National College of Ireland

B.Sc. (Hons) in Technology Management

May 10th, 2020

Analysis of Dublin Fire Brigade and FDNY Ambulance Responses

Carl O'Beirne

X16326186

X16326186@student.ncirl.ie

Supervisor: Dr. Eugene O'Loughlin

Technical Report

Contents

Table of Figures

Table of Tables

Declaration Cover Sheet for Project Submission

Section 1: *Student to complete*

Name: Carl O'Beirne
Student ID: X16326186
Supervisor: Dr. Eugene O'Loughlin

Section 2: *Confirmation of Authorship*

The acceptance of your work is subject to your signature on the following declaration:

I confirm that I have read the College statement on plagiarism (summarised overleaf and printed in full in the Student Handbook) and that the work I have submitted for assessment is entirely my own work.

Signature:  Date: 08/05/2020

Carl O'Beirne

Abstract

The time it takes for an ambulance to respond to an incident is crucial. In some cases, their speed is the difference between life and death, so it is important there is no delay. Understanding what factors contribute to a delay is important for the ambulance service so they can adapt and allocate additional resources when required. This analysis looked to discover trends and patterns in the number of calls that are being made to the Dublin Fire Brigade (DFB) ambulance and Fire Department of New York (FDNY) Emergency Medical Service (EMS), drilling down on the individual quarters, months, weeks, days & hours of the day, as well as their response time, locating where the calls are coming from and look to predict if there was a delay in assigning an ambulance in New York City (NYC) as soon as the call was made. Results from the drill down analysis found not much insights could be gathered from quarterly, monthly & weekly analysis showed constant fluctuation, potentially caused by the increase in number of calls, especially in NYC, where the number of calls increased by almost 68 thousand calls in 2018. Much more insight was found in the day and hour, where it was discovered that the weekend was the busiest for Dublin and the quietest for NYC. None of the machine learning algorithms reached anticipated results, though found the Radial Grid Tuned Support Vector Machine (SVM) performance was the best of the selection used.

1. Introduction

1.1. Motivation

The idea of analysing the calls that are being made to the ambulance service came to mind after negative experiences of requesting an ambulance. I felt that the length of time that I had to wait on two occasions of having to request an ambulance was not adequate. The 1st time I had to request an ambulance was for a family member who had an accident with an electric tool, causing extensive damage, and was left waiting in excess of 30 minutes for the ambulance to arrive while the person was in severe pain.

The 2nd time that I had to call an ambulance was for an elderly neighbour, who had an accident late at night and was left unable to walk. The neighbour managed to make their way back into their house to seek attention by banging on a wall. We went to help the neighbour, and all agreed the best option was to get an ambulance and go to hospital. I made the call to request the ambulance, which took a follow up phone call and 2+ hours of waiting for the ambulance to arrive.

These two experiences were the motivation behind the decision to analyse the response of ambulances.

1.2. Literature Review

According to the Health Service Executive (HSE) (2016), the DFB operate crews of 2 in their 12 emergency ambulances annually, with an additional 4 in reserve in the event of unforeseen circumstances, across 14 fire stations in the County of Dublin, with an exclusion of most the Dun-Laoghaire Rathdown (DLR) region. The DLR region would be where the National Ambulance Service (NAS) would have jurisdiction. In comparison, the FDNY have over 600 ambulances with a target of 450 in service each day (NYC.gov, 2020) in NYC that responded to 4.7 million calls between 2013 and 2016 (Courtemanche, et al., 2019), almost 1.18 million calls per year. On the other hand, in Ireland, the NAS are attending over 300 thousand calls per year (HSE, 2017), while DFB respond to almost 80 thousand per year (see figure 4.1).

Ultimately, the ambulance service can only respond to the amount of calls within the capacity of their resources. The numbers of people that are using the allocated resources, do so inappropriately. A report found that more than 15% of the calls that the London ambulance service responded to could be declared inappropriate (Palazzo, et al., 1998). Related findings that cover other ambulance services show similar results that could end up delivering the inappropriate medical treatment (Mills, et al., 2019). However, it is not clear how inappropriate should be defined, with most paramedics preferring to use the word misuse (Dejean, et al., 2016). Therefore, the call dispatcher must fully

understand the intention of the call before prioritizing and allocating the resources, as well as determine the appropriate medical treatment for the patient prior to the ambulance arriving.

There are many factors that can contribute to the delay in ambulances, such as waiting times in A&E, heavy traffic, and sometimes even the weather. A study from Canada found that the crowding in emergency departments has contributed to the delay in ambulance responding (Schull, et al., 2003). In New South Wales, Australia, it was reported that 12.5% of 381 ambulance transports experienced a delay of 30-60 minutes & 5% had a delay ≥ 60 minutes which was caused by waiting for the patient to be transferred over to the hospital (Cone, et al., 2012). This is a global issue however, with the United States reporting the average waiting time in the emergency department of 5.8 hours (Trzeciak & Rivers, 2006) and only 63.3% of people experience a wait time < 6 hours contributing to only 58.4% of ambulances being out of the hospital in ≤ 30 minutes (HSE, 2019). Different weather conditions can be a contributing factor to a delay in ambulance response times, with (Zhan, et al., 2020) reporting that rain and temperature can influence ambulance responses and Dolney & Sheridan (2006) declaring a 10% rise in the number of calls for an ambulance in Toronto on hot days. The change in traffic flow throughout the day can impact the time it takes to get to the scene. Griffin & McGwin (2013), found that in Alabama, traffic congestion in contributed to an average 10-minute increase in the length of time it takes an ambulance to respond to an incident. With the roll out of a geospatial information system fitted into every ambulance, it could reduce the duration of time travelling to the hospital by up to 20% by determining the shortest route to the hospital on the go (Panahi & Delavar, 2009).

From the time the initial call is made, the ambulance service is on the clock to respond and provide medical treatment. The HSE declare a Key Performance Indicator (KPI) of 80% for all Echo & Delta calls must be in attendance within 19 minutes. In the latest quarterly report for July to September 2019 published by HSE (2019), the current Year to Date (YTD) for Echo calls is 79.7% (0.3% below) and Delta calls is 56.3% (23.7% below). Both call categories are life threatening and only 68% of the total life-threatening calls are meeting the KPI outlined. In 2014, the KPI for these calls to be responded to within 19 minutes was 95% (Lightfoot Solutions, 2015) and also had an additional measure of meeting the a KPI of 80% for Echo & Delta calls within 8 minutes. Between March and August, Lightfoot Solutions (2015) found that the total number of these calls that were responded to within 19 minutes was 67.2%, 27.8% below their 2014 KPI measure. In the 5-year time frame between these 2 reports, the overall performance has only increased by 0.8%, still far from achieving the KPI. This delay in responding to life-threatening calls can lead to fatalities, with Payne (2000) reporting that ambulance delays in rural Ireland could be the cause of up to 700 deaths a year. In 2015, the FDNY published their 3 year strategic plan, where they set a goal of amalgamate

their Fire and EMS service to “Enhance the FDNY’s ability to deliver emergency medical service” (FDNY, 2015). Within this plan, one of their short-term objectives was to bring additional resources to the areas that have the worst response times for the highest severity calls (FDNY, 2015).

Research carried out in Sweden found that predicted risk scores resulted in better decision-making about travelling to hospitals compared to human decisions (Spangler, et al., 2019). In another study prepared by Blomberg, et al. (2019), it was found that a machine learning algorithm identified more pre-hospital cardiac arrests on the emergency call than the dispatcher could. By obtaining chat transcripts between the caller and the dispatcher, machine learning and artificial intelligence could be introduced to make these decisions, as well as determine factors of the call that could indicate the level of severity (Young, et al., 2016), which is something that would massively benefit the ambulance service. Conclusively, the most valuable resource to the ambulance service is the initial information that is provided to the call dispatcher, and the use of machine learning prediction algorithms may contribute to faster, more effective decision making in pre-hospital care, improving the overall response and efficiency of ambulance services.

1.3. Aims & Objectives

From the review carried out, the information that is provided via the emergency call must be clear and describe the reason for the call in the most detail possible. The dispatcher on the phone must allocate the resources adequately and is required to prioritize the calls that need rapid medical attention. Therefore, it is important to analyse the type of emergency calls received, the number of calls, and how the ambulance service responds to them.

1.3.1. Objective 1 – Data Analysis of Ambulance Calls

To understand:

- The number of calls that are being responded to
    - Per year, Quarter, Month, Week, Day & Hour
- The type of calls the ambulance service are attending
- Where the calls originating from for both cities

1.3.2. Objective 2 – Response Time Analysis

To analyse:

- The length of time it took for the ambulance service to respond per criticality
- Potential factors that could be contributing to delays

1.3.3. Objective 3 – Predictive Analysis

Use Machine Learning to:

- Determine if there was a delay in assigning an ambulance immediately in NYC
- Compare 3 models
- Analyse which model has the better performance

1.3.4. Objective 4 – Interactive Dashboard

Produce an R Shiny Dashboard that will:

- Visualize an interactive map of where the calls originated from in NYC

2. Data Preparation

2.1. Methodology

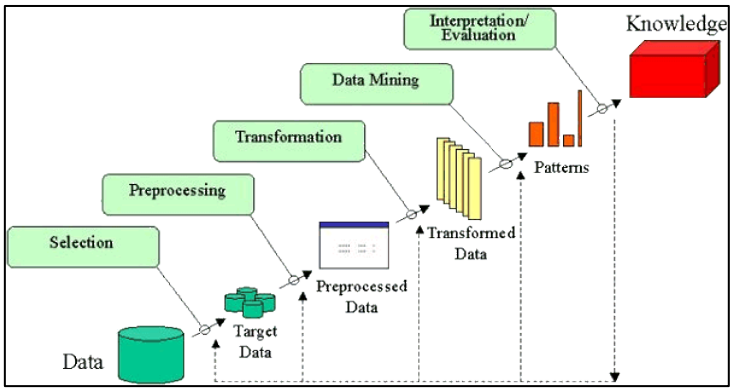


Figure 2.1 Steps of the KDD Process (Univeristy of Regina, n.d.)

There are several methodologies that could have been used, but the one methodology that suited best for the analysis in this project was KDD. KDD is the method of obtaining valuable information from large amounts of data (Fayaad, et al., 1996). As seen in figure 2.1 the process takes an iterative approach containing five main stages. They are:

1. **Selection** – at this point, it is all about reviewing all the different data that are available and looking to identify the target data that would aid in discovering the knowledge you would like.
2. **Pre-Processing** – now that the target data has been chosen, pre-processing of the data must be done. This is when you begin some data cleansing. Wherever applicable, there is the option of removing any outliers, dealing with time series and null values that are in the raw data (Fayaad, et al., 1996).
3. **Transformation** – although the data is cleaned, not all data is relevant for discovering knowledge. Some is useful for some knowledge discovery and the rest can be useful for other

knowledge. Transformation is about identifying that relevant data that will find that knowledge you hope to discover for answering a question. This can include removing irrelevant columns.

4. **Data Mining** – the relevant data has been chosen so now it's time to perform some data mining techniques such as classification, regression, summarization, or clustering algorithms to look for relationships in the data and obtain the knowledge that is required (Fayaad, et al., 1996).
5. **Interpretation / Evaluation** – this stage is all about taking the results from the data mining techniques that have been performed, understanding what it means, assess how well it performed, convert the results into the knowledge and display it in a way that users are aware how to recognize what it means (Fayaad, et al., 1996).

2.2. Technologies Used

The main technology that was used throughout the duration of this project is the programming language called ***R***. R is used to execute statistical and graphical outputs. It has been around since 1993 and was developed by Ross Ihaka & Robert Gentleman.

R Studio is an IDE application that is used in conjunction with the programming language R. It has been used throughout the duration of this project to perform some of my main statistical, descriptive & predictive analysis.

I wanted to make some of the analysis outputs that I generated from the data interactive, and I felt there was no better way to do this than to use ***R Shiny***, a feature built using R & R Studio. It allowed me to create dashboards that included the outputs that I generated and could be dynamically changed.

I decided to implement ***SPSS*** into my project for the statistical test that I performed on the data. The outputs that contained the test results was more visually appealing and generated them into compact tables that made it clear to interpret.

I applied the ***Microsoft Office Suite*** was used for a variety of reasons. ***Excel*** was used for storing the datasets, as well as performing some data manipulation prior to loading the data for analysis. For creating presentations for the project pitch, the midpoint presentation & final presentation, I went with ***PowerPoint***. The data, as well as all documents were stored on ***OneDrive***. Lastly, the thesis was written using ***Word.***

I introduced ***Tableau*** into my project to generate some attractive visualizations of the data. The business intelligence tool had a variety of visualizations to choose from by the click of a button, making it rather easy to get to grips with.

The most crucial technology that I used was ***GitHub***. Using GitHub allowed me to keep track of all changed that I made to my code and enables the option of rolling back any changes if ever there was an bug in the code that could not be identified, as well as it being a backup in the event of any failure.

2.3. Data Source

Since the idea involved analysing the ambulance service in both Dublin and New York, the data had to be obtained from 2 different sources. Starting with the DFB, we searched online and found a source which had data for 2013-2015 (Dublin City Council, 2016), though the data was very minimal. A list of data that we would like to be able to analyse was wrote out and sent to DFB. They replied a few days later, where they informed what information they could and could not disclose, and if happy, that they would happy to send a file with the data (Dublin Fire Brigade, Personal Correspondence [2019]). We accepted and that same day a file that contained the calls that the DFB ambulance had attended throughout 2017 & 2018 was sent.

As for the FDNY data, this was much easier to obtain, as the USA does not have data protection policies as strict as Europe has. A dataset on the New York City OpenData portal was located and then downloaded the excel file to see what data it had to offer. This data was the EMS Incident Dispatch Data (NYC OpenData, 2020) which had the calls that the FDNY ambulance service had attended to for the years 2008-2019. Now that the 2 datasets had been obtained, we could then proceed to review the data.

