Analysis of the SF Fire Department Calls

Sean Malloy
COMP-261 - Data Science
University of the Pacific
Stockton, CA
s_malloy1@u.pacific.edu

***Abstract*— Analyzing the San Francisco Fire Department (SFFD) data can provide valuable insights into factors that impact call effectiveness and resource utilization in San Francisco. By studying variables such as response times, area code, and types of emergencies, we can identify areas that need improvement and prioritize resources more effectively. With this information, we can make data-driven decisions to enhance emergency services in San Francisco, ultimately improving the safety and well-being of the community. This paper takes a comprehensive approach to study the SFFD dataset, including statistical analysis of important variables, exploratory data analysis, correlation analysis, and machine learning models to identify key variables. By analyzing these variables, we aim to improve emergency response times and identify areas for future improvement.**

***Keywords*—Data Analysis, R, Jupiter, AI, Data Analytics, Exploratory Data Analysis**

I. Introduction

In this project, I decided to research the Fire Department Calls for Service dataset provided by data.sfgov.org, which is a massive dataset that includes all fire units' responses to calls in San Francisco. I decided to pursue research on this dataset for several reasons. Firstly, I found it to be an intriguing dataset with the potential to uncover insights that could optimize Fire Department calls, ultimately leading to saving lives. As someone interested in health and medical science, I saw this as an opportunity to learn more about health and system optimization. Secondly, the dataset is a treasure trove of information on San Francisco Fire Department calls, with data spanning from 2001 to the present, updated daily, and containing 46 variables, over 6 million observations, and a file size of 2.5 GB. With such vast amounts of data, I am confident that I have ample room to conduct thorough research and uncover valuable insights that can benefit the community.

However, I encountered several challenges while working with such a large dataset. Firstly, the sheer volume of variables and data made it challenging to determine which variables were crucial for my research and which were extraneous. Additionally, I struggled to identify which correlations were meaningful and required further analysis. Moreover, the data required significant editing and data munging since it was not initially insightful and required additional data and variable interaction to gain more insights.

To overcome these issues, I first categorized each variable into numerical and categorical to get a sense of which were redundant and which were relevant to my analysis. I also combined some variables to extract more information. Additionally, I added weather and location data to the dataset to obtain more insights into the variables of interest. By doing so, I was able to mitigate the impact of irrelevant data, focus on meaningful correlations, and extract valuable insights from the data.

II. Related Work/Background

Based on my preliminary research, while there is a significant body of research on emergency calls and factors that impact them in general, I was not able to find many resources that specifically address this topic for San Francisco. However, I did come across an IEEE conference paper titled "Scalable Real-time Prediction and Analysis of San Francisco Fire Department Response Times" [1], which examines the same dataset for San Francisco Fire Department calls and analyzes response times. Although the paper primarily focuses on the methods employed to scale the process and obtain the results, the discussion of the findings is relatively brief, leaving ample room for further exploration. Thus, I believe there is still much to be discovered and analyzed with this dataset that has not yet been thoroughly researched.

III. Solution and Scalability

I organize my solution into three parts: R Development, Python Jupyter Notebook Development, and Python Spark Development. I began with R because I was familiar with the language and wanted to perform exploratory data analysis and create some visualizations, which is what R excels at. However, Python offered better tools for analyzing complex machine learning models, such as SciKit Learn and

TensorFlow, and was a bit easier for me to use to add additional data to the dataset. While my code was manageable and able to handle the dataset, it became clear that as the dataset grew and included more data, both approaches would struggle. This seems to indicate that these approaches might be able to handle a few Gigabytes of data, but may not be able to handle Terabytes or more. To ensure scalability in the future, I incorporated Python with Spark into my solution. This allowed for continued scalability and efficient processing of large datasets.

"Spark is a general-purpose distributed data processing engine" [2] that evolved from MapReduce, aiming to overcome its limitations and offer enhanced resilience and performance. Similar to MapReduce, Spark is designed for computational clusters, where tasks are distributed across the dataset and executed in parallel for optimal performance. Spark leverages a Resilient Distributed Dataset (RDD), which allows data to be processed in parallel and provides resiliency and performance benefits. While I have performed some analysis using Spark and started migrating my code to it, I have not yet completed the migration of all my code. Completing this migration would make my application more robust and scalable for future research. My focus has been more on learning machine learning with SciKit Learn and utilizing Tableau to create visualizations of my findings.

