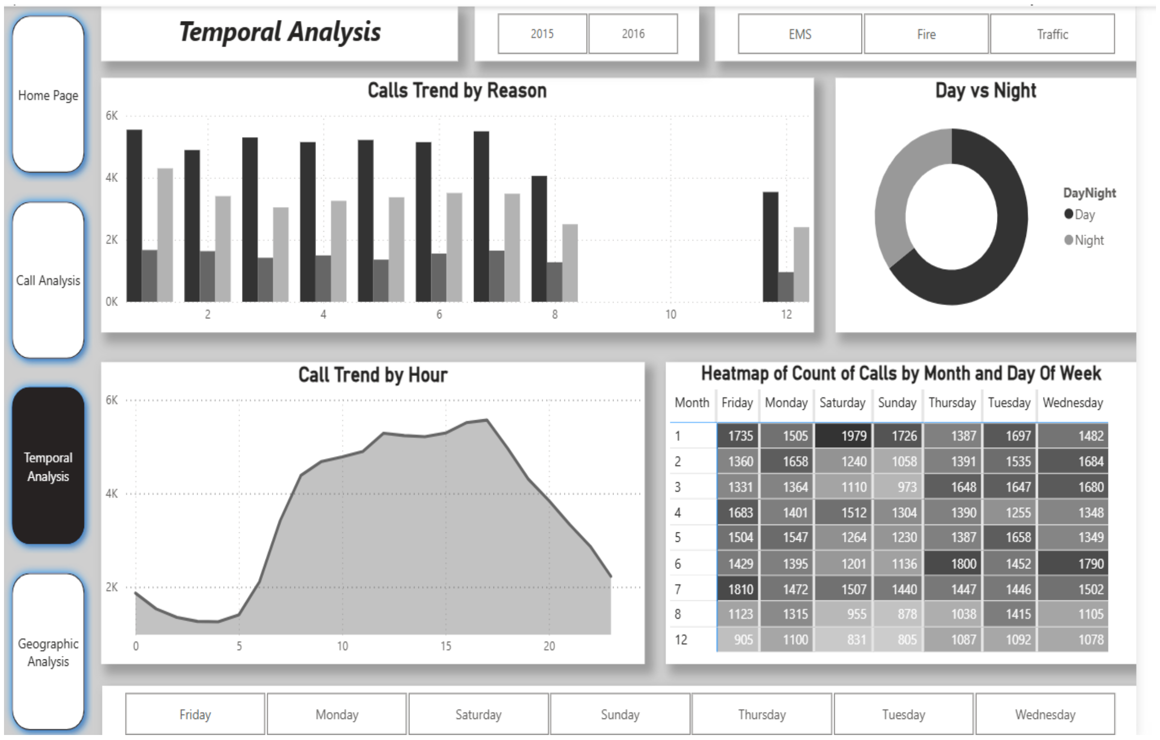
- Home Page :



- Call Analysis :



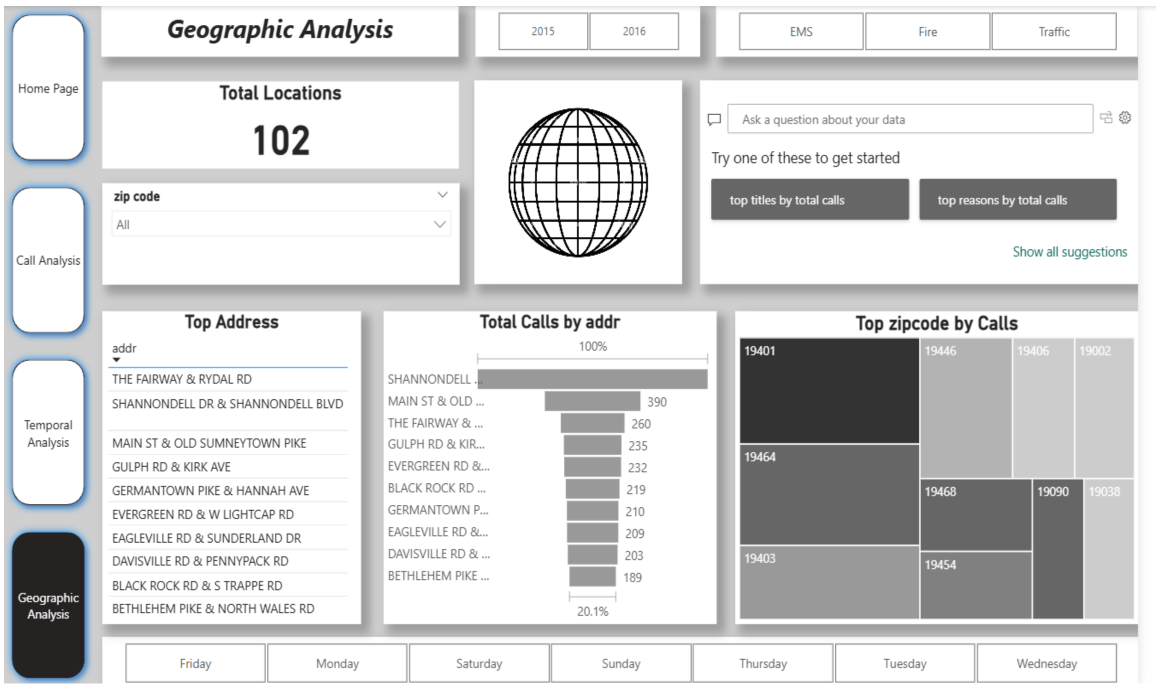
- Temporal Analysis :