2.4. Data, Data Types & Data Descriptions

2.4.1. DFB Data Description & Type Summary

The information in table 2.1 is a summary of the data that was used in the DFB data for this analysis.

Table 2.1 DFB Ambulance Data Description Summary

Column Name	Description
Date	Date of Incident
Criticality_Code	Level of Severity
Hospital_Code	Adult A&E Catchment Area Dublin
IA_LS_Mins	Time in minutes between IA & LS
LS_AH_Mins	Time in minutes between LS & AH
AH_MAV_Mins	Time in minutes between AH & MAV
TOC_IA_Mins	Time in minutes between TOC & IA
LS_CD_Mins	Time in minutes between LS & CD
TOC_CD_Mins	Time in minutes between TOC & CD

It shows a breakdown of some of the variables and their variable type that are important to this assignment. In table 2.2, there is a summary of the same columns in the data description that outlines the data types for each one. The original dataset had 17 columns, where we used the data to create additional insights which will be discussed further in section 2.5.1. The full data description and data type tables are found in appendix 7.2 and 7.3 respectively.

Table 2.2 DFB Ambulance Data Type Summary

Column	Data Type
Date	Factor
Hospital_Code	Factor
Criticality_Code	Factor
IA_LS_Mins	int
LS_AH_Mins	int
AH_MAV_Mins	int
MAV_CD_Mins	int
LS_CD_Mins	int
TOC_CD_Mins	int

2.4.2. FDNY EMS Data Description & Type Summary

Table 2.3 shows variables important to this analysis from the FDNY EMS data. It is a summary of the variables and a short description of what the data in each of the columns mean.

Table 2.3 FDNY EMS Data Description Summary

Column Name	Description
INCIDENT_DATETIME	The date and time the incident was created in the dispatch system
FINAL_CALL_TYPE	The call type at the time the incident closes.
FINAL_SEVERITY_LEVEL_CODE	The segment(priority) assigned at the time the incident closes.
DISPATCH_RESPONSE_SECONDS_QY	The time elapsed in seconds between the incident_datetime and the first_assignment_datetime.
INCIDENT_RESPONSE_SECONDS_QY	The time elapsed in seconds between the incident_datetime and the first_on_scene_datetime.
HELD_INDICATOR	Indicates that for some reason a unit could not be assigned immediately
BOROUGH	The borough of the incident location.
ZIPCODE	The zip code of the incident.
LATITUDE	Co-Ordinate generated using ZIPCODE
LONGITUDE	Co-Ordinate generated using ZIPCODE

Additionally, in table 2.4, a list of the summarized data types can be found. A full data description can be seen in appendix 7.4 and 7.5 respectively.

Table 2.4 FDNY EMS Data Type Summary

Column	Data Type
INCIDENT_DATETIME	Factor
FINAL_CALL_TYPE	Factor
FINAL_SEVERITY_LEVEL_CODE	int
DISPATCH_RESPONSE_SECONDS_QY	int
INCIDENT_RESPONSE_SECONDS_QY	int
HELD_INDICATOR	Factor
BOROUGH	Factor
ZIPCODE	int
Longitude	int
Latitude	int

2.5. Cleansing

2.5.1. DFB Data

When the data was viewed for the first time, there were a lot of noticeable null values in the data that needed to be dealt with. We started to remove some in excel but changed to do it in R because it was much more efficient at removing nulls. Wanting to keep the time data as real as possible was the reason behind removing the rows rather than imputing using mice, as there was the possibility the imputed time could be incorrect. We were selective in what was removed, however, not removing any null values from the At Hospital time, as not all ambulances are required to go to the hospital, so removing nulls from this could potentially end up with some inaccurate results.

Date	Agency	DFB_Station	DFB_Station_ID	Hospital_Code	Hospital_Code_ID	District
0	0	0	0	347	0	270
Description	TOC	ORD	MOB	IA	LS	AH
0	0	0	3605	30757	65398	54588
MAV	CD	Criticality_Code	Criticality_ID	Cor_LONG	Cor_LAT	TOC_ORD_Time
12351	0	0	0	0	0	0
TOC_ORD_Mins	ORD_MOB_Time	ORD_MOB_Mins	MOB_IA_Time	MOB_IA_Mins	IA_LS_Time	IA_LS_Mins
0	3605	3605	30757	30757	65398	65398
LS_AH_Time	LS_AH_Mins	AH_MAV_Time	AH_MAV_Mins	MAV_CD_Time	MAV_CD_Mins	TOC_IA_Time
54588	54588	12351	12351	0	0	30757
TOC_IA_Mins	ORD_IA_Time	ORD_IA_Mins	LS_CD_Time	LS_CD_Mins	TOC_CD_Time	TOC_CD_Mins
30757	0	0	0	0	0	0

Figure 2.2 NA Values found in DFB Dataset Before Cleansing

Afterwards, we felt there was some more that was able to be done with the data, so saved the updated data file, and reopened it in excel. An excel formula was used to calculate the difference in minutes, the time between each stage of the call, as well as other times between different call stages that could have added value to the analysis, whether it would be used or not (*E.g. Difference in time between the time of call and the ambulance arriving at the scene*). An ID was added for the fire station area, hospitals & the criticality of each call to improve efficiency when doing statistical tests. Once back in R, the columns in the data that we felt added no benefit or would no longer be

used, such as the agency where they were all the same value and the OSI Irish Grid References for each call, were removed.

Lastly, there were a lot of outliers in the data, especially ones that indicated calls were exceeding 12 hours in duration. Looking closer at the data, the calls that exceeded 6 hours in some of the columns was due to an excel formula issue with the times, so this justified removing any calls that exceeded 6 hours total duration.

Date	DFB_Station	DFB_Station_ID	Hospital_Code	Hospital_Code_ID	District	Description
0	0	0	0	0	0	0
TOC	ORD	MOB	IA	LS	AH	MAV
0	0	0	0	0	1614	0
CD	Criticality_Code	Criticality_ID	TOC_ORD_Mins	ORD_MOB_Mins	MOB_IA_Mins	IA_LS_Mins
0	0	0	0	0	0	0
LS_AH_Mins	AH_MAV_Mins	MAV_CD_Mins	TOC_IA_Mins	ORD_IA_Mins	LS_CD_Mins	TOC_CD_Mins
1614	0	0	0	0	0	0

Figure 2.3 NA Values Removed from DFB Dataset Post Cleansing

Overall, the data we started with was 158,978 total calls between 2017 & 2018, with 17 columns. Having performed the data cleansing on the data, there was 65,121 calls for 2017 & 2018 remaining, a reduction of approx. 59% of all calls.

For the analysis that did not require the cleaned data, the raw dataset was used, however, when analysis was done on data relating to the times, where cleansing was necessary, then the cleansed dataset was used.

2.5.2. FDNY EMS Data

Opening the data in excel was not an option for the FDNY EMS data, due to its large number of calls, so R was the data cleansing tool yet again. Firstly, the date format in the incident date/time column was changed to R format so that all years except 2017 & 2018 could be removed. The reason for only wanting keep 2017 & 2018 was to remain consistent with the DFB data time frame. This reduced the number of calls by approx. 81.7%. After checking the data types for all variables, we noticed that there were some inconsistencies. For example, there was 6 levels in the factor for Borough, when in fact there are only 5 and the 6th one was classed as unknown, so this level was dropped. Other factor columns had 4 levels, True / False / Y / N, when there was only a need for two, so any data in these columns where True or False was found, was replaced with Y or N. There is a call description of UNKNOW which just meant the reason for the call was unknown. These were removed from the data where the final call type was unknown, so analysis on accurate call types can be done.

CAD_INCIDENT_ID	INCIDENT_DATETIME	INITIAL_CALL_TYPE	INITIAL_SEVERITY_LEVEL_CODE
0	0	0	0
FINAL_CALL_TYPE	FINAL_SEVERITY_LEVEL_CODE	FIRST_ASSIGNMENT_DATETIME	VALID_DISPATCH_RSPNS_TIME_INDC
0	0	128317	0
DISPATCH_RESPONSE_SECONDS_QY	FIRST_ACTIVATION_DATETIME	FIRST_ON_SCENE_DATETIME	VALID_INCIDENT_RSPNS_TIME_INDC
0	164191	531245	0
INCIDENT_RESPONSE_SECONDS_QY	INCIDENT_TRAVEL_TM_SECONDS_QY	FIRST_TO_HOSP_DATETIME	FIRST_HOSP_ARRIVAL_DATETIME
534209	531500	4800484	4891374
INCIDENT_CLOSE_DATETIME	HELD_INDICATOR	INCIDENT_DISPOSITION_CODE	BOROUGH
6626	0	146045	0
INCIDENT_DISPATCH_AREA	ZIPCODE	POLICEPRECINCT	CITYCOUNCILDISTRICT
0	372012	371410	375121
COMMUNITYDISTRICT	COMMUNITYSCHOOLDISTRICT	CONGRESSIONALDISTRICT	REOPEN_INDICATOR
371420	377585	375121	0
SPECIAL_EVENT_INDICATOR	STANDBY_INDICATOR	TRANSFER_INDICATOR	
0	0	0	

Figure 2.4 NA Values found in FDNY EMS Data before Cleansing

It was time now to reduce the data more, this time by removing rows in the data that had a null value in any of the columns. This was completed by using the complete.cases() function in R. Unlike the DFB data, there were no nulls in the data for time arriving at the hospital, so did not have to be specific in what columns we had to keep nulls for. The reason for not using imputations for this can be seen in section 5.3. After the rows with null values were removed, it was noticed that columns that had 2 factors, Y & N for determining if the call dispatch and travel time was within target was all down to just Y after the nulls were removed, so there was no analysis that could be carried out on this data, meaning these columns were removed from the data entirely. Once the data cleansing was complete, the data was saved in a new file so the data loading time would be significantly reduced (see section 5.3).

To achieve one of the objectives, we wanted to display a map of the calls that were made in NYC, though did not have co-ordinates to be able to do so. A package which contained a dataset of all Zip codes across the United States of America (USA) on the R CRAN repository (which has since been removed) called Zipcode. This dataset listed all the zip codes, as well as the latitude and longitude of each zip code.

CAD_INCIDENT_ID	INCIDENT_DATETIME	INITIAL_CALL_TYPE	INITIAL_SEVERITY_LEVEL_CODE
0	0	0	0
FINAL_CALL_TYPE	FINAL_SEVERITY_LEVEL_CODE	FIRST_ASSIGNMENT_DATETIME	DISPATCH_RESPONSE_SECONDS_QY
0	0	0	0
FIRST_ACTIVATION_DATETIME	FIRST_ON_SCENE_DATETIME	INCIDENT_RESPONSE_SECONDS_QY	INCIDENT_TRAVEL_TM_SECONDS_QY
0	0	0	0
FIRST_TO_HOSP_DATETIME	FIRST_HOSP_ARRIVAL_DATETIME	INCIDENT_CLOSE_DATETIME	HELD_INDICATOR
0	0	0	0
INCIDENT_DISPOSITION_CODE	BOROUGH	INCIDENT_DISPATCH_AREA	ZIPCODE
0	0	0	0
POLICEPRECINCT	CITYCOUNCILDISTRICT	COMMUNITYDISTRICT	COMMUNITYSCHOOLDISTRICT
0	0	0	0
CONGRESSIONALDISTRICT	REOPEN_INDICATOR	SPECIAL_EVENT_INDICATOR	STANDBY_INDICATOR
0	0	0	0
TRANSFER_INDICATOR			
0			

Figure 2.5 NA Values Cleansed from FDNY EMS Dataset

An iterative statement (for loop) was created that would iterate through the FDNY EMS data frame and for each of the zip codes in the main data frame, it would compare it to the zip codes in the reference data frame, and where it found a match, it created a new column for latitude and longitude and would add the co-ordinates for each of the calls. It was decided to create a sample of 80000 records to run the loop on due to hardware limitations (see section 5.3).

```
for (i in 1:nrow(NYC_EMS_MapSample)){
    if(length(zipcode$zip[NYC_EMS_MapSample$ZIPCODE[i] == zipcode$zip]) == 1){
        NYC_EMS_MapSample$Latitude[i] <- zipcode$latitude[NYC_EMS_MapSample$ZIPCODE[i] == zipcode$zip]
        NYC_EMS_MapSample$Longitude[i] <- zipcode$longitude[NYC_EMS_MapSample$ZIPCODE[i] == zipcode$zip]
        print(paste("Row: ",i, "Zip Code: ",NYC_EMS_MapSample$ZIPCODE[i],NYC_EMS_MapSample$Latitude[i],NYC_EMS_MapSample$Longitude[i],
"Status: ", TRUE))
    }else{
        NYC_EMS_MapSample$Latitude[i] <- NA
        NYC_EMS_MapSample$Longitude[i] <- NA
        print(paste("Row: ",i, "Zip Code: ",NYC_EMS_MapSample$ZIPCODE[i],NYC_EMS_MapSample$Latitude[i],NYC_EMS_MapSample$Longitude[i],
"Status: ", FALSE))
    }
}
```

Figure 2.6 For Loop to obtain Zip Co-Ordinates

Previously mentioned at the end of the DFB data cleansing, the data used for the analysis that was not using variables that were cleansed, an alternative version of the file was saved prior to the data cleansing that was used. The analysis that did require NA values to be dealt with, then the new, cleansed dataset was used.

3. Implementation

3.1. R Functions & Packages Used

Table 3.1 R Functions Used

R Functions		
data.frame()	set.seed()	subset()
str()	sample()	gsub()
complete.cases()	randomForest()	droplevels()
for()	predict()	leaflet()
fread()	confusionMatrix()	if()
fwrite()	sapply()	read.csv()
write.csv()	naiveBayes()	ovun.sample()

Some functions that were used throughout the duration to perform processes such as reading & writing data, creating samples of data, running and evaluating machine learning algorithms, and understanding what the data types are and where the null values were in the data are all listed in table 3.1. The functions were used in conjunction with the packages listed in table 3.2.