To improve the scalability of my data, I could consider implementing a system similar to the Google File System [3]. This system is capable of storing large amounts of data across multiple servers, allowing for efficient data access and analysis. Its fault tolerance and reliability are achieved through data replication across multiple servers, and it is designed for parallel processing, making it ideal for distributing data processing across multiple machines to speed up analysis tasks. However, implementing such a system would require the use of Google Cloud or a distributed cluster, which may incur additional costs. Nonetheless, it would provide a more resilient and performant storage solution for my data than what is currently utilized.

IV. Implementation

The programming languages I utilized in this project are mainly R and Python. The reason I utilized these two languages is because they are excellent languages for data analysis, and include plenty of tools for visualizations, machine learning, correlation, and exploratory data analysis.

For R I utilized the tidyverse package, which "is an opinionated collection of R packages designed for data science" [4]. I utilized this package to be able to do exploratory data analysis, correlation testing, and visualizations for my dataset. Some of the features I utilized of the tidyverse package include ggplot, dplyr, and cors for data munging and correlation testing.

Additionally, I leveraged Python for my data analysis, as it offers a vast array of packages and tools for Machine Learning development, such as SciKit Learn and TensorFlow. SciKit Learn is a free and open-source machine learning library that offers numerous algorithms and methods to analyze my data effectively. On the other hand, TensorFlow [5] is a powerful machine learning system that employs a data flow graph, which facilitates easy parallelization of computation across multiple CPUs or GPUs, thereby enhancing scalability and high-performance computing. Apart from these machine learning packages, I employed pandas for efficient data storage and management, while Seaborn with Matplotlib was used for creating insightful and impactful visualizations of my research findings.

To enhance my data analysis, I utilized three main supervised machine learning methods: Logistic Regression, Decision Tree classifiers, and Linear Regression. The primary objective of these methods was to predict the outcome variable given input data for a specific row in the dataset using a generated model. In contrast to unsupervised learning, where the model makes its own decisions without any expected output, we provided the algorithm with pairs of inputs and desired outputs, allowing it to discover a method to produce the desired output for a given input. With these models, I was able to predict outcomes for categorical and quantitative variables within the dataset. After creating these models, I analyzed the coefficients assigned to each input variable, determining the most prominent coefficients in determining the output. This provided valuable insight into the variables that influenced specific outcomes the most, which could help us focus on mitigating specific outputs in the future.

To ensure the accuracy of my machine learning analysis, I split the data into two parts: training data and testing data. The training data is used to train the model, while the testing data is used to evaluate the accuracy of the model when presented with new data. This helps to ensure that the model is not overtrained and that it has a certain level of accuracy in predicting outcomes. If the model has a low accuracy on the testing set, it indicates that the model is not accurate in real-life situations. On the other hand, if the model has high accuracy on the training set but low accuracy on the testing set, it suggests that the model is overfitted, and adjustments may need to be made to the coefficients or the balance between the training and testing sets. By using both training and testing data, we can achieve a more accurate and reliable machine-learning model.

Finally, I utilized Tableau to perform a thorough analysis of the dataset and generate insightful visualizations. The primary reason for choosing Tableau was its ability to create visually compelling and interactive dashboards that effectively communicate complex data. By leveraging Tableau's extensive features and functionalities, I was able to