Heatmap of Count of Calls by Month and Day Of Week

Month	Friday	Monday	Saturday	Sunday	Thursday	Tuesday	Wednesday
1	1735	1505	1979	1726	1387	1697	1482
2	1360	1658	1240	1058	1391	1535	1684
3	1331	1364	1110	973	1648	1647	1680
4	1683	1401	1512	1304	1390	1255	1348
5	1504	1547	1264	1230	1387	1658	1349
6	1429	1395	1201	1136	1800	1452	1790
7	1810	1472	1507	1440	1447	1446	1502
8	1123	1315	955	878	1038	1415	1105
12	905	1100	831	805	1087	1092	1078

- Geographic Analysis :



Total Locations

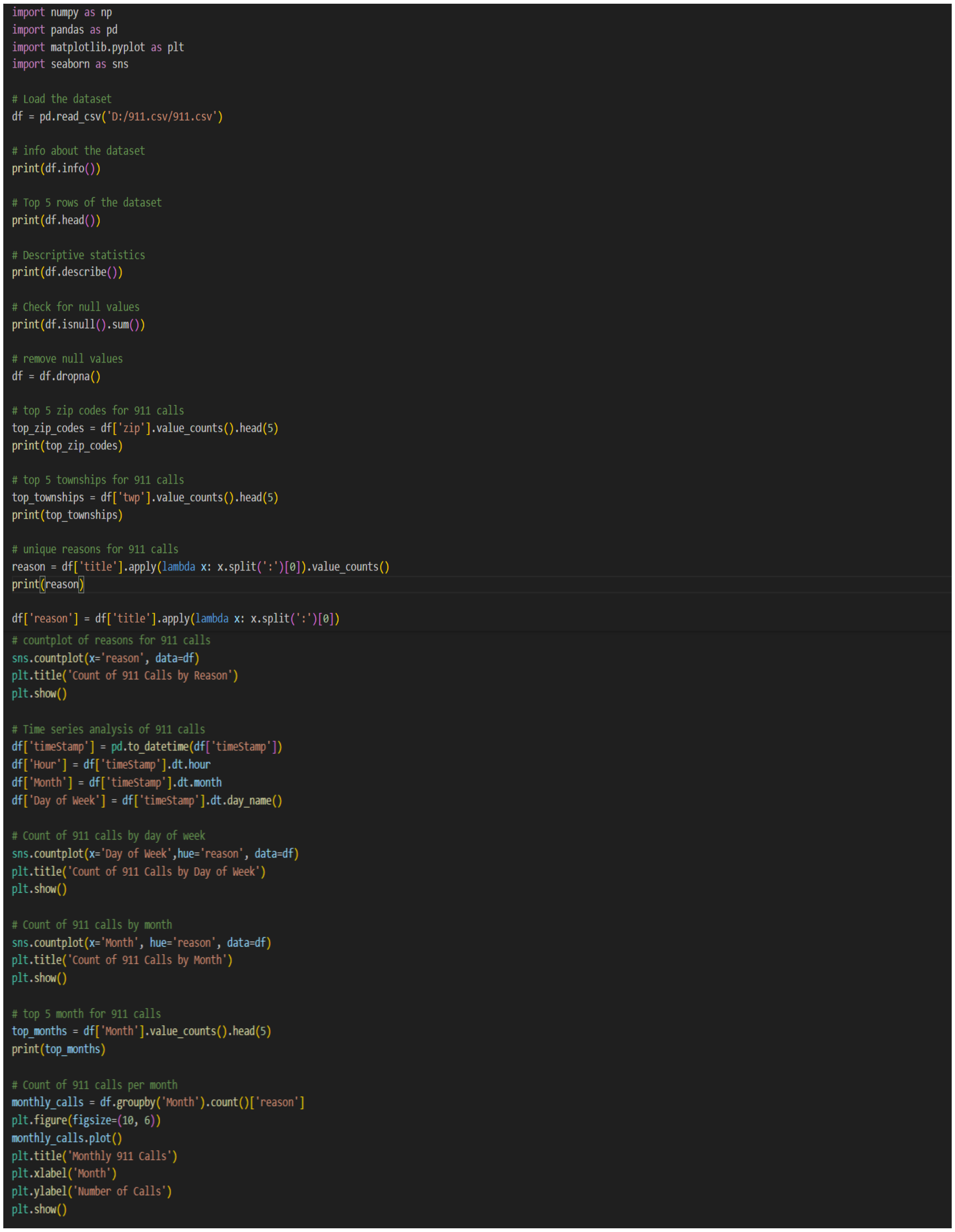
102

Top Address

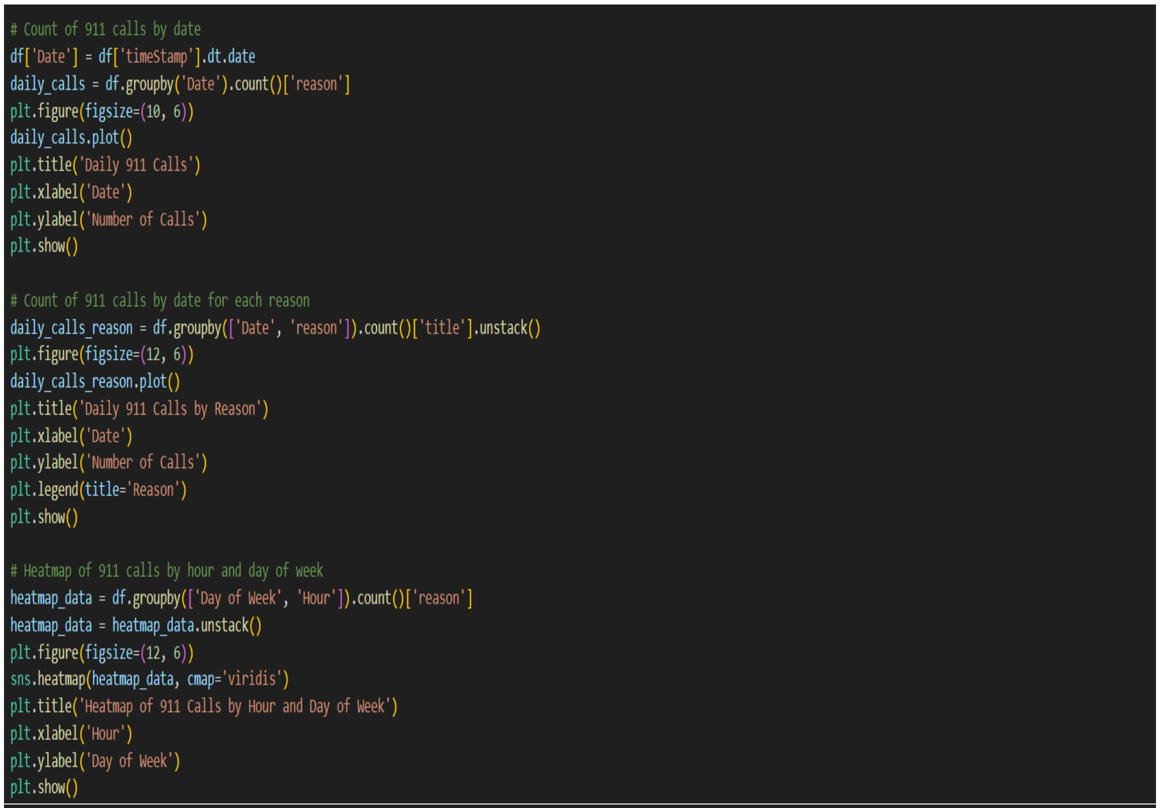
addr
THE FAIRWAY & RYDAL RD
SHANNONDELL DR & SHANNONDELL BLVD
MAIN ST & OLD SUMNEYTOWN PIKE
GULPH RD & KIRK AVE
GERMANTOWN PIKE & HANNAH AVE
EVERGREEN RD & W LIGHTCAP RD
EAGLEVILLE RD & SUNDERLAND DR
DAVISVILLE RD & PENNYPACK RD
BLACK ROCK RD & S TRAPPE RD
BETHLEHEM PIKE & NORTH WALES RD

