IBM Data Science Capstone Project

Lautaro Russo

10/10/2023

[LinkedIn | lautaro-russo]

[Github | Larusso94]

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

Methodologies Summary:

- Explored the data to understand its structure and insights.
- Decided to discard zip codes with a negligible number of calls (less than 0.1%).

Results Summary:

- Distinct patterns were observed in the distribution of 911 calls across different zip codes.
- Certain zip codes showcased atypical patterns, such as 18076, 18974, and 19010.
- Zip code 18076 had a higher than average EMS-related calls, while 18974 had a lower proportion.
- Zip code 18974 exhibited a substantially higher percentage of Traffic-related calls than the average.
- Some regions consistently deviated from the average across multiple call categories.
- The data confirmed the hypothesis of anomalous 911 call categories in certain zip codes.
- Findings highlight the significance of localized analysis for effective emergency response planning.

Introduction

Project Background and Context:

- Emergency call systems are a cornerstone of modern societal safety.
- They connect the public with vital emergency response agencies: fire departments, medical services, and law enforcement.
- Analyzing these calls offers insights into public safety concerns, vulnerabilities, and response efficiencies.

Problems We Aim to Address:

- Exploratory Data Analysis (EDA):
  - What is the structure and nature of our 911 call dataset?
  - What kind of information does the dataset provide?
- Visualization:
  - How can we visually represent the patterns and trends in the call data?
  - How can we make the insights more digestible for non-data experts?
- Anomaly Detection:
  - Are there zip codes with unusually high or low numbers of specific emergency types?
  - How can these anomalies inform resource allocation and emergency response planning?

Section 1

Methodology

Data Collection

- Data sourced from Kaggle
- Data is already presented as tabular format in CSV file.
- Loading and filtering is performing using Pandas in order to avoid NaN values.

EDA with Pandas & SQL

- Pandas dataframe methods are used to explore the features of the dataset:
  - **df.info()** Overview of columns, data types, and null values.
  - **df.describe()** Statistical summary for numerical data (e.g., mean, median).
  - **df.head()** Displays the first few rows for a quick peek.
  - **df.value_counts()** Count of unique values for categorical columns.

Note : SQL version queries of the Pandas methods are used to explore the DataFrame. This are performed as required by the course.

Feature engineering

- **Title Splitting:** The **title** column will be divided into two new columns:
  - **cat_1**: This will represent the primary category of the call.
  - **cat_2**: This will represent the sub-category of the call.

- **Timestamp Decomposition:** The **timestamp** column will be split into four distinct columns to capture various temporal aspects:
  - **monthDay**: Specific day when the call was made.
  - **year**: Year of the call.
  - **month**: Month when the call was recorded.
  - **hour**: The specific hour of the day when the call was made.
  - **weekDay**: Day of the week when the call was recorded.

Seaborn Scatter plots & Interactive Map with Folium

- **Seaborn scatter:**
  - Distribution of 911 calls by coordinates and colored by zipcode.
  - x_axis : longitude of the call.
  - y_axis : latitude of the call.
  - color : lin_space generated colors from zipcode list (around 60 zipcodes).
    -
- **Seaborn scatter:**
  - Distribution of 911 calls by coordinates and colored by cat_1.
  - x_axis : longitude of the call.
  - y_axis : latitude of the call.
  - color : One of the three categories of cat_1 (EMS, Fire, Traffic)
  -
- **Interactive Foilium map:**
  - FastMakerCluster of the distribution of 911 by coordinates.
  - Markers are 911 calls.

- **Seaborn Line plots:**
  - Number of calls by lapse of time (hour, monthDay, weekDay, month)
  - color : One of the tree categories of cat_1 (EMS, Fire, Traffic)

Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model
- You need present your model development process using key phrases and flowchart
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose