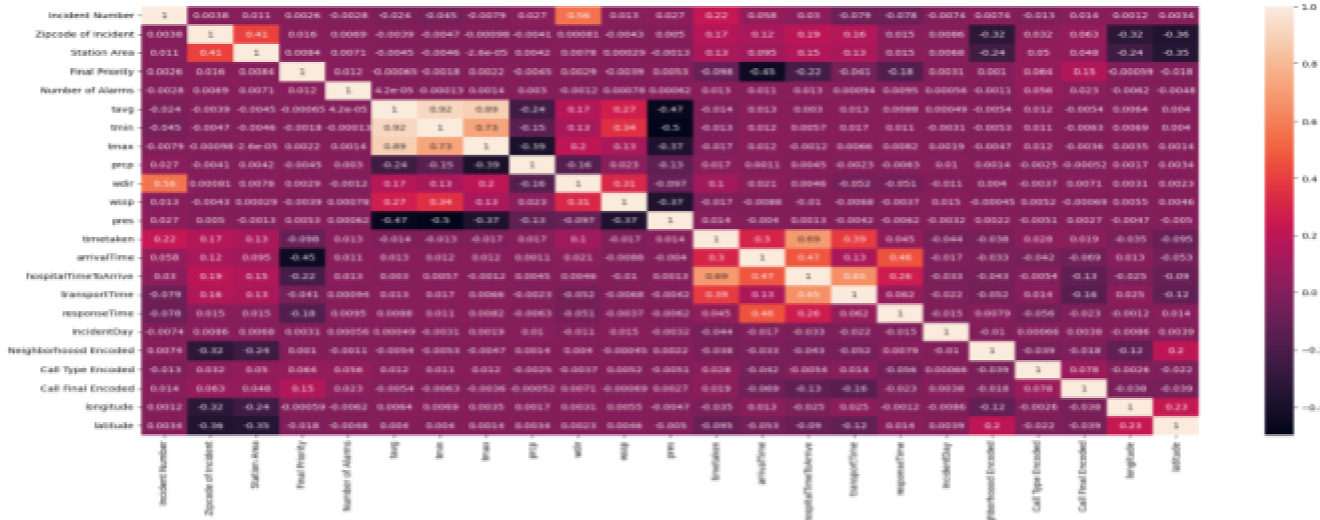

Figure 1: Correlation Results

showcase the key findings and trends in the data in a more accessible and engaging manner, which can greatly influence the audience's understanding [6].

V. Results and Observations

One of my primary goals with the data was to explore the relationships between different variables and identify interesting connections. To achieve this, I performed a correlation analysis, and the results can be seen in Fig 1. The analysis revealed that there are positive correlations between the Zipcode and the area of an incident and the time taken, response time, transport time, and time to arrive at a hospital. This suggests that location may have an impact on the time it takes for an incident to be resolved or taken care of. It could indicate that some areas struggle more with resolving issues, or that additional resources are required to improve efficiency. On the other hand, the analysis also revealed negative correlations between priority and response/efficiency. This means that as the priority of an event increases, the response time and time to fix the issue decrease, indicating that the fire department acts faster to resolve higher-priority events.

During my analysis, I was particularly interested in understanding the impact of the number of Alarms on the response time of the units and the availability of the resources. To achieve this, I conducted a series of experiments, and here are some of the key results that I found.

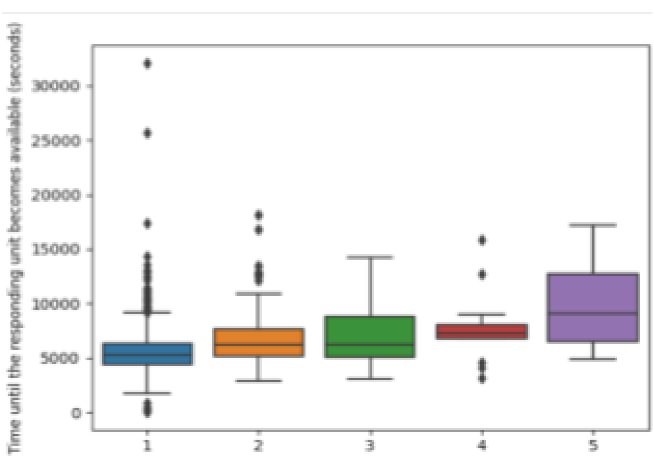

Figure 2. Alarms vs Time unit becomes available again

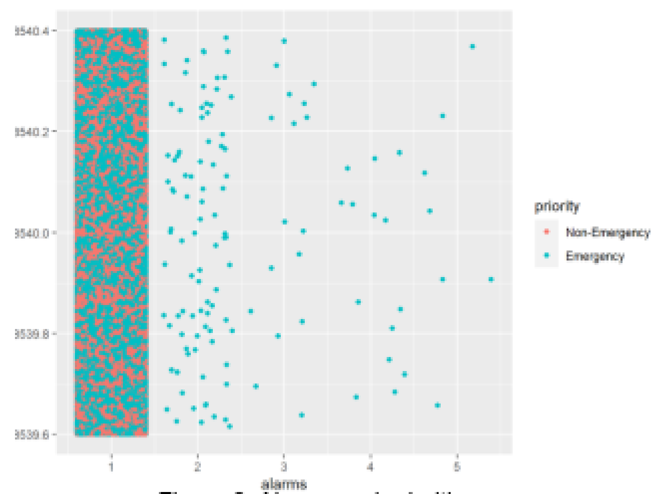

Figure 3. Alarms and priorities

As we can see in Figure 2, as the number of alarms increases at an incident, the longer it takes for a unit that responds to that incident becomes. There seems to be a positive correlation between these two variables. Albeit there is no direct relationship as we can see in our correlation testing, it is interesting to note that increasing the number of alarms could help us predict the time taken at an incident. As we can see in Figure 3, this could be largely due in part to the fact that most if not all of the higher number of alarm incidents are emergency incidents, which would indicate that these incidents are getting more care overall.