Table 3.2 R Packages Used

R Packages		
Data.table	randomForest	tidyverse
Zipcode	ROSE	E1071
Caret	Leaflet	mlbench
shiny	shinyDashboard	

3.2. Machine Learning Algorithms Used

Machine learning was used to determine if we could predict whether an ambulance from the FDNY EMS division was held up (not able to be assigned to a call immediately). Three classification machine learning algorithms were used head to head to determine which of the algorithms performed the best in predicting the variable. The algorithms being used are Random Forest, Naïve Bayes & Support Vector Machines (SVM). For SVM, three different variations were used to determine which was best fit, Linear Grid Tuned, Radial Grid Tuned & Radial Random Tuned.

3.2.1. Random Forest

Random Forest is a supervised, decision tree, classification machine learning model (Schott, 2019). The data that was used for the random forest prediction of whether an ambulance was held up was sample of the main dataset using the seed 546513 for reproducibility. The random sample consisted of 300,000 records and 29 variables. Some variables were removed from the sample that either too closely linked to the model, was unrelated to what was being predicted, or was a factor that consisted of more than 53 levels, which random forest did work with. This left us with 17 variables to help predict the dependent variable.

The sample data was then randomly split using a seed of 325146 into a 75/25 proportion for the train and test dataset, respectively, offering us 225,000 calls to train the data and 75,000 calls to evaluate the model. The dependent variable was removed from the test data frame and stored in its own variable so that we could evaluate the prediction without the dependent variable being visible. The model was performed on the training data frame, using all the variables that were not removed from the sample data frame. Once the model completed, we began predicting the variable's using the test data which previously mentioned, excluded the dependent variable. Once complete, the results were evaluated using a confusion matrix against the dependent variable with the positive class of *"Y"* (see section 4.3.1).

As a class imbalance was found in the training data, we used Random Over Sampling Examples (R.O.S.E.) to introduce synthetic results to make each of the classes of the dependent variable even. A separate train and test data were created for R.O.S.E. using the same proportions, this time using a seed of 69745 to split them. The number of observations in the majority class was identified and we doubled the number using ovun.sample() formula to over sample the data, creating a new train data for the random forest model. Once again, we ran the random forest model and the prediction and evaluated the model in a confusion matrix once again with *"Y"* as the positive class (see section 4.3.2).

3.2.2. Naïve Bayes

Naïve Bayes is classification machine learning model that utilizes Bayes Theorem to make predictions on probability (Gandhi, 2018). Like the random forest model, a random sample using the seed 215145 of 300,000 calls was taken from the main data, split into the same 75/25 proportion for the train and test data, respectively. The seed used for this split was 234158. The same variables that were removed from the random forest model were taken out of the Naïve Bayes model also to keep the models the same, and the dependent variable was also removed from the test data and stored in its own variable.

When the data was ready, the Naïve Bayes model was performed on the held up dependent variable to train the data. When the model finished running, it was predicted using the test data. Once the prediction was completed, using a confusion matrix, with the positive class of *"Y"*, the model performance was evaluated (see section 4.3.3).

3.2.3. Support Vector Machine (SVM)

The 3rd and final machine learning algorithm that will be used to try and predict if the ambulance was going to be held up was SVM. It is another classification, supervised machine learning model that works with numeric data to distinctly classify data on a hyperplane (Gandhi, 2018). The seed 65451 was used for all three SVM kernels.

We first created a sample of 20,000 calls from the main FDNY EMS dataset, then removed the variables that were not numeric, which finally then broken down into a 75/25 train and test dataset, respectively. Three kernels of SVM are being used to identify which tuning process performs the best for the analysis, to see which of them returns the most accurate and reliable model. We first assigned the cost to be the 2 to the power of all integers between 1 & 8 to determine for the linear grid to see which parameter is best suited for the model. The first kernel, linear grid tuned, configured the control group to be a 10-fold cross validation method.

Once the control and grid are prepared, the first model was run on the training data to see what the best Cost parameter was best. When we found that cost 2 was the best for the model, the cost was refined further to see which was most optimal, a new tuning grid was configured, and the model was once again ran on the training data, then predicted the dependent variable and evaluated it in a confusion matrix (see section 4.3.4).

For the radial grid tuned, a larger cost was used to, and sigma was introduced to keep the width of the tuning grid within the normal distribution bell curve values. The model was run on the

training data once again, with the new tuning grid. We found which cost was the optimal one for this model and performed the prediction and evaluated it with a confusion matrix in section 4.3.5.

The last kernel we used was radial random tuned. A new control measure was created, which was also a 10-fold cross validation, however this time we added the parameter search and assigned it to random. The train model was run using the new control measure for random tuning and predicted, and finally, in section 4.3.6, it was evaluated.

4. Results

4.1. Ambulance Call Analysis

To understand the demand on the ambulance services, this analysis focuses on when the calls were made through years, before drilling down into the quarters, months, weeks, days and even hours of the day. We will be looking at the number of calls that were made during these periods, as well as what type calls people were making and where the calls originated from. The models work by

4.1.1. Number of calls Per Year

The workload that each of these services have dealt with has been far from easy, with figure 4.1 showing just how many calls both ambulance services had responded to Year on Year (YoY). With DFB operating just 12 ambulances for a population of 1.13 million people (Central Statistics Office, 2016) (excluding the Dun-Laoghaire Rathdown area where the DFB ambulance does not respond to unless there are no available NAS ambulances), they responded to 79,080 calls throughout 2017, while in 2018, they had increase of just 1.03% in the number of calls that they had responded to, totalling 79,898 calls for the calendar year.

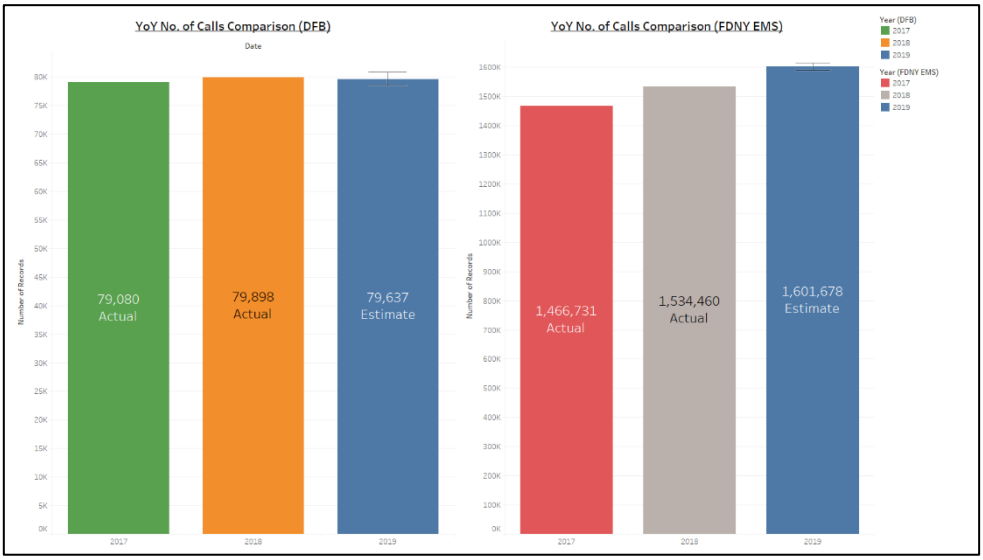


Figure 4.1 YoY Comparison with Forecast of No. of Calls for DFB & FDNY EMS

Meanwhile, across the Atlantic in NYC, the FDNY EMS had responded to a staggering 1,466,731 calls in 2017. While the population sizes between the 2 cities is significantly different, with NYC having a population of approx. 8.18 million (NYC.gov, 2010) according to the last census carried out in 2010. Like Dublin, the number of calls that the FDNY EMS responded to in 2018 had increased by 4.62% to 1,534,460 calls using roughly 450 ambulances to serve these calls.

Using the forecast tool built into Tableau, using exponential smoothing we were able to gain insights into the projections for the number of calls that each service would respond to in 2019. By using the data from 2017 & 2018, we were able to estimate that the number of calls that DFB would respond to in 2019 would be 79,637, a decrease of 0.33% on the previous year, while the FDNY EMS was forecasted to respond to 1,601,678 calls, an increase of 4.38% on the previous year. The data for DFB was not publicly available for 2019 so we could not verify how well the forecast performed. However, the FDNY EMS data was and the actual number of calls that they responded to in 2019 was 1,536,225. The forecast had overestimated by 4.26%, where the actual increase of calls was just 0.12%.

4.1.2. Number of Calls Per Quarter

Looking at the number of calls that are being made to both services per quarter on a YoY basis, seen in figure 4.2. The number of calls that DFB attended to in 2017 Q1 started off with 19,425 calls and had a small increase in the number of calls for each quarter following. Comparing the quarters YoY for DFB, we found that the number of calls received in each had increased, with Q1 increasing by 0.44% YoY, Q2 had an increase of 1.83% YoY, Q3 then had the greatest increase of 2.1% YoY, before Q4 had a YoY decrease in the number of calls by 0.18%. Although there is the decrease YoY for Q4, by adding the number of calls for each quarter together, Q4 was still found to be the busiest quarter of the years, responding to a total of 40,974 calls between October and December for 2017 & 2018.

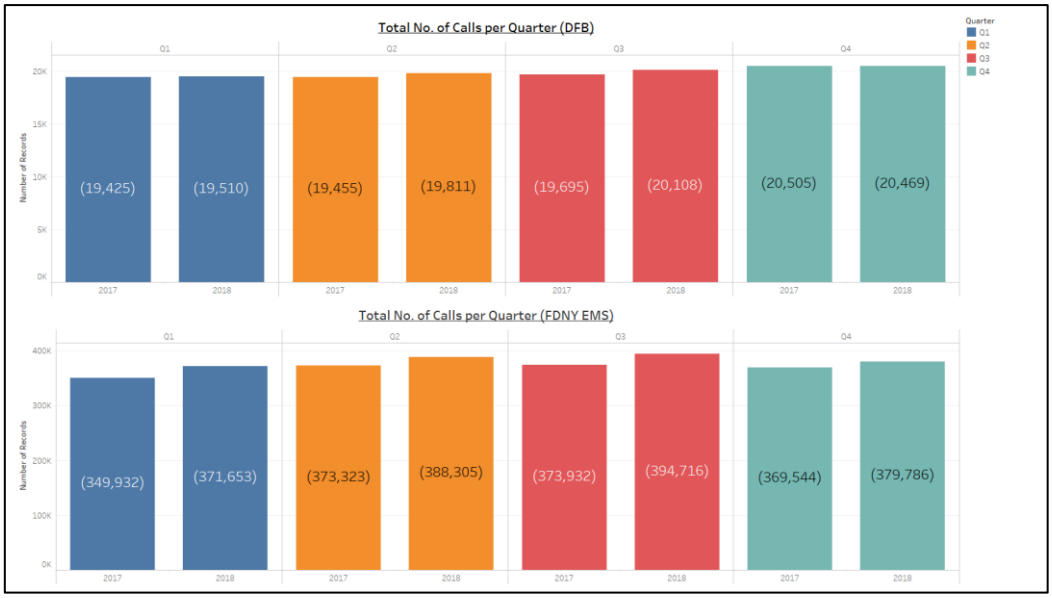


Figure 4.2 YoY No. of Calls by Quarter Comparison

Like DFB, it was discovered that the FDNY EMS showed a similar pattern where the number of calls taken in each quarter, except for Q4, where there was an increase rather than a decrease in the number of calls taken for both 2017 and 2018. When comparing the quarters YoY, we found that there was greater percentage increase for each of the quarters. In Q1, there was a YoY increase in the number of calls by 6.21%, a slightly smaller increase in Q2 of 4.01% YoY, increasing again to 5.41% YoY for Q3, before recording the smallest YoY value of 2.77% for Q4. Unlike the DFB, the busiest quarter for the FDNY EMS was Q3, responding to a total of 768,646 calls between July and September for 2017 & 2018.

4.1.3. Number of Calls Per Month

Drilling down into the numbers even further, looking at the number of calls that the services are responding to every month for the two years combined. We found that the busiest month of the year for DFB for the years two years was December. The difference in the number of calls for the busiest and 2nd busiest month for the service, July, was 7.38%

Other months of the year recorded approximately the same number of calls, though it had a constant fluctuation. For every month of the years that had a high number of calls, the month that followed it almost always decreased, except for September where it decreased for a 2nd month in a row. As expected, February recorded the least number of calls, as there is only 28 days in the month. However, had there been 31 days in the month, the calls could have been as high as approx. 13,300 per month for the 2 years.

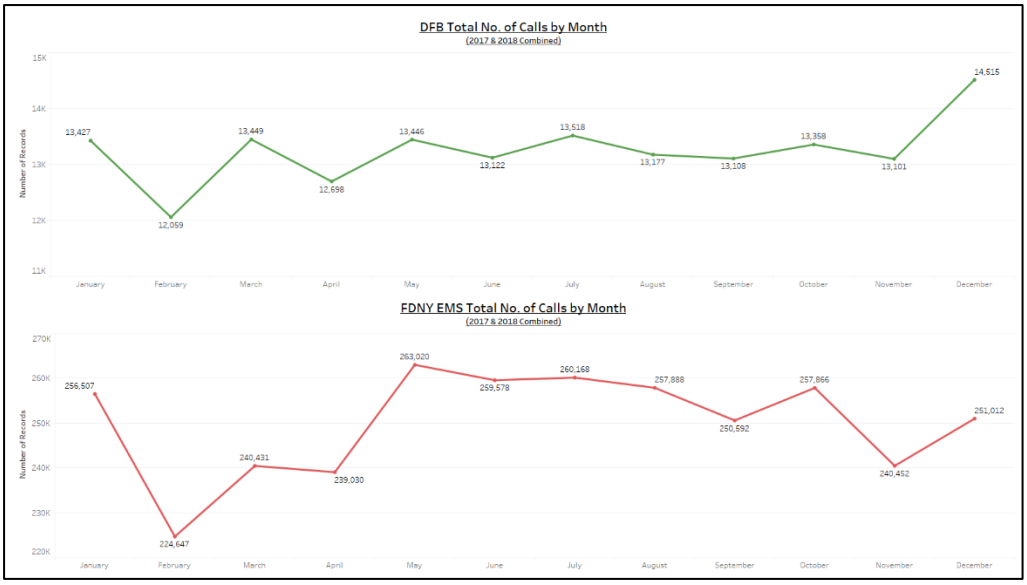


Figure 4.3 Total No. of Calls by Month Comparison

The FDNY EMS showed similar results with the variation for the first 6 months of the years, following a similar trend to DFB of increased results one month and declining on the next. The number of calls that were responded had a continual decline for the summer months, though were still some of the busiest, and in November, but rose again in October and December. We discovered that May was the busiest month for the FDNY EMS where the number of calls logged was 263,020. January was the 2nd busiest month of the years for the EMS, recording 256,507 calls, which was roughly 5% below the number of calls in May.