Code Snippets :

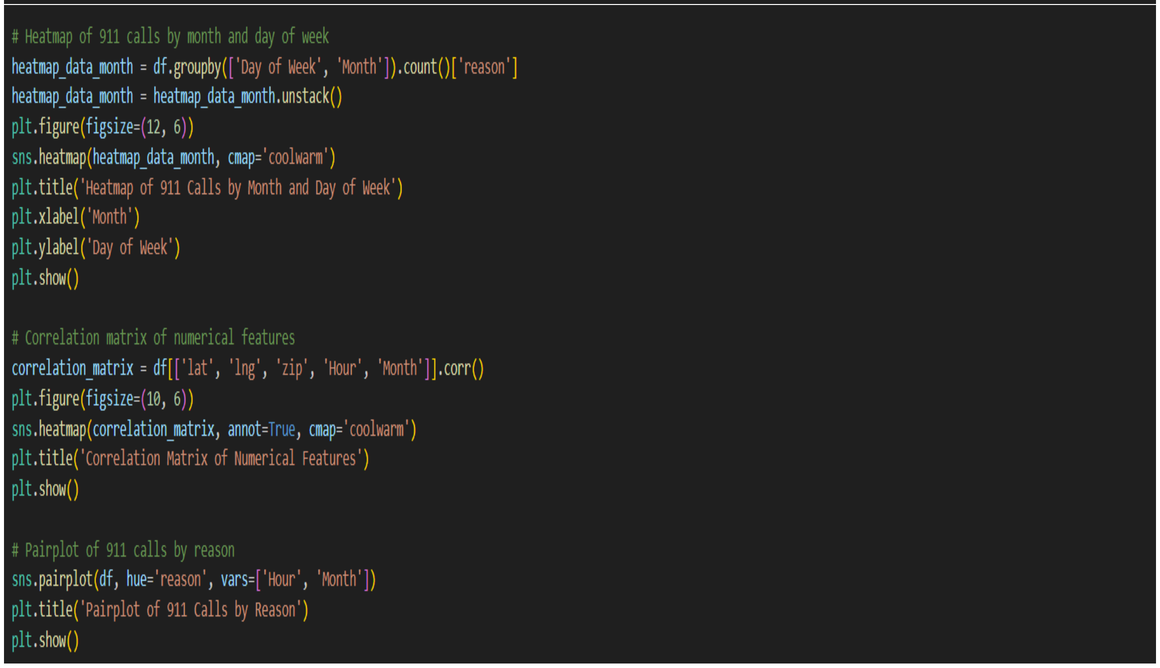
EDA :



```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
df = pd.read_csv('D:/911.csv/911.csv')

# info about the dataset
print(df.info())

# Top 5 rows of the dataset
print(df.head())

# Descriptive statistics
print(df.describe())

# Check for null values
print(df.isnull().sum())

# remove null values
df = df.dropna()

# top 5 zip codes for 911 calls
top_zip_codes = df['zip'].value_counts().head(5)
print(top_zip_codes)

# top 5 townships for 911 calls
top_townships = df['twp'].value_counts().head(5)
print(top_townships)

# unique reasons for 911 calls
reason = df['title'].apply(lambda x: x.split(':')[0]).value_counts()
print(reason)

df['reason'] = df['title'].apply(lambda x: x.split(':')[0])
# countplot of reasons for 911 calls
sns.countplot(x='reason', data=df)
plt.title('Count of 911 Calls by Reason')
plt.show()

# Time series analysis of 911 calls
df['timeStamp'] = pd.to_datetime(df['timeStamp'])
df['Hour'] = df['timeStamp'].dt.hour
df['Month'] = df['timeStamp'].dt.month
df['Day of Week'] = df['timeStamp'].dt.day_name()

# Count of 911 calls by day of week
sns.countplot(x='Day of Week',hue='reason', data=df)
plt.title('Count of 911 Calls by Day of Week')
plt.show()

# Count of 911 calls by month
sns.countplot(x='Month', hue='reason', data=df)
plt.title('Count of 911 Calls by Month')
plt.show()

# top 5 month for 911 calls
top_months = df['Month'].value_counts().head(5)
print(top_months)

# Count of 911 calls per month
monthly_calls = df.groupby('Month').count()['reason']
plt.figure(figsize=(10, 6))
monthly_calls.plot()
plt.title('Monthly 911 Calls')
plt.xlabel('Month')
plt.ylabel('Number of Calls')
plt.show()
```