Figure 4. Number of units VS last arrived

Another set of analyses that I performed was analyzing the number of units that arrive at an incident and the time it takes for all units to arrive and to be done with the incident.

Figure 5. Number of units VS time until last is available again

As we can see from Figures 4 and 5, the number of units that end up responding to an incident tends to increase the amount of time that it takes for every unit to arrive, and for those units to become available again. The data further shows that this trend is most prevalent in outside fires, which may require more units and may explain why they take more time at a scene and more time to become available again, while other incidents require fewer units and hence would generally be faster to responding and being done with an incident.

Another interest I had with the dataset was to measure the effectiveness of the Fire Department over the years and to check whether there has been a steady increase in efficiency over time or a decrease.

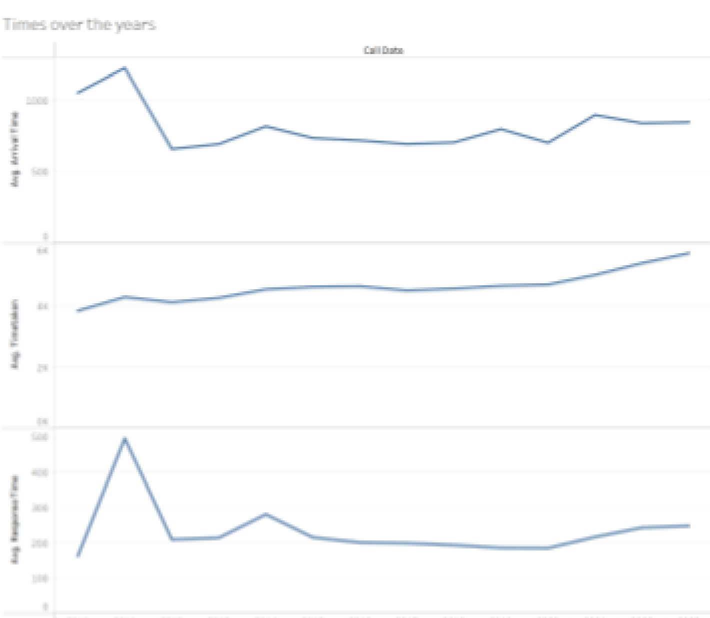

Figure 6. Times VS Years

From my analysis of the data that we can see in Figure 6, is that there was a steady decline in the amount of time to respond and resolve issues in the past, however as 2020 began, the times to respond and resolve issues tended to increase. This could coincide with Covid, but would require further analysis to ensure that is the case between the correlation, however, it is an interesting measure to note and be aware of to try to increase efficiency going forward.

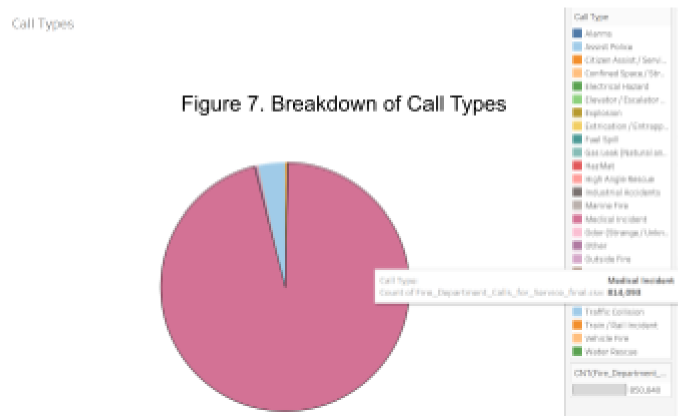

Figure 7. Breakdown of Call Types

One noteworthy observation that emerged from my analysis was the distribution of call types in the dataset. Figure 7 depicts a pie chart illustrating that the majority of calls in the dataset are medical response calls, indicating that medical incidents are the most frequent types of incidents that the San Francisco Fire Department receives. This finding raises concerns regarding the medical services and ambulance system in San Francisco, as it seems that too many medical incident calls are directed to the Fire Department as the first responders. The high volume of medical incidents may imply that the ambulance services in San Francisco need improvement to ensure timely response rates, and not rely heavily on the Fire Department as the first line of defense. I carefully filtered out the medical incident calls to examine the results for other call types, as this finding could affect the outcomes of other analyses.