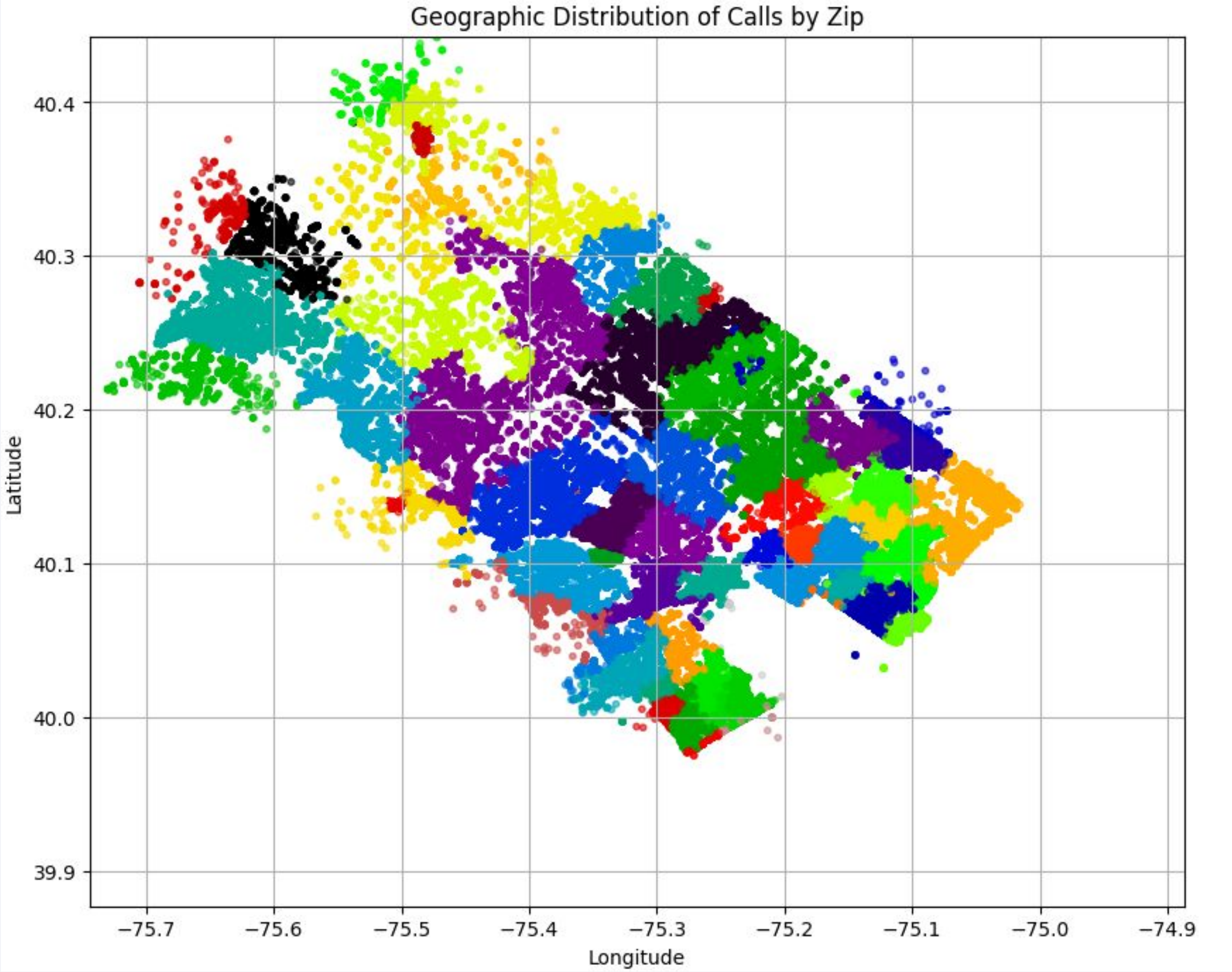
Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