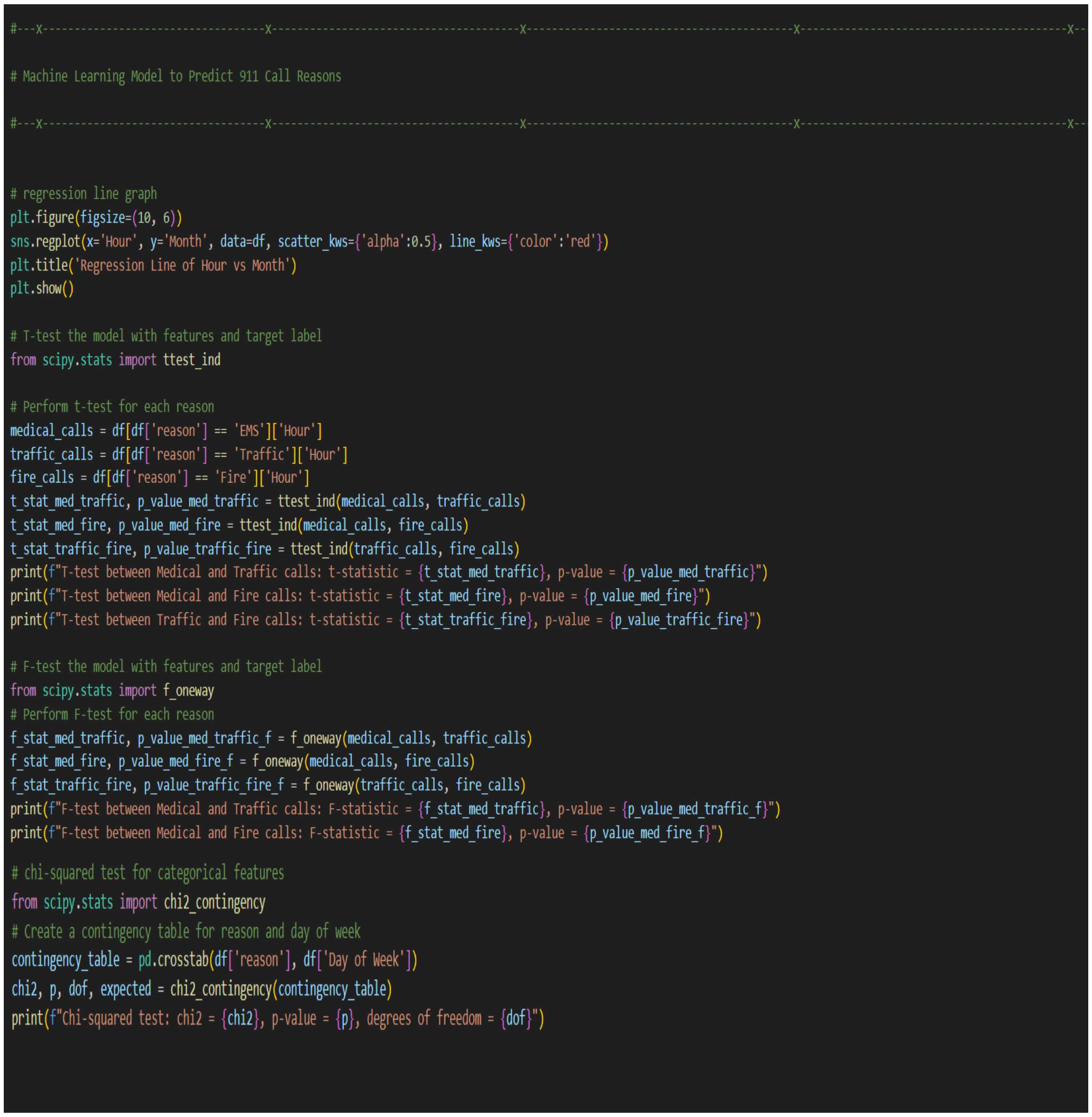


```python
# Count of 911 calls by date
df['Date'] = df['timeStamp'].dt.date
daily_calls = df.groupby('Date').count()['reason']
plt.figure(figsize=(10, 6))
daily_calls.plot()
plt.title('Daily 911 Calls')
plt.xlabel('Date')
plt.ylabel('Number of Calls')
plt.show()

# Count of 911 calls by date for each reason
daily_calls_reason = df.groupby(['Date', 'reason']).count()['title'].unstack()
plt.figure(figsize=(12, 6))
daily_calls_reason.plot()
plt.title('Daily 911 Calls by Reason')
plt.xlabel('Date')
plt.ylabel('Number of Calls')
plt.legend(title='Reason')
plt.show()

# Heatmap of 911 calls by hour and day of week
heatmap_data = df.groupby(['Day of Week', 'Hour']).count()['reason']
heatmap_data = heatmap_data.unstack()
plt.figure(figsize=(12, 6))
sns.heatmap(heatmap_data, cmap='viridis')
plt.title('Heatmap of 911 Calls by Hour and Day of Week')
plt.xlabel('Hour')
plt.ylabel('Day of Week')
plt.show()
```