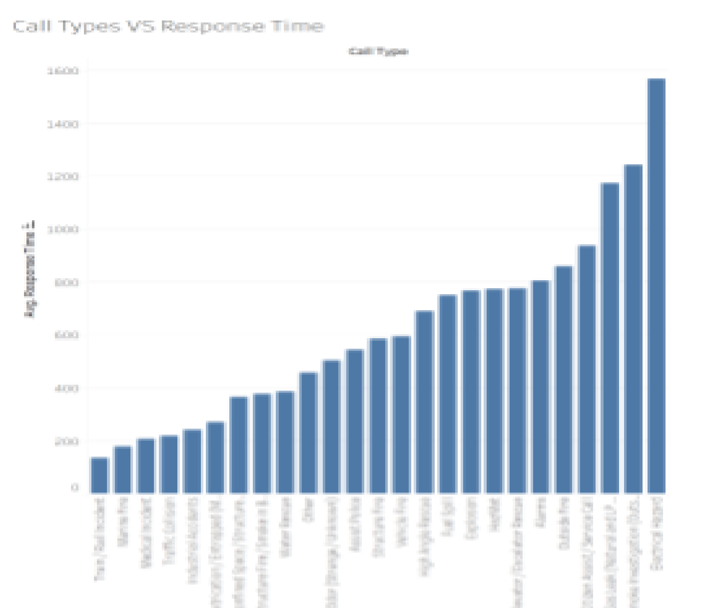

Figure 8. Call Types VS Response Time

Another analysis that I did was a comparison of the Call types against the average response times. As we can see in Figure 8, the highest response times are associated with electrical hazards, which could be because it is less important and more complex to handle, while the one with the shortest response time is Train/Rail incidents, which could be at the highest priority and hence requires the quickest and most urgent response. Some interesting observations to note is that most fire incidents seem to have a lower response time, except for Outside fires which tend to have the highest time that it takes on average to respond to. This could be because outside fires could be harder to detect since there is no built-in alarm system to alert the fire department of the danger right away, and may require a bystander to alert the fire department, or for it to get out of hand. This seems to indicate that better care and improvements could be made to detect outdoor fire incidents so that a quicker and more efficient response can be made.

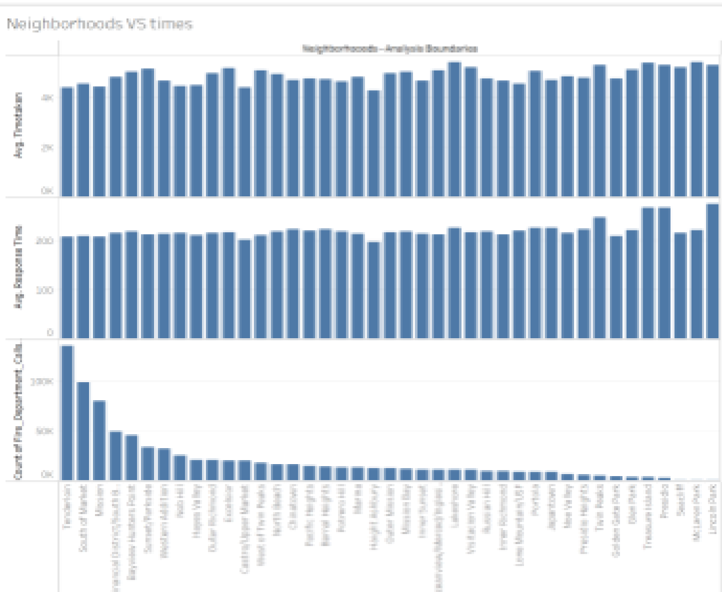

Figure 9. Response times VS Neighborhood

During my analysis, I also looked at how response times vary across different neighborhoods. The results showed that response times for incidents don't change significantly between neighborhoods with a high incidence rate. However, in neighborhoods where incidents are infrequent, the response times are generally higher. This suggests that some areas may lack the necessary resources to respond to an incident quickly. Therefore, allocating more resources to these areas could help to reduce response times and improve overall emergency services. Furthermore, it's worth noting that these areas tend to be sparsely populated and have difficult terrain, which may also contribute to longer response times.