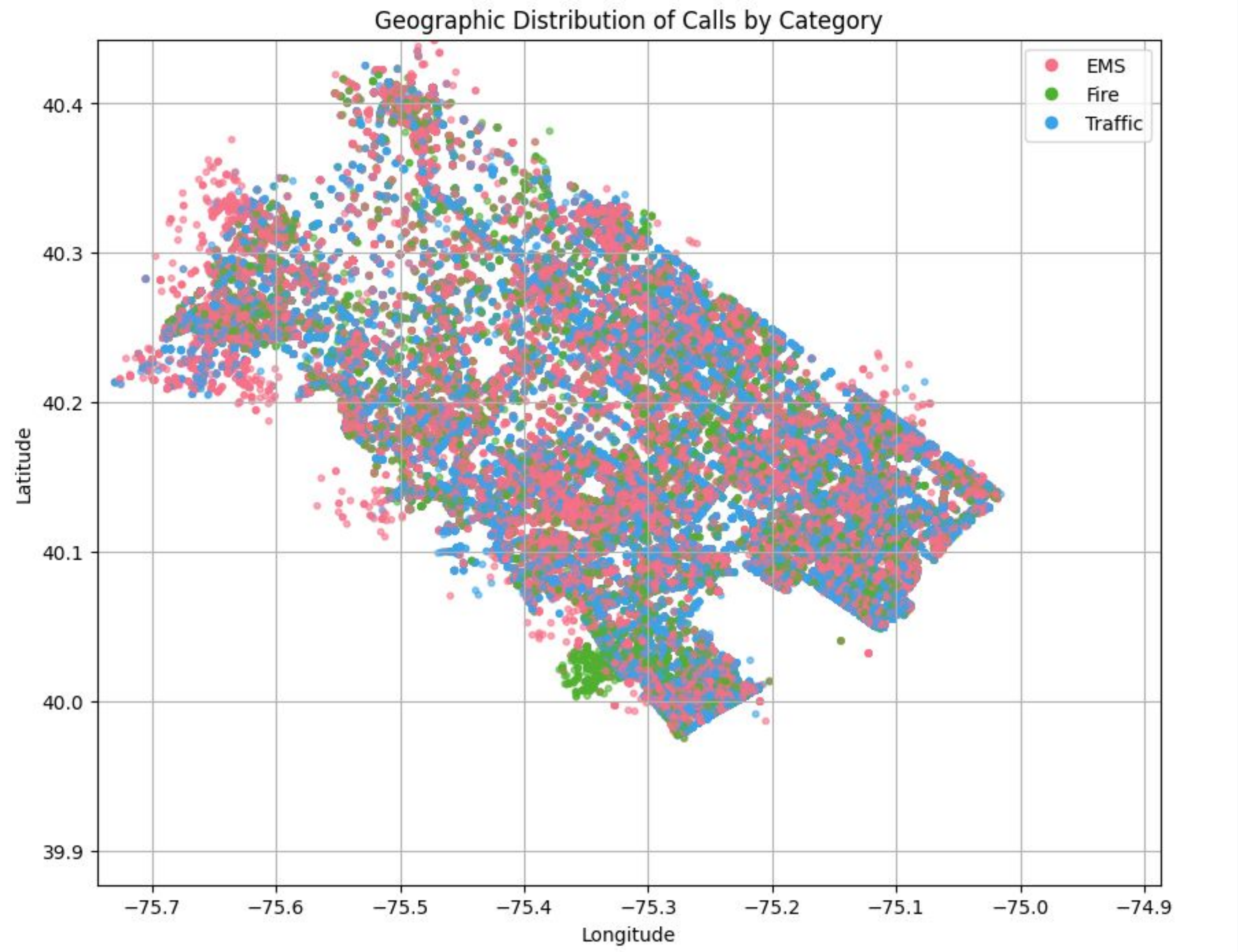
Section 2

Insights drawn from EDA

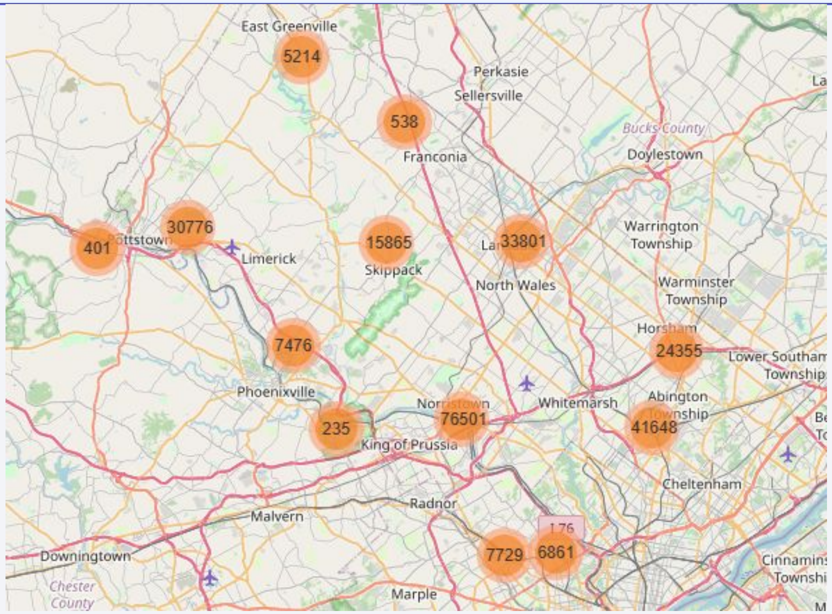
Scatter plot by zipcode

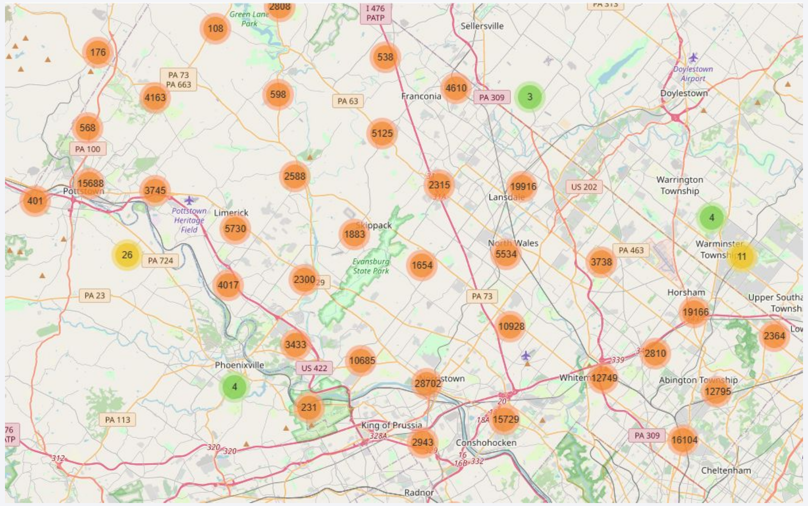


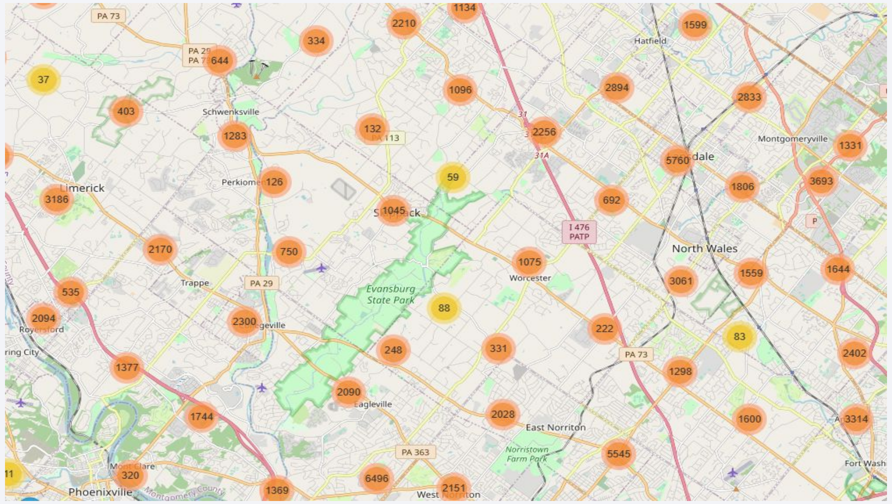
Scatter plot by zipcode



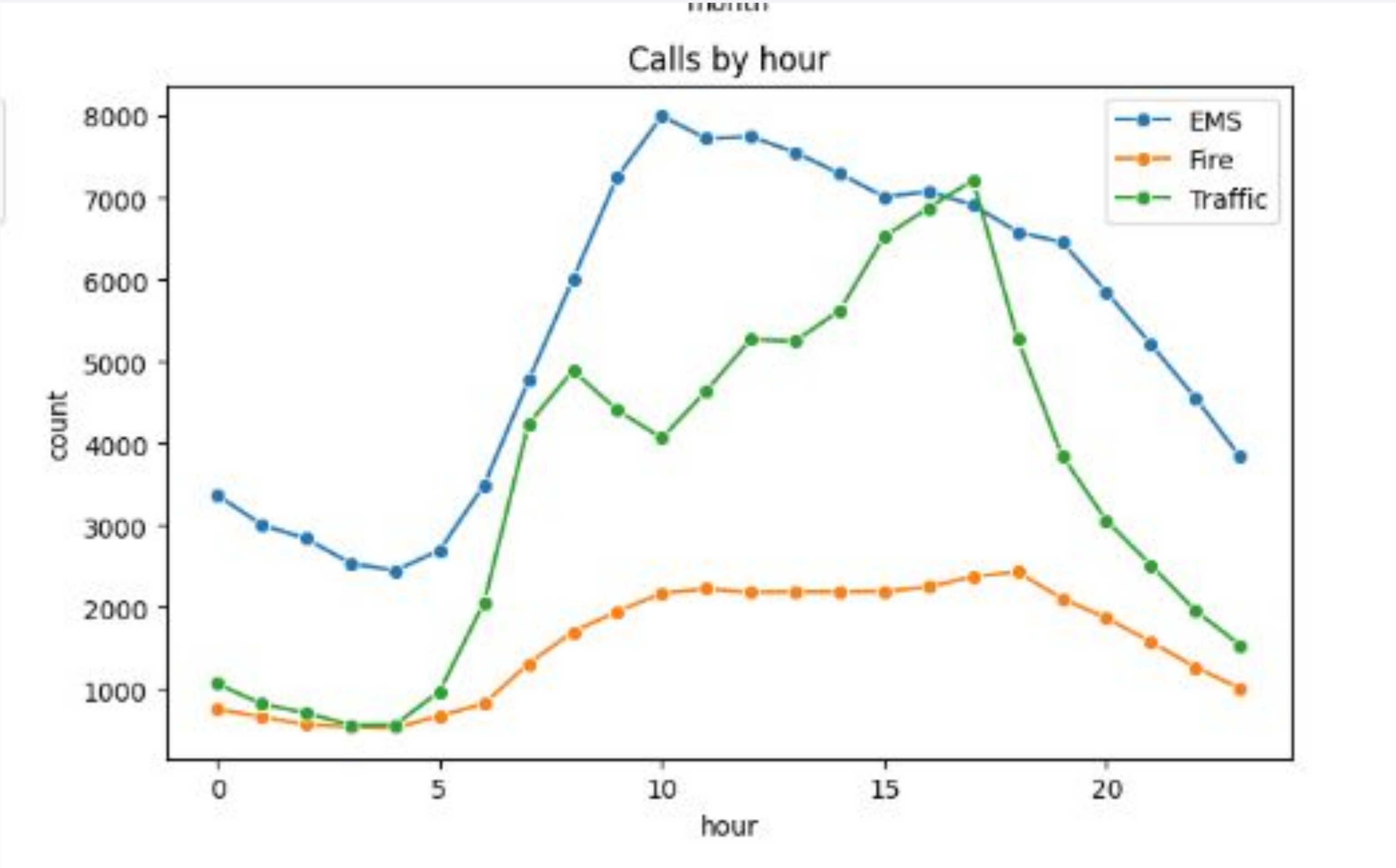