4.1.4. Number of Calls Per Week

Following on from the months that the calls are coming in, it was interesting to see how many calls were being made by the week. The data in figure 4.4 shows the number of calls that were made to both the DFB and the FDNY EMS for the years 2017 & 2018 combined to identify a pattern in the data.

For both cities, we found that there was a lot of fluctuation in the weeks, often following a trend similar to one seen in the month by month analysis (see figure 4.3), where it followed an increase in calls one week, and it started to decline again for the next.

The graph in figure 4.4 clearly indicated that week 51 and 52, which was the week prior to and of Christmas had the most calls recorded, were the busiest for the DFB, with 3,395 calls made in a week. Other high numbers of calls were found in week 7 (3,126 calls), approximately around the time that schools in Ireland are on mid-term holidays, and week 46 (3,152 calls). Weeks that were found to have significant drops in the number of calls was week 15 (2,856) & week 32 (2,872).

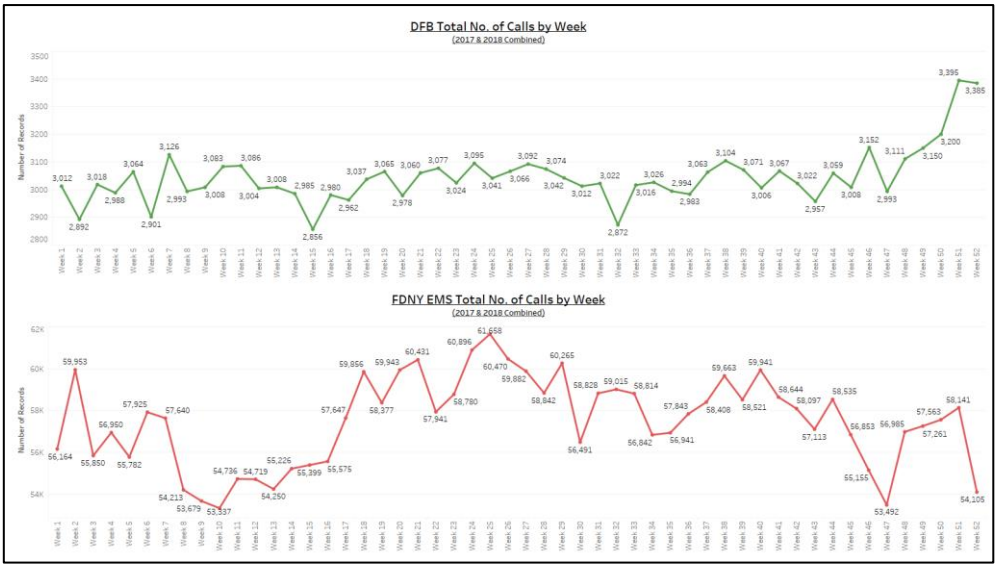


Figure 4.4 Total No. of Calls by Week Comparison

Unlike DFB, we discovered that the busiest week of the year for the FDNY EMS, with 61,658 calls logged was week 25, and the quietest period was week 10 with 53,337 calls. The difference in the number of calls in the busiest and quietest week was 6,811, and we found that the majority of the number of calls per week was between 55,000 & 60,000 calls for the two years combined, so roughly 27,500-30,000 calls per week each year.

4.1.5. High-Peak & Low-Peak Days & Times

Now that the number of calls that are being responded to are known for each year, quarter, month & week, it was then time to understand what days and hours of the day are the busiest for the ambulance services.

The YoY comparison seen in figure 4.4 for DFB shows that for 2017 & 2018 the weeks started off considerably high in the number of calls on Mondays, but by Tuesday there was a significant decline in the number for calls. From Tuesday onwards, there is a continuous growth in the number for calls responded to until in 2017 it hits its peak with 12,266 calls received on Sundays. Interestingly, in 2018 the graph showed a similar growth to 2017 but declines again on Sundays from 12,081 on Saturday's to 11,917 on Sunday's.

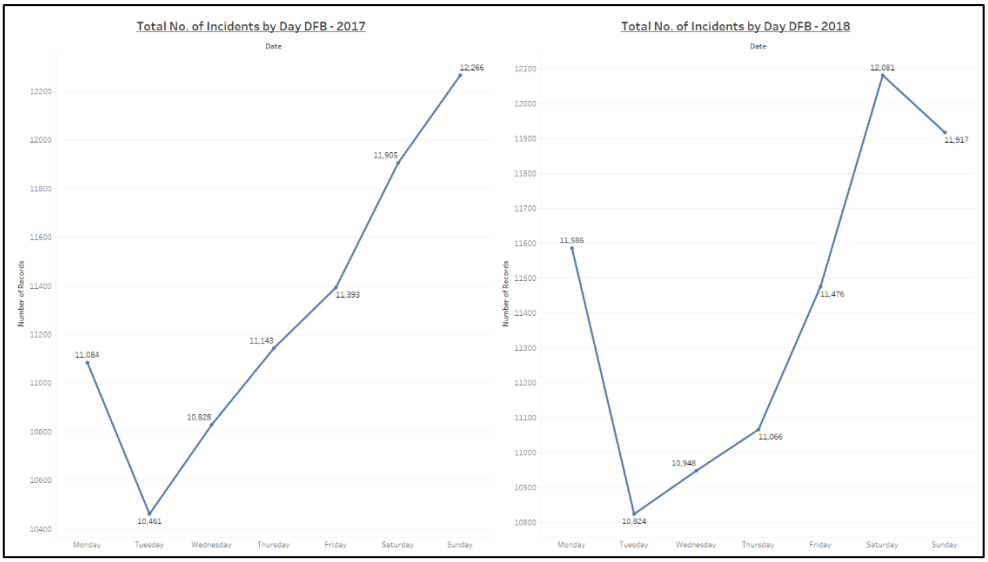


Figure 4.5 YoY No. of Calls by Day of the Week (DFB)

The results found for NYC in figure 4.6 were rather surprising and unexpected to say the least. Most of the calls to the FDNY EMS were received Monday-Friday. In 2017, the number of calls per day once again followed a pattern on increasing one day and declining the next up as far as Friday, whereas with 2018 the number of calls decreased from Monday-Wednesday and started climbing back up again towards Friday and experienced a sharp decline in number of calls for the weekend.

The unexpected result was when the weekend came along, where a significant drop in the number of calls was found. In 2017, there was a drop in the number of calls between Friday and Saturday of 4.91% with a further decrease of 0.81% on Sundays, and in 2018, the decline in calls for the same two days was 4.97% with an additional decline of 5.01% on Sundays. The days went from experiencing their busiest days of the weeks, to having the quietest days of the weeks on the weekend, however, 200,000+ calls to the EMS is far from quiet.

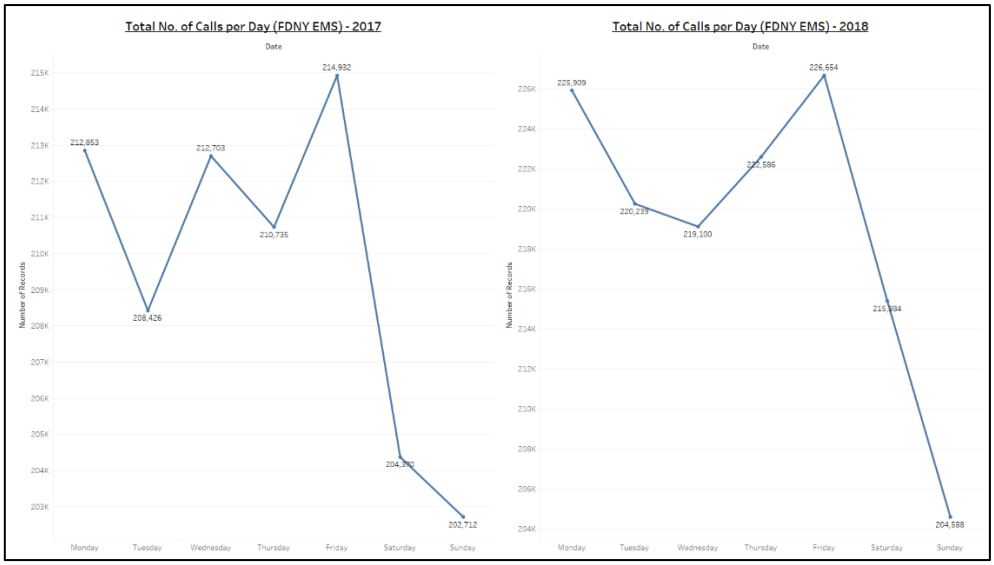


Figure 4.6 YoY No. of Calls by Day of the Week (FDNY EMS)

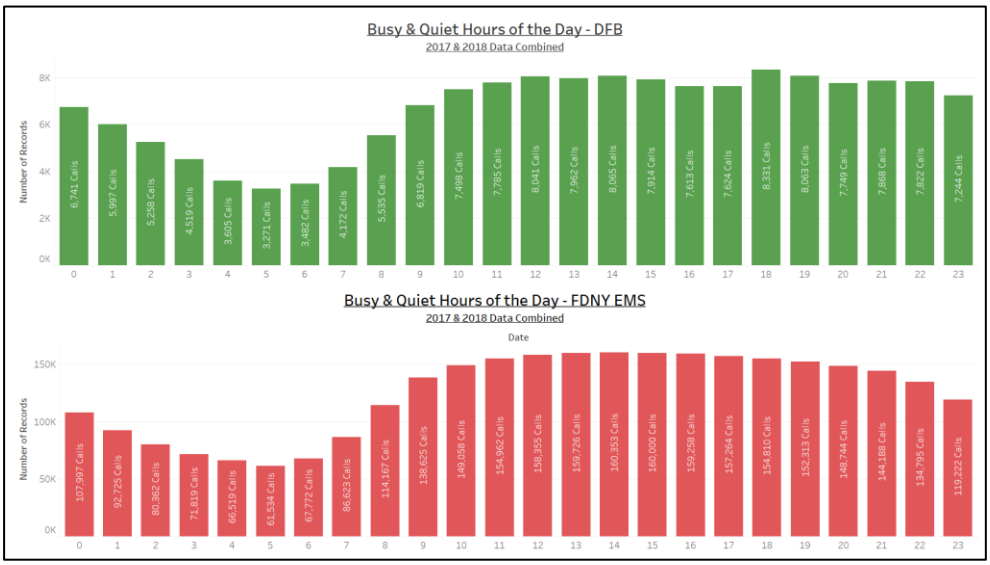


Figure 4.7 Comparison of the Number of Calls Attended to by DFB & FDNY EMS

It is important for the ambulance service to understand what times of the day are the busiest. The graph in figure 4.7 is showing the number of calls that are coming in at each hour of the day over the two years being analysed. Both services have shown a similar curve in the line as the day progresses.

DFB had some fluctuation in the number of calls towards the 2nd half of the days, while FDNY EMS maintains the shape of the curve. We found that the number of calls started to pick up from 7am, however, the busiest time for DFB was between 10am & 12pm before slowly declining and peaking again at 6pm. Similar to DFB, from 7am onwards, it is all go for the FDNY EMS as their busy time extended throughout the entire afternoon, until it starts to gradually wind down from 10pm.

Both services were found to have similar off-peak time, recording all their lowest calls between midnight and 7am.

4.1.6. About the Calls

While it is important to know how many and when the calls are coming in, it is also important to understand the types of calls that are being made too. The bar chart in figure 4.8 shows the number of calls that the DFB and the FDNY EMS are responding to year on year, broken down into the levels of severity for both cities.

Very clearly, the most common level of severity for the DFB for both 2017 & 2018 was Delta calls, the 2nd highest severity, which we also found in NYC. These calls are classified as life threatening, excluding conditions such as heart or respiratory attack. The highest level of severity calls that the DFB dealt with recorded the lowest number of calls, and the 2nd lowest number of calls for FDNY EMS. The remaining levels of severity between the medium-high and low severity all were found to have a very similar number of calls YoY.