When I tried to find which variables tend to influence times the most, I utilized linear regression and this is my findings

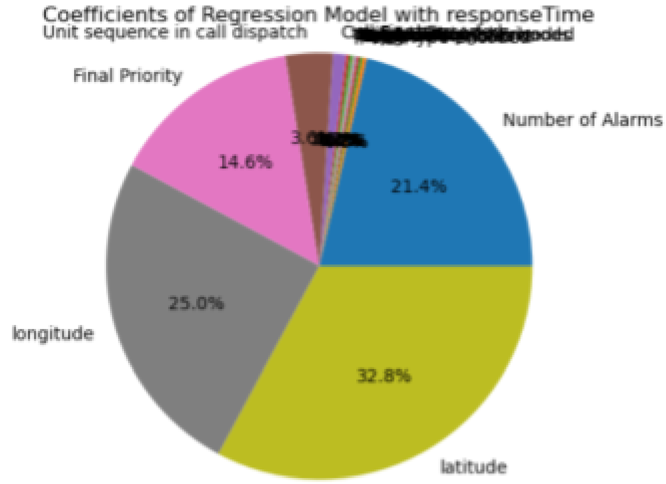

Figure 10. Response Time coefficients

As we can see in Figure 10, the response time for an incident seems to be most influenced by the location of the incident, the number of alarms, and the final priority. In my analysis, the location tends to be a positive correlation, while the number of alarms and final priority tend to be negative, meaning that as the number of alarms and priority grows, the response times tend to be smaller.

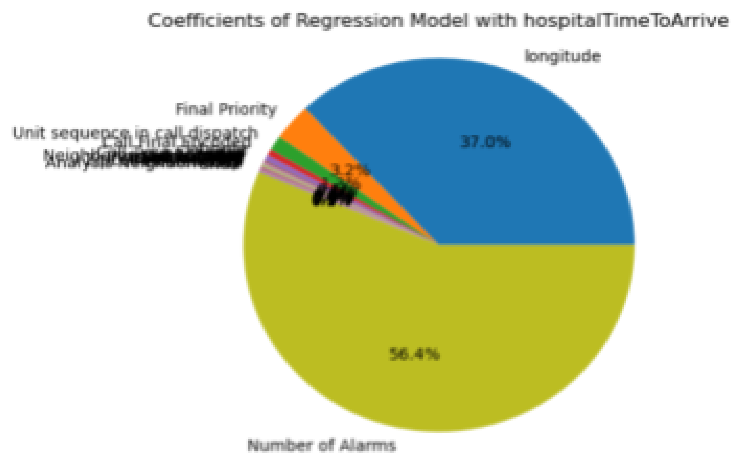

Figure 11. Coefficients for Hospital Time To Arrive

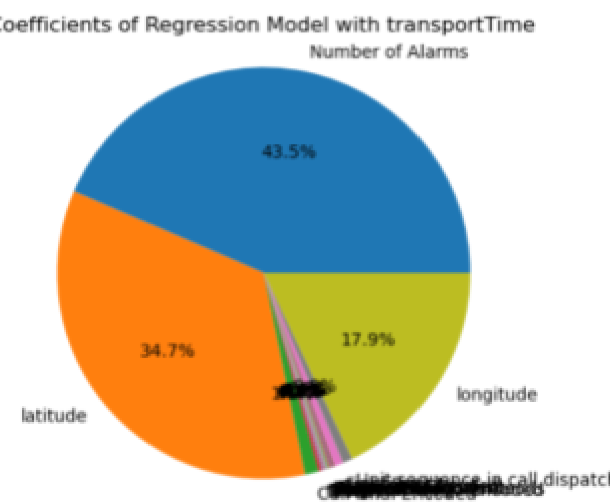

Figure 12. Coefficients for Transport Time

Figures 11 and 12 demonstrate that the time it takes for a unit to transport a person to the hospital from start to finish is similar to what we had seen with response time. These results are the same for these two variables since hospital time to arrive and transport time are interrelated. Hospital time to arrive is the time taken from the start of the call to arrival at the hospital, while transport time is the time taken by a unit to reach the scene and arrive at the hospital.