Interactive Folium map



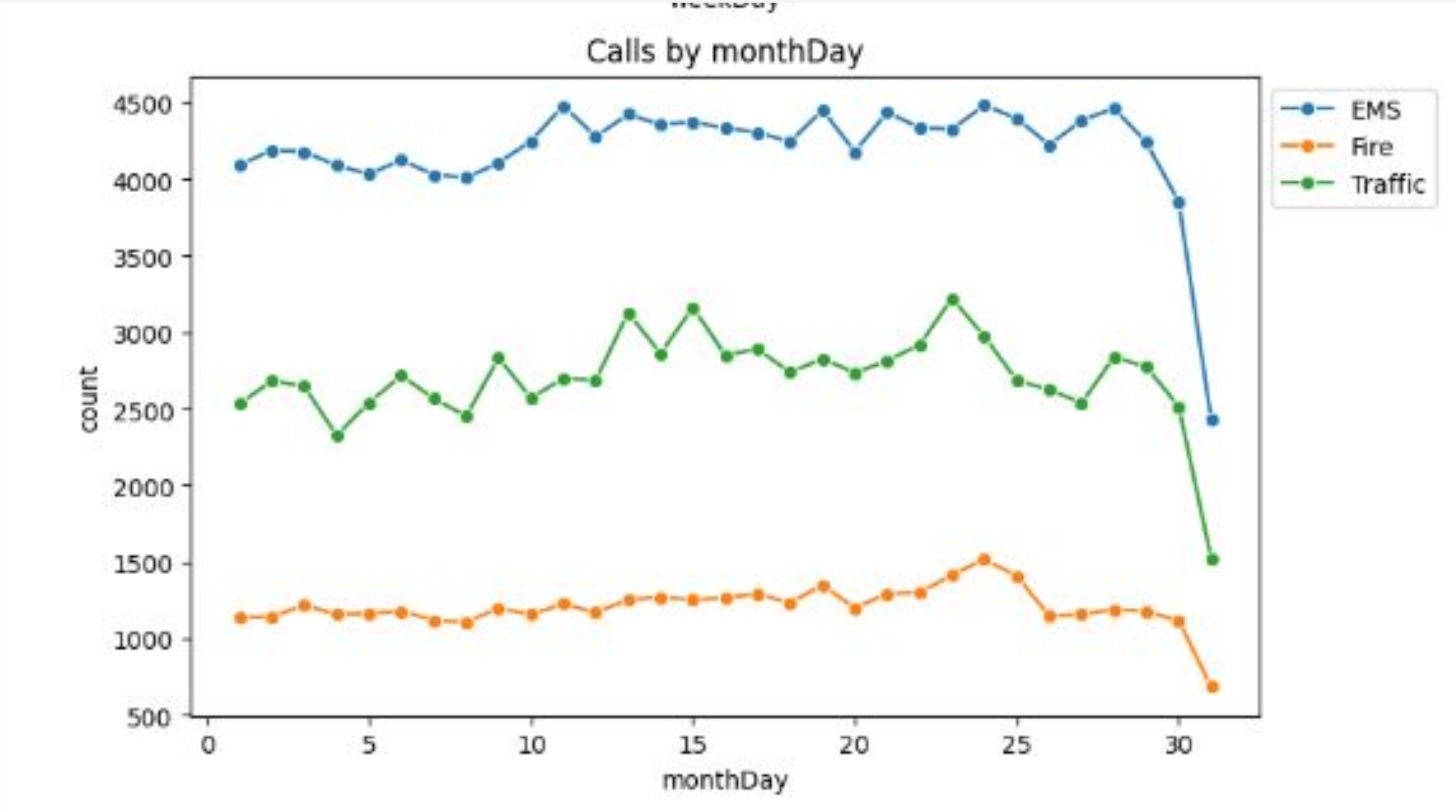




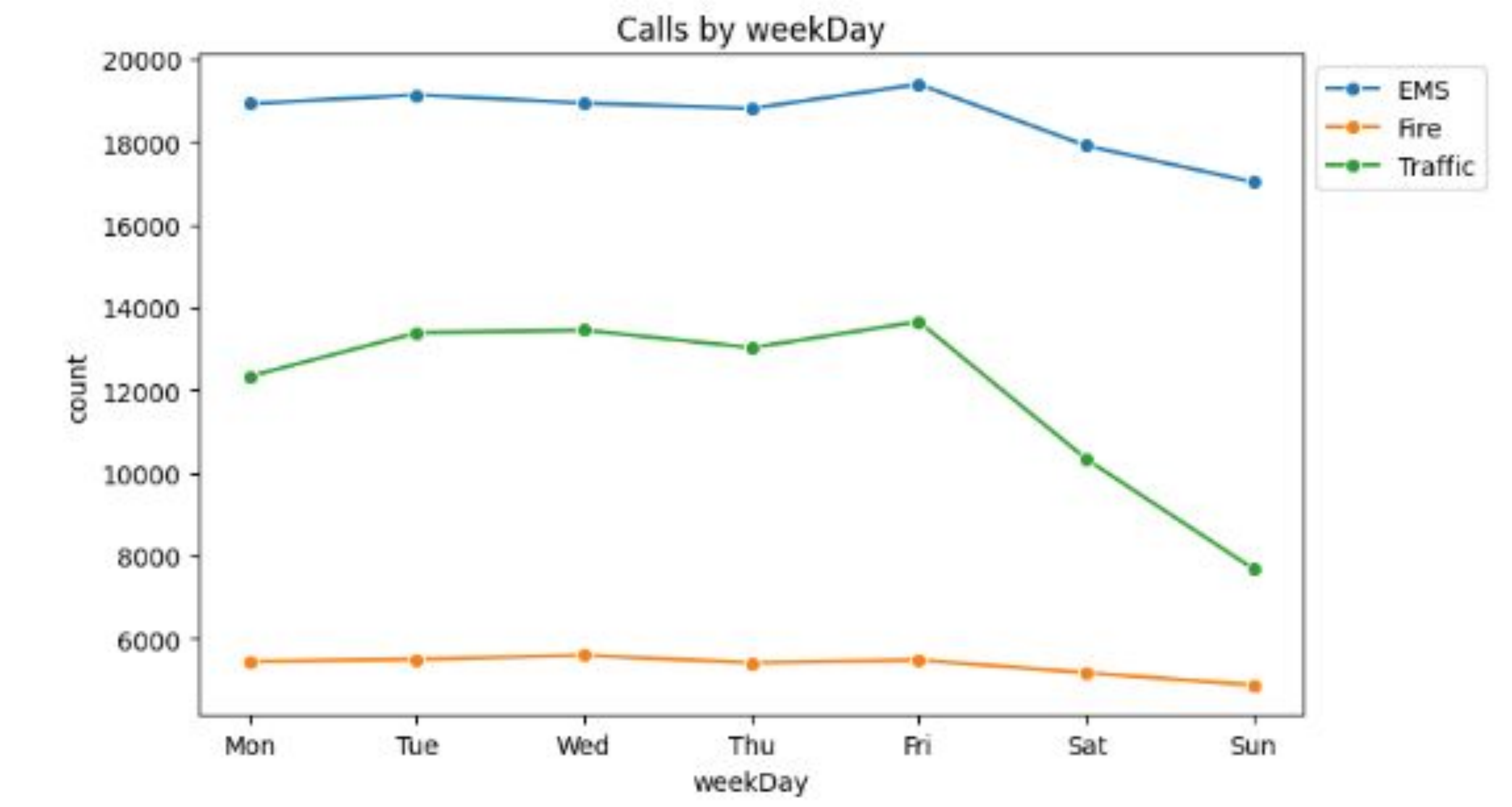
LinePlot by hour



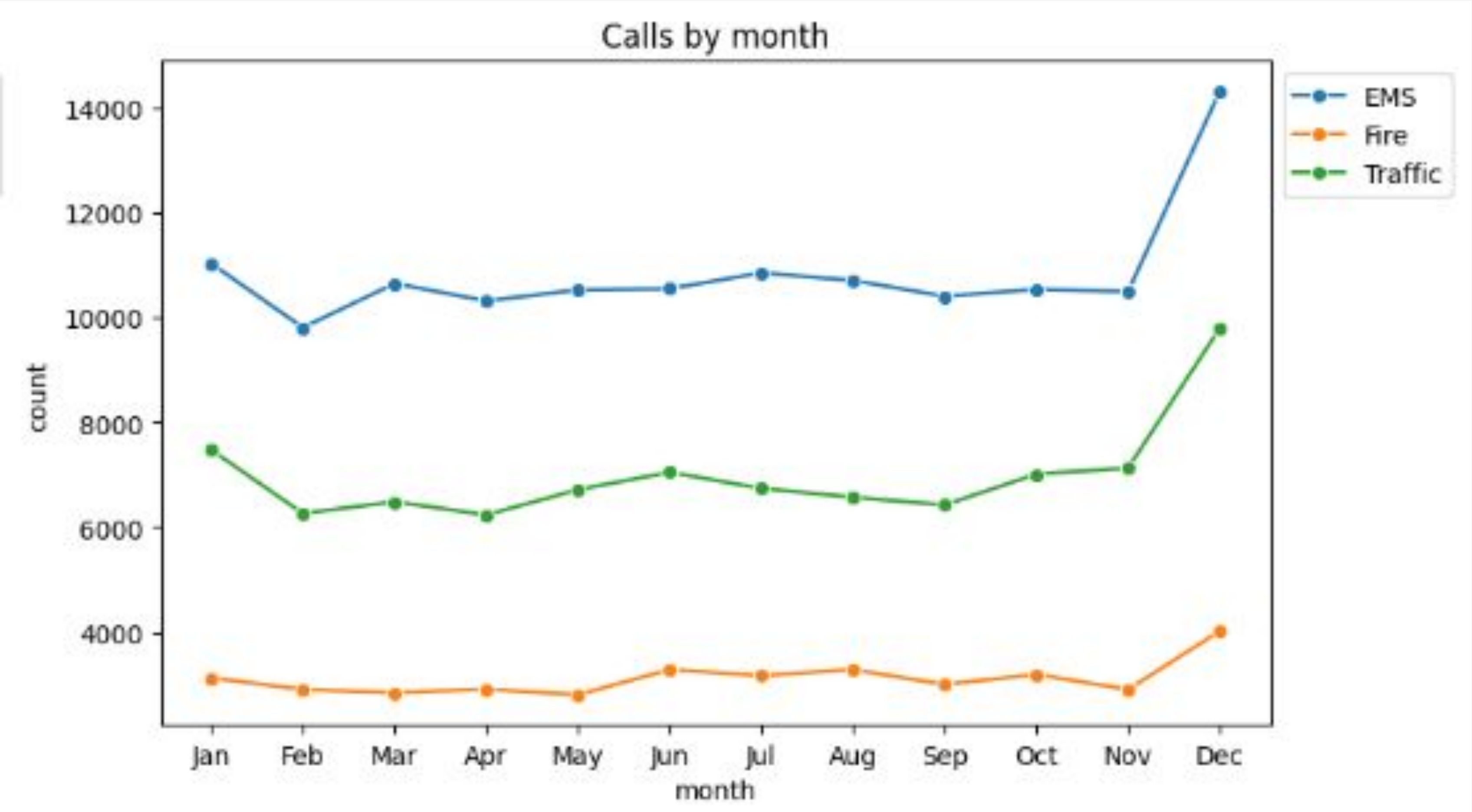
LinePlot by day



LinePlot by weekDay



LinePlot by month



Section 5

Predictive Analysis (Classification)

Methodologies summary

- Task to perform is to predict ZipCode by Latitude and longitude Coordinates.
- Classifiers:
  - Tree Classifier
  - Random Forest Classifier
  - Logistic Regression Classifier
  - K-neares neighbor Classifier
  - Stack Ensemble Classifier of the previous classifiers.
- Use of:
  - Polynomial Features as feature engineering
  - Stratified train/test split
  - Grid search for parameters optimization
  - Ensemble methods to enhance prediction capabilities

Grid search

After using grid search for model parameters selection:

```
for est in best_estimators.items():
    print(est)
```

```
('KNN', KNeighborsClassifier(n_neighbors=3, weights='distance'))
('DecisionTree', DecisionTreeClassifier(criterion='entropy', max_depth=20))
('RandomForest', RandomForestClassifier())
('LogisticRegression', LogisticRegression(C=10, max_iter=10000, solver='newton-cg'))
```