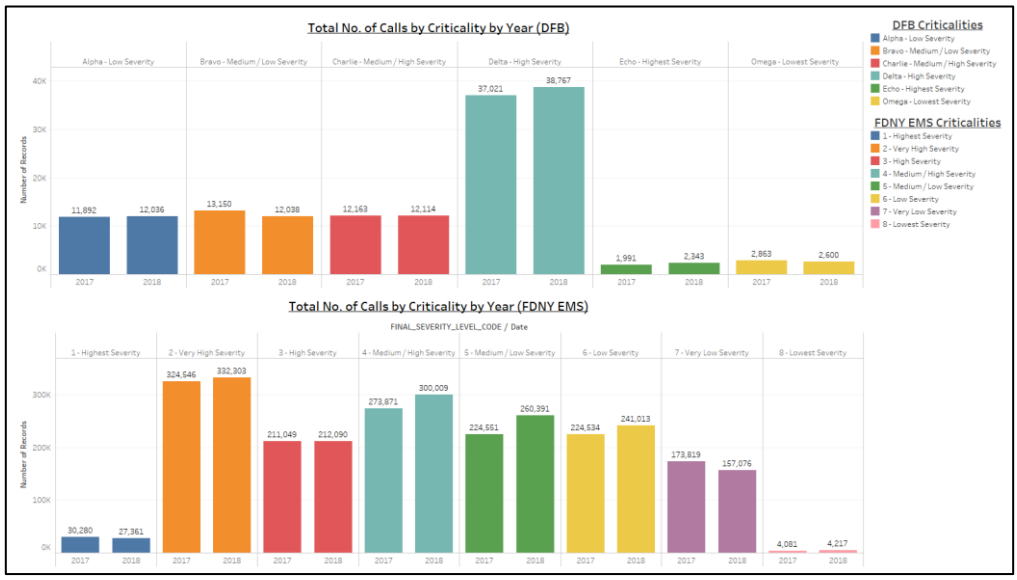


Figure 4.8 YoY No. of Calls by Severity Comparison

While it was not possible to understand the types of calls that are linked to each level of severity for the DFB, it was however for the FDNY EMS. Looking at figure 4.9 we could see the top 5 types of calls that the EMS are responding to for each of the 5 Boroughs. 4 of the 5 Boroughs all record the same 5 most common call types.

The first one was SICK, which is just a general sickness, and is the most common. Following that, there was INJURY, recognised a non-critical injury. The third most common was DIFFBR, understood to be a difficult breather, which was only in 4 of the 5 boroughs, not in Manhattan. Next

there was DRUG, classified as history of drug or alcohol abuse. The fifth call type was EDP, which is referred to as a psychiatric patient. Lastly, there was UNC, which only features in Manhattan, and means the patient is unconscious.

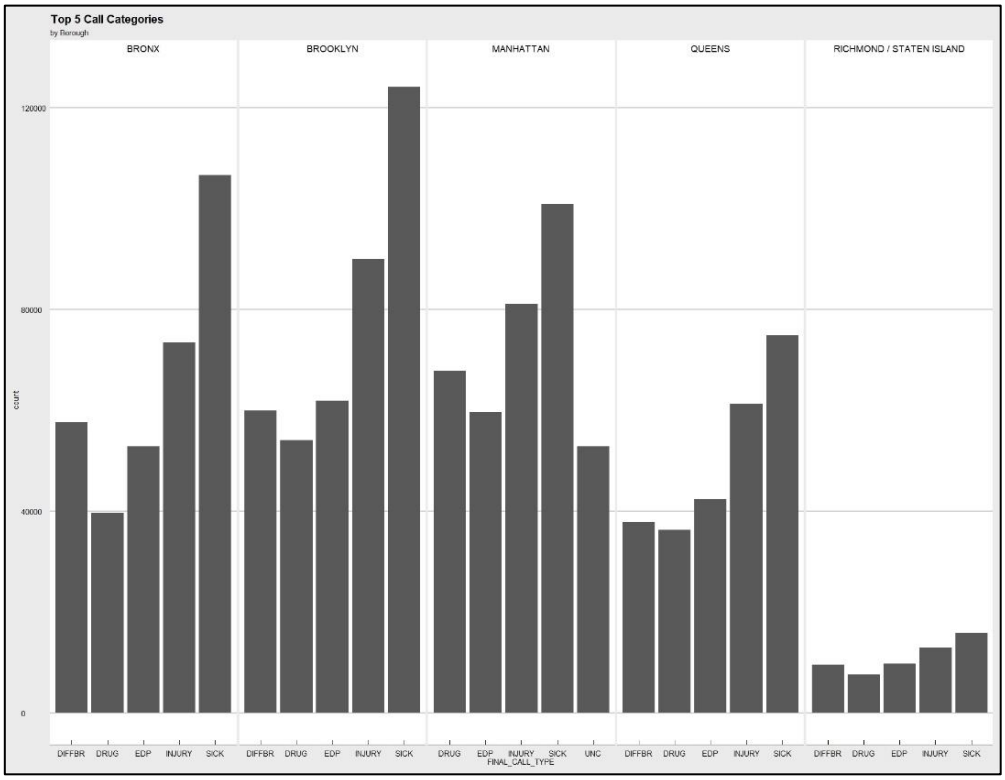


Figure 4.9 Top 5 Calls Types by Borough in NYC

Over the course of the two years, it was interesting to find that the Mater hospital was the most frequently visited hospital by DFB (see figure 4.10). We found that between 2017 & 2018, there was 40,150 visits by the ambulance services. The ambulances ended up at the Mater 25.31% of the time, with St. James's in 2nd (34,914 visits, 22.01% of total) and Beaumont in 3rd (30,483 visits, 19.22% of total). The least visited hospital, St. Vincent's, only had 9,745 visits, which was 7.46% visits less than the 2nd least frequently visited hospital.

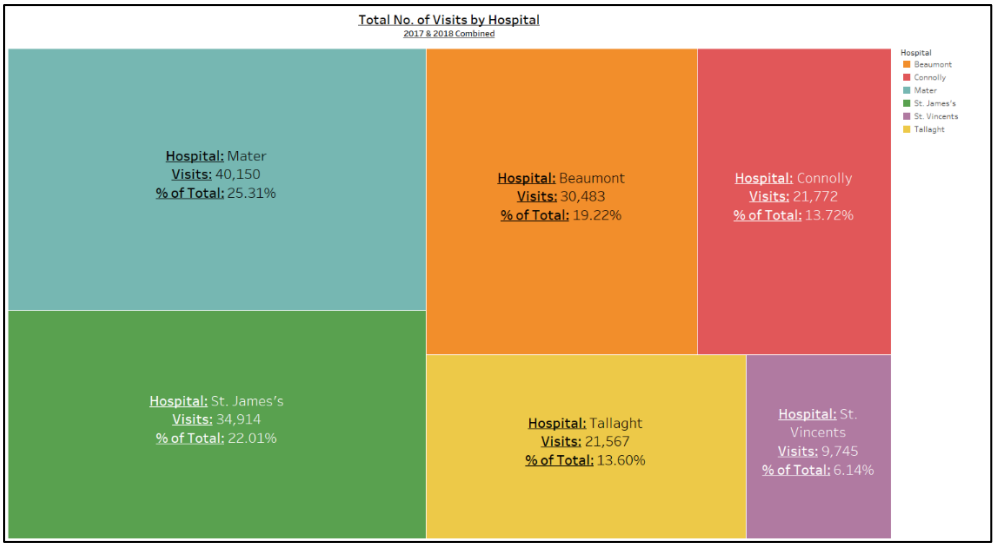
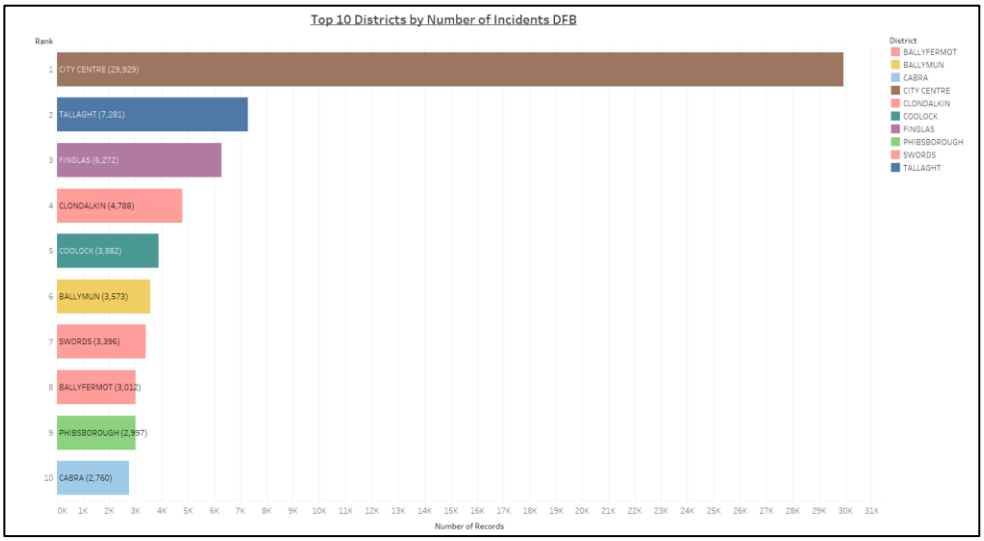


Figure 4.10 Total No. of Visits per Hospital (2017 & 2018 Combined)

To understand why both hospitals in the city centre are the busiest of the 6 that DFB have visited. A bar chart seen in figure 4.11 was used to visualize the top 10 districts that the DFB responded to calls from. It was very clear that link can be seen between the number of calls that the DFB responded to in the city centre, where there were almost 30,000 calls made in the city for the two years.



For further analysis to see a breakdown of how many calls were made each month, a ranked bar chart race was created to visualize the top 10 districts by how many calls were attended to for every month of the 2017 & 2018. We found that the City Centre had the most calls every month for both years, which helped understand why both City Centre hospitals had the most visits.

Tableau Public Ranked Racing Bar Chart of Top 10 Districts by Month: https://tabsoft.co/3fv5kHK

Alternatively, we looked at how many calls were responded to by the FDNY EMS for each of the boroughs in NYC. The borough with the most calls was very clearly Brooklyn, with 762,612 calls across the two years, with Manhattan in 2nd with 11.94% less calls responded to. These results we found were as expected considering Brooklyn and Manhattan are the boroughs with the largest population, while Staten Island had the least number of calls and the lowest population.

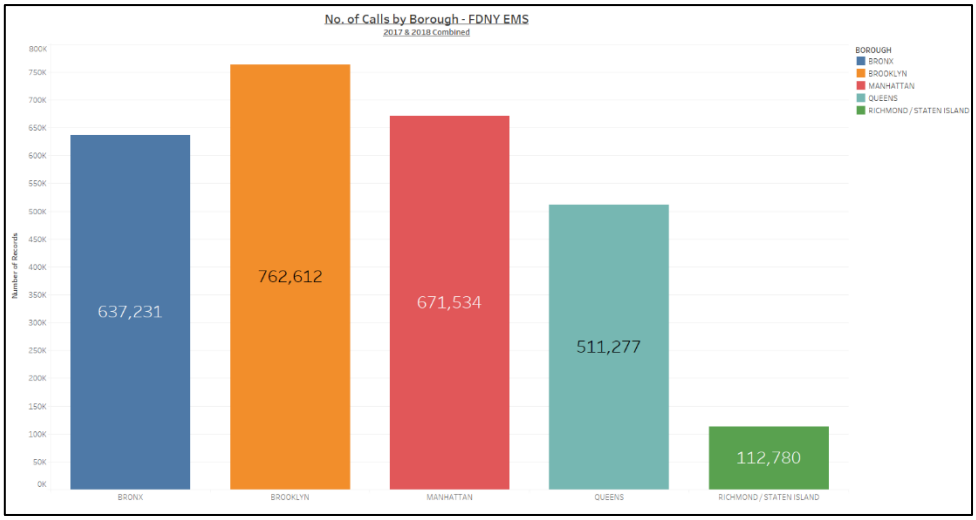


Figure 4.12 Total No. of Calls by NYC Borough - FDNY EMS (2017/18 Combined)

In NYC, the locations of the calls made to the FDNY EMS were mapped to visualize where the calls were coming from around the city. Figure 4.13 allowed us to identify where the clusters of calls are occurring in each of the 5 boroughs.

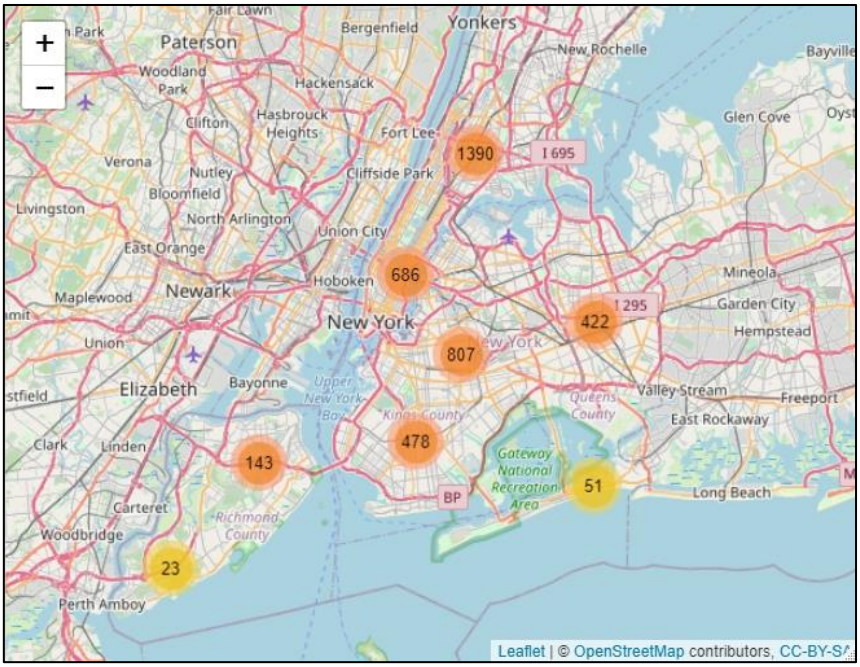


Figure 4.13 Cluster Map of a Sample of Calls made to FDNY EMS

Using a sample of 4000 calls of the two years, we found that the borough with the largest volume of calls was the Bronx, with a cluster of 1,390 calls. This was unexpected, considering the Bronx was the third in the rankings for most calls (see figure 4.12) although because it was just a sample of 4,000 out 3 million calls, it could have been any of the boroughs with the largest cluster.