As for the Decision Tree Analysis I had done, I mainly focused on whether or not I can determine the “Call Final Disposition” and “Call Type” given the input we had.

```
Accuracy: 0.88
Precision: 0.88
Recall: 0.88
F1 score: 0.88

These are the coefficients:
transportTime 0.11305719631201507
timetaken 0.10793173505365013
hospitalTimeToArrive 0.09841644996460602
latitude 0.07704162595827549
longitude 0.0628945691664242
```

Figure 13. Decision Tree Results for predicting “Call Final Disposition”

As we can see in Figure 13, these are the results of the analysis with Final Disposition. Based on our decision tree classification analysis, we achieved an accuracy rate of around 90% in predicting the final disposition of an incident, which refers to the code that describes the outcome of the call. For example, TH2 indicates Transport to Hospital - Code 2, and FIR means Resolved by Fire Department. Our analysis also demonstrated higher recall and F1 scores, with recall being the ratio of true positive results to the total actual positive results, and the F1 score being the harmonic mean of precision and recall, with scores ranging from 0 to 1. These scores are crucial as they assess the model's performance in identifying both positive and negative classes, which can help in assessing its overall effectiveness. Furthermore, the primary predictors in this analysis were transport time, time taken, hospital time to arrive, and location. This indicates that we can predict the final disposition of an incident with a reasonable degree of accuracy using the incident's response times and location. This suggests that 1) certain areas may have a higher likelihood of a particular final disposition and that 2) it is easier to determine the final disposition based on the time taken to resolve the incident, as different dispositions may require different amounts of time to resolve.

```
Accuracy: 0.93
Precision: 0.93
Recall: 0.93
F1 score: 0.93

These are the coefficients:
responseTime 0.08550472986449993
longitude 0.0697452393506663
hospitalTimeToArrive 0.06764896951883766
Unit sequence in call dispatch 0.066351362613
arrivalTime 0.06350553777821745
timetaken 0.05736123173937946
transportTime 0.05399832760829157
```

Figure 14. Decision Tree Results for predicting “Call Types”

The analysis of the decision tree results for predicting call types in Figure 14 demonstrates an accuracy of over 90% and an F1 score of over 90%, indicating a high level of precision. Notably, the times related to the incident had the highest contributing coefficients. This suggests that utilizing the response and resolution times for incidents could be effective in predicting the incident type. Additionally, it appears that the type of incident strongly influences the amount of time required for resolution, providing valuable insights that can be used to optimize the process and increase efficiency.

VI. Conclusion

In conclusion, this analysis of the San Francisco Fire Department data has shown that there is a wealth of information that can be gained from this dataset. Using machine learning with Decision Tree classifiers and Linear Regression models, along with exploratory data analysis methods and visualizations, we were able to make several interesting observations. We found that the location of an incident can affect the response times and disposition of a call and that outdoor fire calls tend to take the longest time for firefighters to resolve. Additionally, we discovered that the Fire Department's response times have not improved since 2020 and may require further investigation. The results obtained from Decision tree classifications were highly accurate, suggesting that this model could be further developed to predict the final disposition and types of incidents in specific areas. This project has demonstrated the usefulness of visualizations and data analysis in extracting insights from large, complex datasets. Overall, this analysis highlights the importance of continuous monitoring and data-driven decision-making to improve the efficiency and effectiveness of emergency response services.

References

[1] X. Lian, S. Melancon, J. -R. Presta, A. Reevesman, B. Spiering and D. Woodbridge, "Scalable Real-time Prediction and Analysis of San Francisco Fire Department Response Times," 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of

People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), Leicester, UK, 2019, pp. 694-699, doi: 10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00154.

[2] M. Zaharia et al, "Spark: Cluster computing with working sets." HotCloud, vol. 10, (10-10), pp. 95, 2010.

[3] Ghemawat, Sanjay, Howard Gobioff, and Shun-Tak Leung. "The Google file system." Proceedings of the nineteenth ACM symposium on Operating systems principles. 2003.

[4] "Tidyverse," RStudio, 2017. [Online]. Available: https://www.tidyverse.org/. [Accessed: February 1st, 2023]

[5] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. TensorFlow: a system for large-scale machine learning. In Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation (OSDI'16). USENIX Association, USA, 265–283.

[6] Segel, Edward and Jeffrey Heer. “Narrative Visualization: Telling Stories with Data.” IEEE Transactions on Visualization and Computer Graphics 16 (2010): 1139-1148.