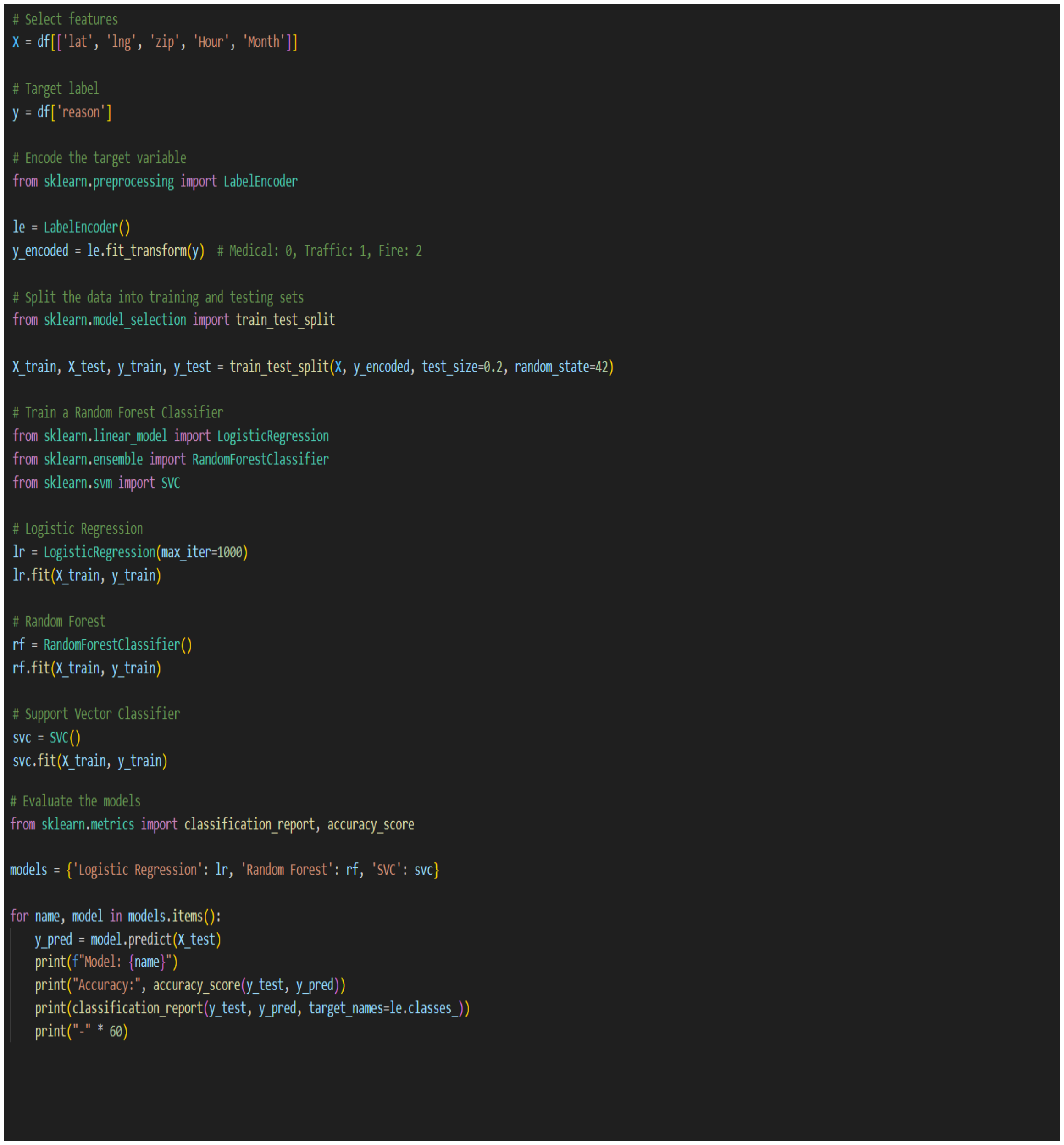


```python
# Heatmap of 911 calls by month and day of week
heatmap_data_month = df.groupby(['Day of Week', 'Month']).count()['reason']
heatmap_data_month = heatmap_data_month.unstack()
plt.figure(figsize=(12, 6))
sns.heatmap(heatmap_data_month, cmap='coolwarm')
plt.title('Heatmap of 911 Calls by Month and Day of Week')
plt.xlabel('Month')
plt.ylabel('Day of Week')
plt.show()

# Correlation matrix of numerical features
correlation_matrix = df[['lat', 'lng', 'zip', 'Hour', 'Month']].corr()
plt.figure(figsize=(10, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix of Numerical Features')
plt.show()

# Pairplot of 911 calls by reason
sns.pairplot(df, hue='reason', vars=['Hour', 'Month'])
plt.title('Pairplot of 911 Calls by Reason')
plt.show()
```