Figure 4.14 Map Distribution of Calls Attended to by FDNY EMS

Finally, to see how where the calls were distributed, using the same sample, we mapped the calls using latitude and longitude (see section 2.5.2 to see how latitude and longitude were obtained). As other results have suggested, we found that the FDNY EMS did not have much work to do in Staten Island, and most of the activity was in the other 4 borough, and by selecting the individual pins, we could identify what the reason for all of the calls was. For further improvement on the maps that are produced, see section 4.4.

4.2. Ambulance Response Time Analysis

4.2.1. Average Response Times

The bar charts in figure 4.15 and figure 4.16 are visualizing the average length of time it took for an ambulance to arrive at the scene from the time the call was made for DFB and the FDNY EMS. The severity of calls is grouped into categories Omega, Alpha, Bravo, Charlie, Delta, Echo, with Omega being the lowest and Echo being the highest severity, and for the FDNY EMS, it is categorized at severity 1-8, with 1 being highest and 8 being lowest. So, assumptions were that it took less time to respond to a category Echo or severity 1 call than it does to a category Omega or severity 8 call.

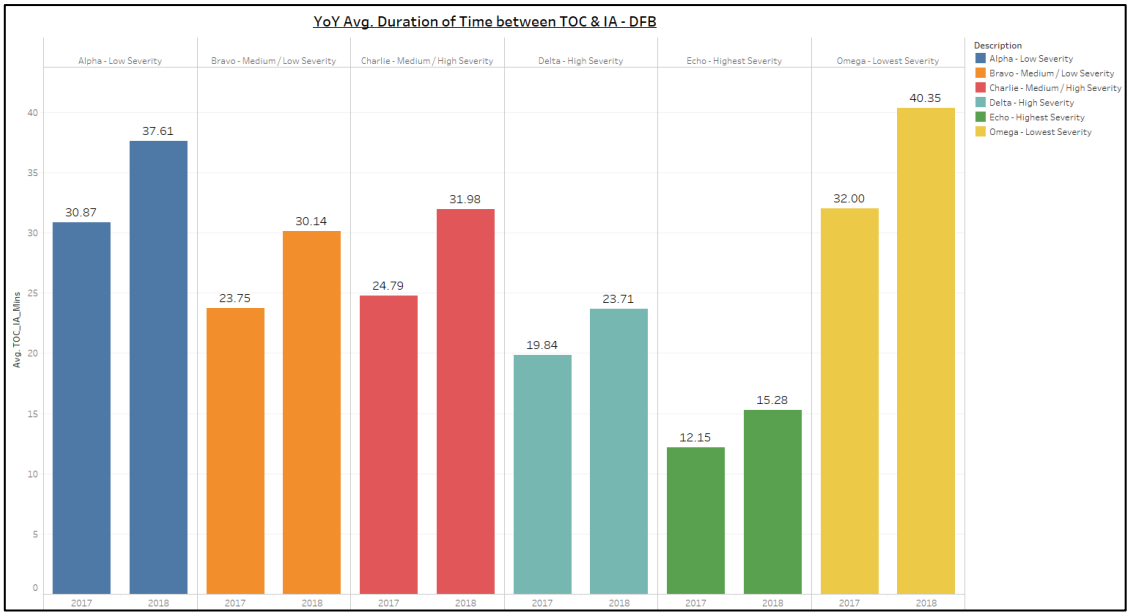


Figure 4.15 YoY Comparison of DFB Average Response Times by Severity

It was very clear that the length of time it took for the ambulance to arrive to the scene for category E calls was very quick, less than 15 minutes for each year. The YoY comparison indicates that there was an increase in the length of time it took for the ambulance to respond for all categories. The increase for each category was: Omega (26.09%), Alpha (21.83%), Bravo (26.91%), Charlie (29%), Delta (19.51%), Echo (25.76%), which is interesting because the number of calls for DFB did not drastically increase YoY from 2017 to 2018.

What stood out was that on average, it took slightly less time to respond to Bravo calls (Serious, Not Life Threatening, Urgent), finding an average response across the two years of approx. 26 minutes, than it did to respond to Charlie calls (Serious, Not Life Threatening, Immediate), which we found took approx. 27 minutes.

Comparing this to NYC, the average response times for the NYC indicated something more like what we expected to see in both services. The average length of time that it took to respond started off rapidly, and the response time gradually increased as the level of severity decreased. That was until there was an increase of roughly 344.45% in the average length of time it took to respond to the calls of the lowest level of severity. Like the DFB results, the average response time did increase in 2018, but not by as much, even though the percentage increase for the FDNY EMS was much higher than that of the DFB.

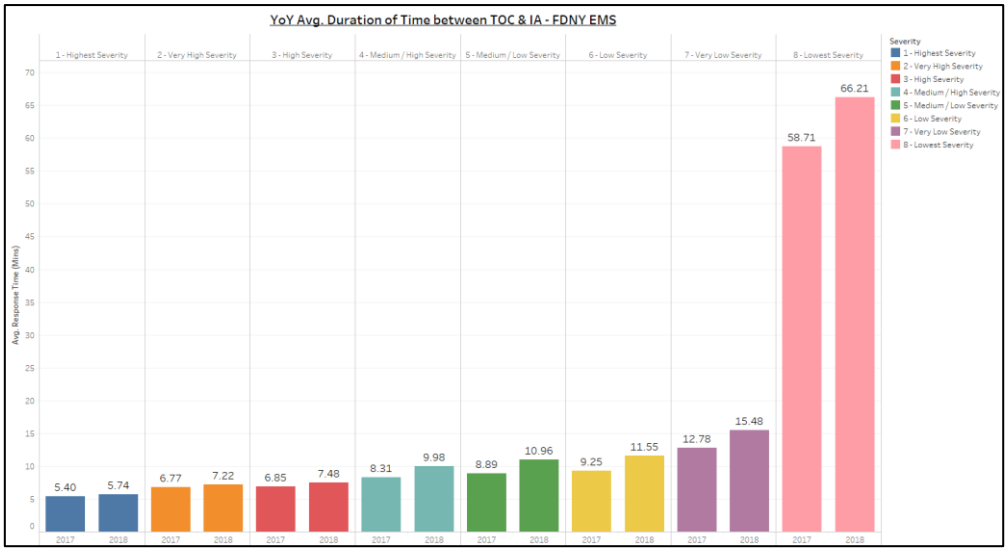


Figure 4.16 FDNY EMS Average Duration of Time between TOC & IA (Mins)

Although a difference is notable by looking at the YoY comparison bar plots, we wanted to test if there was a statistically significant difference in the response times between each of the categories of severity for the DFB. To do this we ran a statistical test.

4.2.2. Test for Data Normality

First, we checked whether the data was normal or not. This determined whether we could carry out a parametric or nonparametric test on the data. The test that we used was a Shapiro-Wilk test as we used a sample of the overall data which had less than 5000 records in each category. The declare hypothesis for the normality test was:

- H0: The data are normal
- H1: The data are not normal

The null hypothesis (H0) was declaring that the sample data that we conducted the test on was normally distributed, which would mean a parametric test could be performed on the on the data. On the other hand, the alternate hypothesis (H1) indicated that the sample data being used was not normally distributed and that would mean a non-parametric statistical test would have to be conducted. For the purpose of the test of normality, an alpha value was declared at $\alpha = 0.05$. This meant that we accepted there was a risk of 5% of committing a type 1 error, which would have been rejecting the H0 when in fact it should have been retained. As this was a test that just investigated the normality of data for the response times based on the severity of the calls, and nothing more, it was a significant enough risk to accept.

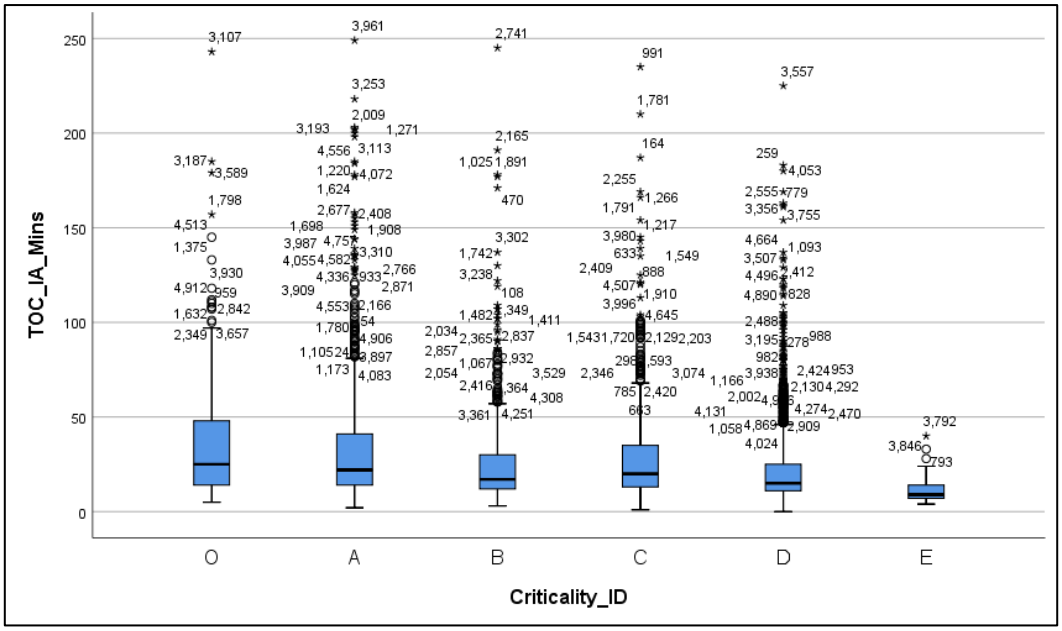


Figure 4.17 Boxplot of Response Times per Severity Category (DFB)

The boxplot in figure 4.17 was showing a strong indicator that the data was not going to be normal, showing that each of the categories was being heavily influenced by plenty of outliers. In regard to the descriptive statistics (see appendix 7.6), the skewness for each category was all > 2 and the kurtosis was > 5 for each category, representing a heavily right tailed skew in the data, which is a strong indication that the data was not going to be normal. However, we wanted to say statistically that it was not normal, so a Shapiro-Wilk test was conducted to determine whether the data that being used was normally distributed or not, which would determine the type of statistical test that could be conduct. The test results for each of the groups were:

Table 4.1 Results of Shapiro-Wilk Test on DFB Response Times per Severity (DFB)

Shapiro-Wilk Test of Normality Results		
Call Severity	**Test Statistic**	**P Value**
Omega	W = .761	p < 0.001
Alpha	W = .707	p < 0.001
Bravo	W = .665	p < 0.001
Charlie	W = .706	p < 0.001
Delta	W = .671	p < 0.001
Echo	W = .817	p < 0.001

As the results for each of the severities have a p value < 0.05, the declared alpha value, meaning H0 can be rejected in favour of H1. It would appear that the data was statistically proven to be not normal. This meant that we could run a non-parametric statistical test.

After discovering that the data was not normal, we the proceeded to test whether there was a significant difference in response times between each of the severities. To do this, we performed a non-parametric Kruskal-Wallis H.

4.2.3. Kruskal-Wallis H Non-Parametric Test

Criticality ID Breakdown [A – Alpha | B – Bravo | C – Charlie | D – Delta | E – Echo | O – Omega]

- H0: There is no difference in the distribution of average response times between the call severity categories
- H1: There is a difference in the distribution of average response times between at least two of the call severity categories
- $\alpha = 0.05$

The test was conducted on the 6 severity categories, Omega (n = 159, Sd = 37.216), Alpha (n = 786, Sd = 38.722), Bravo (n = 754, Sd = 30.606), Charlie (n = 825, Sd = 26.231), Delta (n = 2388, Sd = 20.492), Echo (n = 88, Sd = 9.499). The obtained test statistic, which was adjusted for ties in the data was H = 276.759, p < 0.001. As the p value < 0.05, it is clear that H0 can be rejected in favour of H1. It appeared that there was a difference between some of the groups, and to find out which groups, we used a pairwise comparison.

Table 4.2 Kruskal-Wallis H Results for Average Response Time per Criticality (DFB)

Independent-Samples Kruskal-Wallis Test Summary

Total N	5000
Test Statistic	276.759[a]
Degree Of Freedom	5
Asymptotic Sig.(2-sided test)	.000

a. The test statistic is adjusted for ties.

Table 4.3 Pairwise from Kruskal-Wallis Test for Average Response Time per Criticality (DFB)

Pairwise Comparisons of Criticality_ID

Sample 1-Sample 2	Test Statistic	Std. Error	Std. Test Statistic	Sig.	Adj. Sig.[a]
E-D	1164.895	159.233	7.316	.000	.000
E-B	1420.799	165.042	8.609	.000	.000
E-C	1678.101	164.278	10.215	.000	.000
E-A	1858.041	165.222	11.246	.000	.000
E-O	1944.311	187.802	10.353	.000	.000
D-B	255.904	60.187	4.252	.000	.000
D-C	513.206	58.060	8.839	.000	.000
D-A	693.147	60.679	11.423	.000	.000
D-O	779.417	107.950	7.220	.000	.000
B-C	-257.302	72.489	-3.550	.000	.006
B-A	437.243	74.603	5.861	.000	.000
B-O	523.513	116.348	4.500	.000	.000
C-A	179.941	72.898	2.468	.014	.204
C-O	266.211	115.263	2.310	.021	.314
A-O	86.270	116.604	.740	.459	1.000

Each row tests the null hypothesis that the Sample 1 and Sample 2 distributions are the same.

Asymptotic significances (2-sided tests) are displayed. The significance level is .05.

a. Significance values have been adjusted by the Bonferroni correction for multiple tests.