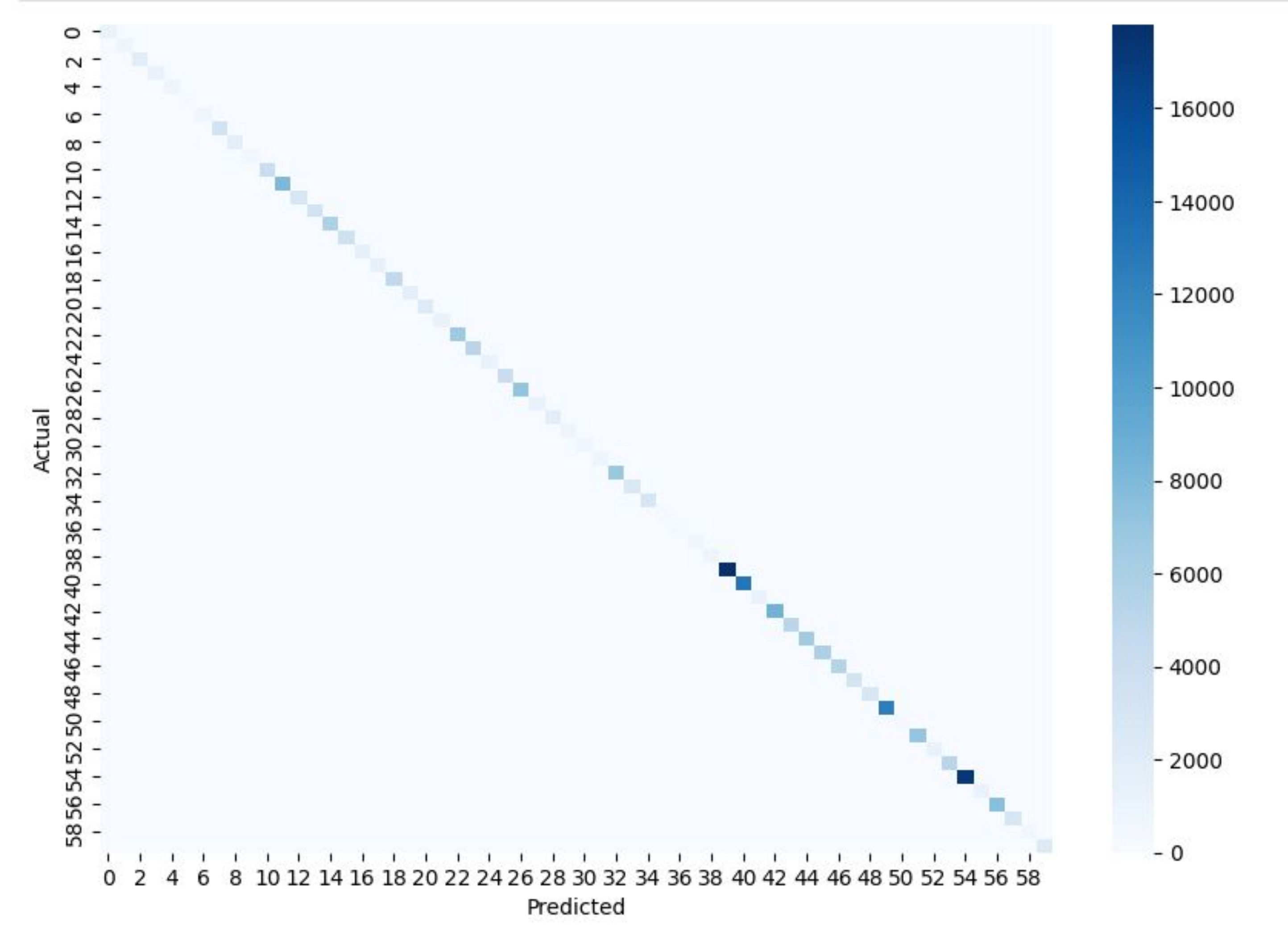
Stack Ensemble and Results

- Stacked ensemble consists of a logistic regression on top of previous model predictions

- Four metrics are calculated in order to assess performance:
  - Accuracy
  - Recall
  - Precission
  - F1-Score

```
Model:KNN, Accuracy: 0.9882,Precision: 0.9883, Recall: 0.9882, F1-Score: 0.9882
Model:DecisionTree, Accuracy: 0.9878,Precision: 0.9879, Recall: 0.9878, F1-Score: 0.9878
Model:RandomForest, Accuracy: 0.9895,Precision: 0.9895, Recall: 0.9895, F1-Score: 0.9895
Model:LogisticRegression, Accuracy: 0.7650,Precision: 0.7351, Recall: 0.7650, F1-Score: 0.7235
Model:Stacked Ensemble, Accuracy: 0.9897,Precision: 0.9897, Recall: 0.9897, F1-Score: 0.9897
```

Confusion Matrix



Thank you!