Machine-Learning Algo :



```
#---x----------------------------------x--------------------------------------x-----------------------------------------x-----------------------------------------x--

# Machine Learning Model to Predict 911 Call Reasons

#---x----------------------------------x--------------------------------------x-----------------------------------------x-----------------------------------------x--


# regression line graph
plt.figure(figsize=(10, 6))
sns.regplot(x='Hour', y='Month', data=df, scatter_kws={'alpha':0.5}, line_kws={'color':'red'})
plt.title('Regression Line of Hour vs Month')
plt.show()

# T-test the model with features and target label
from scipy.stats import ttest_ind

# Perform t-test for each reason
medical_calls = df[df['reason'] == 'EMS']['Hour']
traffic_calls = df[df['reason'] == 'Traffic']['Hour']
fire_calls = df[df['reason'] == 'Fire']['Hour']
t_stat_med_traffic, p_value_med_traffic = ttest_ind(medical_calls, traffic_calls)
t_stat_med_fire, p_value_med_fire = ttest_ind(medical_calls, fire_calls)
t_stat_traffic_fire, p_value_traffic_fire = ttest_ind(traffic_calls, fire_calls)
print(f"T-test between Medical and Traffic calls: t-statistic = {t_stat_med_traffic}, p-value = {p_value_med_traffic}")
print(f"T-test between Medical and Fire calls: t-statistic = {t_stat_med_fire}, p-value = {p_value_med_fire}")
print(f"T-test between Traffic and Fire calls: t-statistic = {t_stat_traffic_fire}, p-value = {p_value_traffic_fire}")

# F-test the model with features and target label
from scipy.stats import f_oneway
# Perform F-test for each reason
f_stat_med_traffic, p_value_med_traffic_f = f_oneway(medical_calls, traffic_calls)
f_stat_med_fire, p_value_med_fire_f = f_oneway(medical_calls, fire_calls)
f_stat_traffic_fire, p_value_traffic_fire_f = f_oneway(traffic_calls, fire_calls)
print(f"F-test between Medical and Traffic calls: F-statistic = {f_stat_med_traffic}, p-value = {p_value_med_traffic_f}")
print(f"F-test between Medical and Fire calls: F-statistic = {f_stat_med_fire}, p-value = {p_value_med_fire_f}")

# chi-squared test for categorical features
from scipy.stats import chi2_contingency
# Create a contingency table for reason and day of week
contingency_table = pd.crosstab(df['reason'], df['Day of Week'])
chi2, p, dof, expected = chi2_contingency(contingency_table)
print(f"Chi-squared test: chi2 = {chi2}, p-value = {p}, degrees of freedom = {dof}")
```



```
# Select features
X = df[['lat', 'lng', 'zip', 'Hour', 'Month']]

# Target label
y = df['reason']

# Encode the target variable
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
y_encoded = le.fit_transform(y)  # Medical: 0, Traffic: 1, Fire: 2

# Split the data into training and testing sets
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y_encoded, test_size=0.2, random_state=42)

# Train a Random Forest Classifier
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC

# Logistic Regression
lr = LogisticRegression(max_iter=1000)
lr.fit(X_train, y_train)

# Random Forest
rf = RandomForestClassifier()
rf.fit(X_train, y_train)

# Support Vector Classifier
svc = SVC()
svc.fit(X_train, y_train)

# Evaluate the models
from sklearn.metrics import classification_report, accuracy_score

models = {'Logistic Regression': lr, 'Random Forest': rf, 'SVC': svc}

for name, model in models.items():
    y_pred = model.predict(X_test)
    print(f"Model: {name}")
    print("Accuracy:", accuracy_score(y_test, y_pred))
    print(classification_report(y_test, y_pred, target_names=le.classes_))
    print("-" * 60)
```