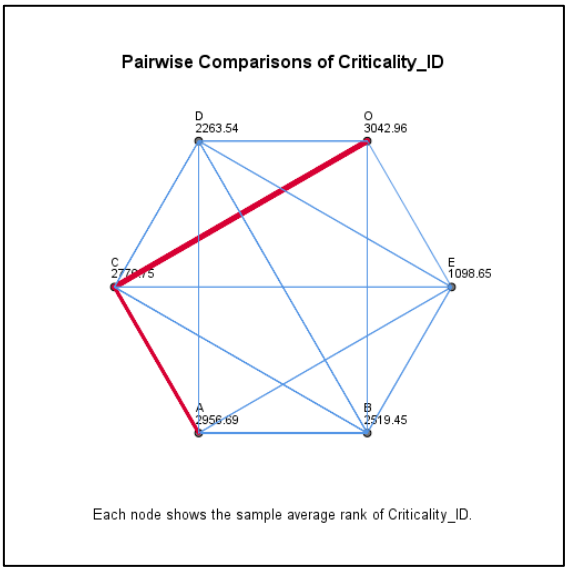


Figure 4.18 Pairwise Comparison of Average Response Time per Criticality (DFB)

We can see from looking at table 4.3 & figure 4.18 where the differences are found. For example, we can see there was a significant difference in the average response times between Delta

& Echo calls. However, a difference was not found between Charlie & Alpha / Omega calls. what makes this interesting is Charlie calls are considered serious but non-life threatening requiring immediate attention, while Alpha and Omega calls are not serious or life threatening.

4.2.4. Handover Time in Dublin Hospitals

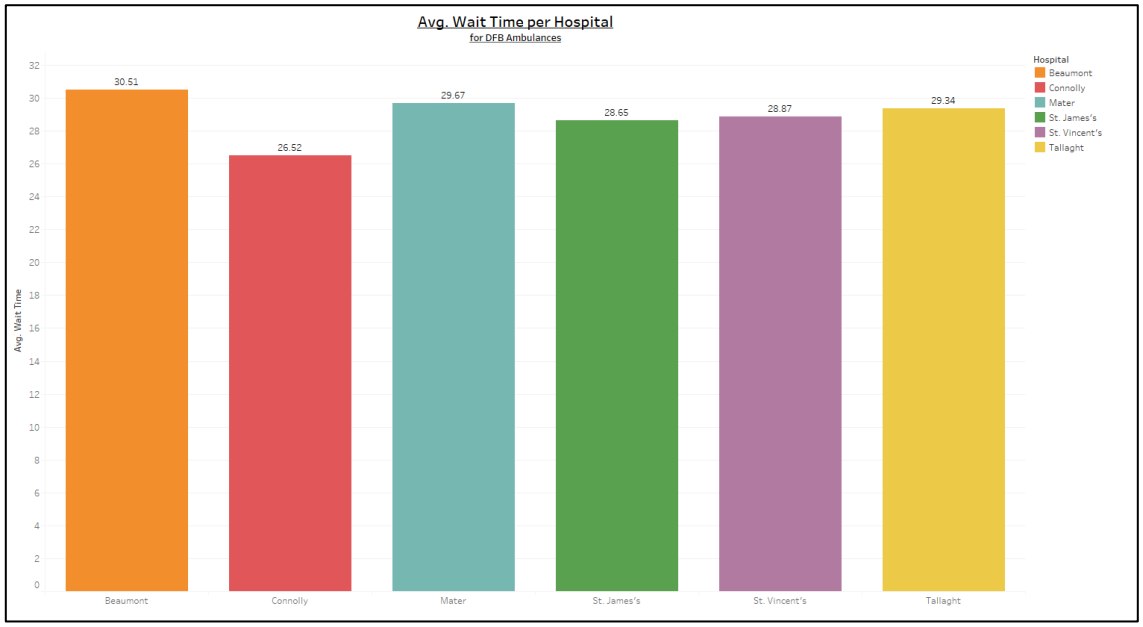


Figure 4.19 DFB Average Wait Time & No. of Visits per Hospital 17/18

To understand potential factors that can lead to delays for a quick ambulance response, we investigated the handover time in each of the hospitals that the DFB visited. In general, the average handover time for all the hospitals is not quick at all. All hospitals were found to have an average turnaround time of 30 minutes, where discovered that Beaumont had the longest handover time, with an average of almost 31 minutes between the ambulance crew arriving at the hospital and being available to respond to their next call, and Connolly having the shortest turnaround of roughly 27 minutes.

4.3. Machine Learning Analysis

4.3.1. Random Forest Results

The overall performance of the random forest model seen in table 4.4 was good, but far from excellent. The accuracy of the model was 97.42%, which is great, and a 95% CI [97.31, 97.54] that the accuracy will lie between these two numbers. On the other hand, kappa, which measures the reliability of the model & sensitivity, which looks at the number in the actual positive class (Y) that were predicted accurately, did not have such great results. Our kappa value of 64.86% meant that there was a still a 35.14% chance of guessing the majority class, in this case was N (No), and it

being the right answer. The specificity however, which looks at the actual number in the negative positive class (N) and how accurately it predicted them, in this case, 99.67% accurate.

Table 4.4 Random Forest Prediction Confusion Matrix Evaluation

Confusion Matrix

	Reference	
Prediction	**N**	**Y**
N	71184	1694
Y	238	1884

Accuracy	0.9742
95% CI	0.9731, 0.9754
Kappa	0.6486

Sensitivity	0.52655
Specificity	0.99667

After evaluating the performance of the model, we looked for anything that stood out that could have increased the kappa and sensitivity value. A large class imbalance was identified, where out of the 225,000 calls in the training data set, 214,297 were the class N.

4.3.2. Random Forest with Random Over Sampling Examples (ROSE)

In an attempt the deal with the class imbalance in the random forest model, the over sampling technique, R.O.S.E., was introduced to balance the training data classes of the dependent variable. This provided a data frame that would be used to train the model, with 428,594 rows now added with an even number of held up classes.

When evaluating the model, we found that using a balanced class, it resulted in weakening the model's performance, with the accuracy, kappa and specificity all decreasing in score, though only by a small amount. The only value, sensitivity, which performed poorly in the normal random forest model, had increased by approx. 22% overall. This resulted in 74.06% of the actual positive class, Y, being accurately predicted.

Table 4.5 Random Forest with R.O.S.E. Prediction Confusion Matrix Evaluation

Confusion Matrix

	Reference	
Prediction	**N**	**Y**
N	69743	918
Y	1907	2623

Accuracy	0.9624
95% CI	0.961, 0.9638
Kappa	0.6304

Sensitivity	0.74075
Specificity	0.97338

Overall, this model also performed good, but was a little worse than the random forest with the class imbalance.

4.3.3. Naïve Bayes

Naïve Bayes was the 2nd of the 3 machine learning algorithms performed to determine if we could predict whether an ambulance was held up in being assigned to a call or not using. It was clear that the results for this model were far worse off than the random forest model was.

Table 4.6 Naive Bayes Prediction Confusion Matrix Evaluation

Confusion Matrix

	Reference	
Prediction	N	Y
N	66432	1585
Y	5012	1971

Accuracy	0.912
95% CI	0.91, 0.9141
Kappa	0.3321

Sensitivity	0.55427
Specificity	0.92985

For start, the accuracy of the model was only 91.2%, with a 95% CI [91, 91.41]. The very low kappa for the Naïve Bayes model indicated that just randomly guessing the value by using the most common class had an approx. 67% chance of being correct, making the model very unreliable. Yet

again, the model only accurately predicted the positive class 55% of the time, and 93% of the time accurately predicted the negative class.

While the accuracy was still considerably high for the Naïve Bayes model, the low kappa made in an unreliable model to use and for that reason has come out as the worst model so far to use when predicting the dependent variable.

4.3.4. SVM – Linear Grid Tuned

The first of the three types of SVM models being ran is the linear grid tuned. The results in this model lined up closer to the random forest model, than it did to the Naïve Bayes. The accuracy of and kappa results are indicators that once again, we found a moderately good model for predictions. The accuracy of the model was 97.74%, with a 95% CI [97.29, 98.13], and found the kappa was 68.94%, indicating is less likely that we can just guess the predicted value by choosing the most frequent class. Our sensitivity was low again, with only 55% of the positive class predicted correct, and a specificity of 99.89% of accurately predicting the actual negative class.

Table 4.7 SVM Linear Grid Tuned Confusion Matrix Evaluation

Confusion Matrix

	Reference	
Prediction	N	Y
N	4755	108
Y	5	60

Accuracy	0.9774
95% CI	0.9729,0.9813
Kappa	0.6894

Sensitivity	0.5500
Specificity	0.9989

4.3.5. SVM – Radial Grid Tuned

We discovered that the radial grid tuned SVM moved was one of our best models so far, where we discovered the accuracy of 97.96%, 95% CI [97.53, 98.33]. The kappa value was found to be 73.1%, the highest so far and the first one of the models to exceed 70%.

Table 4.8 SVM Radial Grid Tuned Confusion Matrix Evaluation

Confusion Matrix		
	Reference	
Prediction	N	Y
N	4752	94
Y	8	146

Accuracy	0.9796
95% CI	0.9753, 0.9833
Kappa	0.731

Sensitivity	0.6083
Specificity	0.9983

The sensitivity for this model was the highest so far in all models that class imbalance in the data was not dealt with. The model accurately predicted the 60.83% of the actual positive class, and specificity was just as high as other models, accurately predicting the negative class 99.83% of the time.

4.3.6. SVM – Radial Random Tuned

The final machine learning model that was performed on the data to try and predict whether the ambulance would be held up was the Radial Random Tuned SVM model. Like the previous model, this was one of the best models that we found, and only the 2nd model with a kappa value above 70%. The accuracy of the model had a slight decline on the previous model, only being 97.86% accurate, 95% CI [97.42, 98.24], and found a kappa of 71.02%.

Table 4.9 SVM Radial Random Tuned Confusion Matrix Evaluation

Confusion Matrix		
	Reference	
Prediction	N	Y
N	2372	60
Y	5	62

Accuracy	0.9786
95% CI	0.9742, 0.9824
Kappa	0.7102

Sensitivity	0.5750
Specificity	0.9989

The sensitivity was another one of the best, though still very far from great, was only 57.5% accurate at predicting the actual positive class of Y for the dependent variable, while the specificity as always remained at a very high 99.89% accurate at predicting the actual negative class of N.

4.4. Interactive Dashboard

R Shiny was used to visualize the location of a sample of 2000 calls made to the FDNY EMS between 2017 & 2018. The map was created using the leaflet package and published online. There was two pages created, the first was a map, and a table that listed all of the calls that were on the map, and the second had the descriptions of the final call type and the disposition code of the calls, which was the outcome of the call (*E.g. Code 82 – Transporting Patient, Code 92 – Treated, Not Transported*).

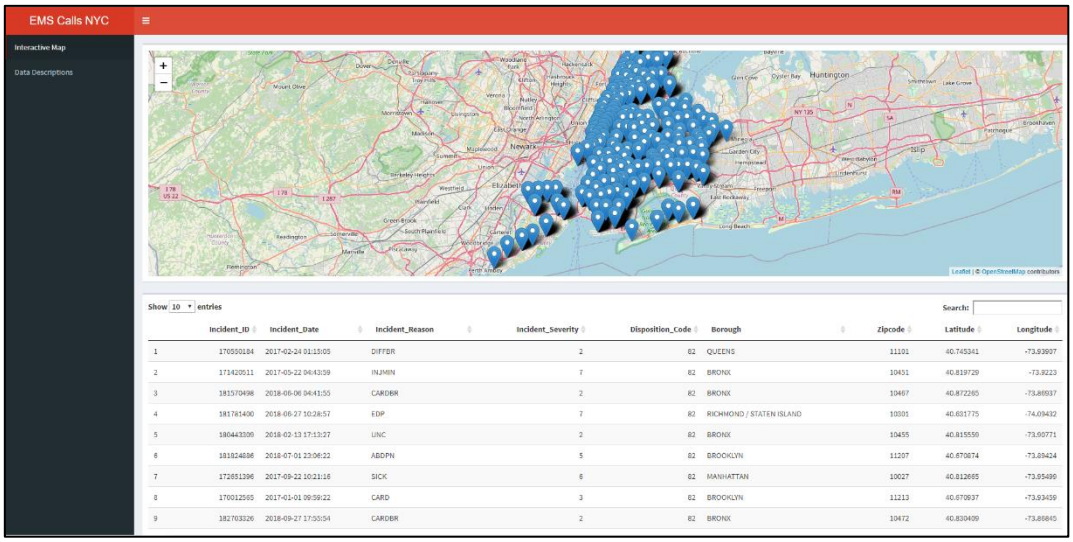


Figure 4.20 Main Page of R Shiny Dashboard

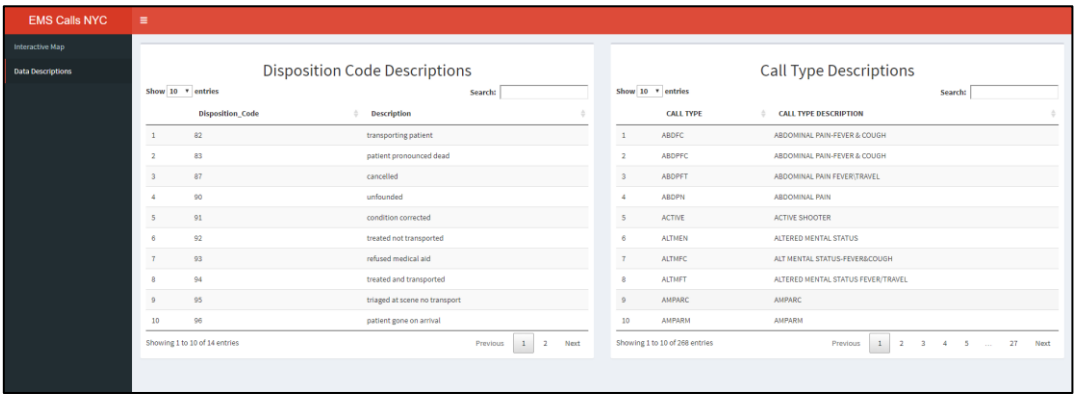


Figure 4.21 Data Description of Call Type & Disposition Code

The pins that were on the map could be selected and the incident ID would display, where it could be entered in the search bar on table below, which would filter to the specific call and display the information about the call. To understand what the reason for the call was or the outcome was,

the call type or the disposition code can be entered in the search bar for the relative section and can get an understanding of what exactly the code represents.