Results and Discussion

Output / Report :

The analysis of the 911 emergency call dataset generated valuable insights into emergency response trends and patterns. Here is a summary of the key outcomes with brief descriptions:

- **Most Frequent Emergency Type – EMS:** Emergency Medical Services (EMS) accounted for the majority of calls, reflecting the high demand for health-related emergency response.
- **Peak Calling Hours:** The majority of 911 calls occurred between 9 AM and 4 PM, aligning with working hours and daytime activities.
- **Busiest Day – Monday:** Analysis showed that Mondays consistently had the highest call volumes across all emergency types.
- **Geographical Insights:** Townships and zip codes with the highest call frequencies were identified, enabling focused resource planning.
- **Temporal Patterns:** Time-based heatmaps and trend lines highlighted repeating patterns across different days of the week and hours of the day.
- **Statistical Validation:** T-test, F-test, and Chi-square test validated differences in emergency call frequencies and relationships between variables.
- **Model Accuracy – Random Forest:** Among the ML models used, Random Forest achieved the highest classification accuracy for predicting emergency types.
- **Interactive Dashboard:** A dynamic Power BI dashboard was created to allow users to filter data by emergency type, date, and township, offering an intuitive way to explore and interpret insights.

Challenges Faced :

Throughout the project, the team encountered several technical and practical challenges that required careful problem-solving and collaboration to overcome. These challenges not only tested our understanding of data science concepts but also helped us develop resilience and adaptability.

- **Handling missing and inconsistent data:** The dataset contained several missing values and inconsistencies in key columns such as zip codes and timestamps. We had to clean the data meticulously to avoid inaccurate results.
- **Encoding categorical labels:** Transforming textual categories (like 'EMS', 'Traffic', 'Fire') into numerical formats for machine learning models required proper encoding techniques to ensure the models interpreted the data correctly.
- **Aligning Python results with Power BI visuals:** It was challenging to ensure consistency between the Python-generated insights and those presented in the Power BI dashboard. Data formatting and filtering had to be synchronized between both environments.
- **Model evaluation and tuning:** Selecting the best-performing model involved repeated training, testing, and tuning. Hyperparameter optimization and avoiding overfitting were ongoing concerns.
- **Data visualization clarity:** Presenting complex insights in a clear and understandable way, especially for non-technical stakeholders, was a key design challenge in building the dashboard.

Learnings :

This project provided the team with valuable insights into real-world data science practices. Each phase of the work—from preprocessing to dashboard creation—contributed to practical learning and skill-building. Below are the key takeaways:

- **End-to-end data science project workflow:** Gained hands-on experience in managing all stages of a data science project including data cleaning, EDA, statistical testing, model building, and reporting.
- **Team collaboration and problem-solving:** Improved communication, task-sharing, and decision-making through collaborative work on technical challenges and project planning.
- **Statistical reasoning and testing:** Learned how to apply statistical tests like T-test, F-test, and Chi-square test to validate hypotheses and understand relationships in data.
- **Machine learning implementation:** Practiced applying classification algorithms, evaluating models, and tuning parameters to enhance performance.
- **Visual storytelling with dashboards:** Acquired skills in converting analytical insights into clear, interactive visuals using Power BI for non-technical users.

- **Interpretation and communication of results:** Strengthened ability to interpret data patterns and communicate findings effectively through structured documentation and visualization.

Conclusion

Summary :

The project successfully demonstrated how real-world emergency service data can be leveraged for actionable insights through statistical and machine learning techniques. The Power BI dashboard facilitated effective communication of trends and predictions to stakeholders. This hands-on group experience improved technical skills and fostered teamwork and analytical thinking.