The dashboard was published to shinyapps.io through R Studio, where it can be accessed online by anyone with the link. The interactive dashboard can be found here: https://bit.ly/2LeFqd7

5. Discussion & Further Development

5.1. Discussion

After successful completion of the analysis on the data from the DFB and FDNY EMS ambulance services, there were some indicators we found that may be leading to the delays in response from the services. The sheer factor that in Dublin, there was almost 80 thousand calls a year, and in NYC there was between 1.45 million and 1.55 million calls for both years, alone is enough to indicate that there will be a delay for an ambulance to arrive at the scene of a call. The forecast for the number of calls in 2019 (see figure 4.1) showed no sign of getting any easier for either service any time soon.

Over the two-year period, out of the almost 159 thousand calls, the 12 ambulances for DFB clocked up over 13 thousand calls each. In comparison, the FDNY EMS responded to just over 3 million calls, and the 450 ambulances that they target to have on the streets every day (NYC.gov, 2020), each one only responded to under 7 thousand calls in the same two-year period. Using the 12 of 16 ambulances available meant they were working at just 75% of their capacity. Increasing the number of ambulances used going forward to 14 of the 16 (or 87.5% capacity) would reduce the number of calls each ambulance responds to, to approx. 11.5 thousand calls each, a reduction of over 10%.

Focusing on these years, the quarterly comparisons did not offer much insights as there was not much change YoY. While each month remained quite similar in terms of number of calls for the FDNY EMS, it changed for each month for the DFB. Looking at the busy and quiet days and hours that each service responded to, it's clear that as the week started off for the DFB, it was relatively quiet, recording their least number of calls on a Tuesday, and progressively increased as the week continued. As for the FDNY EMS, their week fluctuated in busy days from Monday to Friday, before unexpectedly witnessing a sudden decrease to the quietest days of the week over the weekends. The same trend was identified for both 2017 and 2018 and indicated that if the DFB were to allocate more resources for responding to the calls, it should be towards the end of the week that they do so, and for the FDNY, it should be midweek.

Both services had very similar trend in the number of calls they each responded to every hour, and like the days, were they to increase the number of resources allocated, then it should be

between 10am and 10pm when the calls are at their highest. As the HSE declares that their KPI for responding the two most critical calls (Delta & Echo) was 80% within 19 minutes, we found that the average was almost 20 minutes in 2017, but had increased to almost 24 minutes in 2018 for Delta calls, and for Echo call, the average response time was always within KPI. Figure 5.1 shows the distribution of calls that meet & exceed the KPI of 19 minutes for both levels of severity. We found that only 64.4% of the calls were meeting the KPI, which was 15.4% below the reported target.

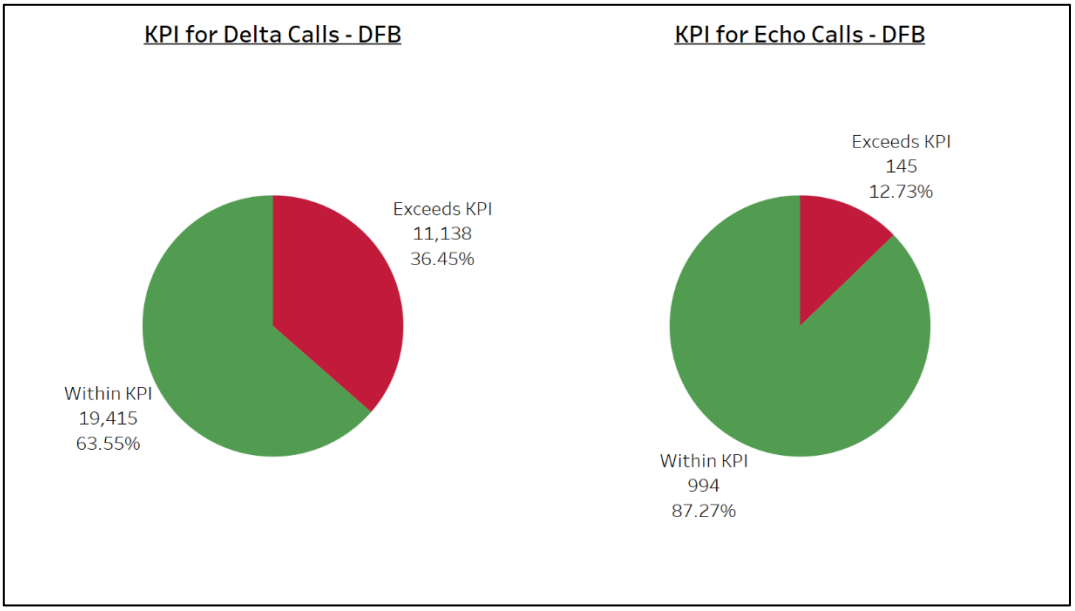


Figure 5.1 Years Combined KPI Standards for Delta & Echo Calls - DFB

Response times were found to be far from perfect for the DFB. The results indicate that the length of time it took to respond only gradually decreased for each level of severity from lowest to highest, with the exception of Bravo & Charlie calls, where the average response time increase from Bravo calls, when theoretically, it should have continued to decline. The pattern expected was seen in the FDNY EMS response times, where the average length of time it took to respond gradually increased from highest to lowest severity, and done their best to keep all calls within 20 minutes of time to response, though it was not always the case, especially in the lowest level of severity where the average response time was found to be approx. 1 hour.

Additionally, it was found that the average length of time that the DFB crews are left waiting at each of the 6 hospitals is rather problematic. Further analysis revealed the average length of time it takes for the hospitals to accept the patients before the ambulance crew can leave the hospital and prepare for the next call was approx. 30 minutes. This was a concerning discovery, considering most calls that were made to the DFB were Delta calls, which are life threatening, and if all the ambulances were stuck waiting for the hospital handover, the risk is then getting to the next patient on time. The Kruskal-Wallis H test that was performed was to show how there is a difference in the

response times between all levels of severity, however, we found that there was no statistically significant difference in the response time between Charlie calls, which are medium / high severity & Alpha / Omega calls which are both the lowest levels of severity.

Machine learning was introduced in the hope that we could predict whether the ambulance was going to be held up, meaning a call was not able to be assigned an ambulance immediately. Having ran the 3 different models, we identified that the best performing model was the Radial Grid Tuned SVM model, which recorded the best overall accuracy, kappa, sensitivity & specificity (see appendix 7.8 for all model performances). Dealing with the class imbalance using R.O.S.E with Random Forest was found to not strengthen the model and using probabilities with Naïve Bayes indicated that we had a better chance and random guessing using the majority class and being correct. None of the models performed exceptionally well when it came to predict whether the ambulance would be held up, so additional information or a different model would have to be considered.

Conclusively, the ambulance service is a front-line pre-hospital service attempting to keep as many people as safe as possible. In order for the DFB to have a chance in reducing their response times, they should be looking into increasing their capacity by utilizing some of the ambulances that are not being used, hospitals & the government should be establish methods to reduce the length of time the ambulance crews have to wait before being available for the next call. With the number of calls continuously on the rise with the ever-increasing populations, the relative parties should be looking to implement these changes sooner rather than later.

5.2. Further Development

Having completed this analysis, there is now a clearer vision of where this could be taken next. As previously mentioned, the data that was provided by the DFB was limited to just time and call severity analysis, when originally sought for much more information. It would be interesting to collaborate with the HSE, DFB or other ambulance service to analyse the calls that are being made every day.

We found that none of the machine learning models performed exceptionally, so further along the line, it would be interesting to implement additional machine learning models, or attempt predictions on other dependent variables, especially if there was more information available. It would be interesting to also obtain the chat transcripts of the calls being made to the ambulance service to perform word sentiment analysis and combine that with machine learning to try and identify any patterns and predict the type of call and how severe the call is.

5.3. Project Limitations & Challenges

Given the sensitivity of some of the data being used for this analysis, limitations and challenges in some sense were inevitable. The first issue that arose was when the data was received from the DFB, it was very limited in comparison to what was requested. No personal information regarding the call was provided, such as gender, reason for the call or age bracket etc., so for that reason, the type of analysis that could be done was limited. Mapping was also an issue with the DFB data, as the coordinates we were provided with were the Ordinance Survey Ireland (OSI) grid references, which we could not easily convert the large volume of these manually, and resulted in not being able to produce a map of the calls. A python script was developed to automate this, though it would have potentially breached some privacy policies, so this idea was disbanded.

The same email that was sent to the DFB, was sent to the HSE requesting the same information for the NAS. After several follow up phone calls, and several months of waiting, the data was not received, so this meant all calls made in Dublin could be analysed, which was the second challenge we encountered.

Moving onto the NYC data, the large volume of over 16 million calls made to the FDNY EMS meant that the data took in excess of 45 minutes to load the data into R Studio. After we began the analysis on the data, there was some analysis that could be done, but there was not much that could be analysed in regard to comparisons, other than the length of time that it took for the ambulance to respond to the scene from the time of call.

Hardware was one of the biggest issues to this analysis. Originally, everything was being analysed and stored on a laptop. Two months prior to completion, the laptop ceased functioning and anything that had been completed and was lost. A replacement part was found for the laptop, though when it turned back on, all the data stored on the drive was lost, so it was back to square one. In the meantime, while waiting for the laptop part to arrive, work restarted on a desktop, though hardware was also an issue, as there was only 8GB of memory which resulted in all processes taking longer than it should have. Additional memory was ordered and installed into the machine, which reduced processing time significantly. Another step that was taken to avoid data loss again was to utilize OneDrive cloud storage & GitHub more.

6. References

Blomberg, S. N. et al., 2019. Machine learning as a supportive tool to recognize cardiac arrest in emergency calls. *Resuscitation,* Volume 138, pp. 322-329.

Central Statistics Office, 2016. *EY004: Population and Actual and Percentage Change 2006 to 2016 by Sex, County and City, Census Year and Statistic.* [Online]
Available at:
https://statbank.cso.ie/px/pxeirestat/Statire/SelectVarVal/Define.asp?maintable=EY004&PLanguage=0
[Accessed 26 April 2020].

Cone, D. C., Middleton, P. M. & Pour, S. M., 2012. Analysis and impact of delays in ambulance to emergency department handovers. *Emergency Medicine Australasia,* 24(5), pp. 525-533.

Courtemanche, C., Friedson, A. I. & Rees, D. I., 2019. Association of Ambulance Use in New York City With the Implementation of the Patient Protection and Affordable Care Act. *JAMA Network Open,* 2(6), p. e196419.

Dejean, D. et al., 2016. Inappropriate Ambulance Use: A Qualitative Study of Paramedics' Views. *Healthcare Policy | Politiques de Santé,* 11(3), pp. 67-79.

Dolney, T. J. & Sheridan, S. C., 2006. The relationship between extreme heat and ambulance response calls for the city of Toronto, Ontario, Canada. *Environmental Research,* 101(1), pp. 94-103.

Dublin City Council, 2016. *Fire Brigade and Ambulance Call Outs.* [Online]
Available at: https://data.gov.ie/dataset/fire-brigade-and-ambulance
[Accessed 27 September 2019].

Dublin Fire Brigade, Personal Correspondence [2019]. *DA 2017 and 2018.* Dublin: s.n.

Fayaad, U., Piatetsky-Shapiro, G. & Smyth, P., 1996. The KDD Process fo Extracing Useful Knowledge from Volumes of Data. *Communications of the ACM,* 39(11), pp. 27-34.

FDNY, 2015. *FDNY Strategic Plan 2015-2017,* New York: FDNY.

Gandhi, R., 2018. *Naive Bayes Classifier.* [Online]
Available at: https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c
[Accessed 26 April 2020].

Gandhi, R., 2018. *Support Vector Machine — Introduction to Machine Learning Algorithms.* [Online]
Available at: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47
[Accessed 28 April 2020].

Griffin, R. & McGwin, G., 2013. Emergency medical service providers' experiences with traffic congestion. *The Journal of Emergency Medicine,* 44(2), pp. 398-405.

HSE, 2016. *National Ambulance Service Strategic Plan 2016-2020,* Dublin: HSE.

HSE, 2017. *National Ambulance Service Operational Plan 2017,* Dublin: HSE.

HSE, 2019. *Performance Profile July -September 2019 Quarterly Report,* Dublin: HSE.

Lightfoot Solutions, 2015. *National Ambulance Service of Ireland Emergency Service Baseline and Capacity Review,* Berkshire: Lightfoot Solutions.

Mills, B. et al., 2019. What constitutes an emergency ambulance call?. *Australasian Journal of Paramedicine,* Volume 16.

NYC OpenData, 2020. *EMS Incident Dispatch Data.* [Online]
Available at: https://data.cityofnewyork.us/Public-Safety/EMS-Incident-Dispatch-Data/76xm-jjuj
[Accessed 26 February 2020].

NYC.gov, 2010. *Population: Census Information and Data (Total Populations 1990-2010).* [Online]
Available at: https://www1.nyc.gov/assets/planning/download/pdf/data-maps/nyc-population/census2010/pgrhc.pdf
[Accessed 26 April 2020].

NYC.gov, 2020. *Fleet Report.* [Online]
Available at: https://www1.nyc.gov/assets/operations/downloads/pdf/fleet_report.pdf
[Accessed 26 April 2020].

Palazzo, F. F., Warner, O. J., Harron, M. & Sadana, A., 1998. Misuse of the London ambulance service: How much and why?. *Emergency Medicine Journal,* 15(6), pp. 368-370.

Panahi, S. & Delavar, M. R., 2009. Dynamic Shortest Path in Ambulance Routing Based on GIS. *International Journal of Geoinformatics,* 5(1), pp. 19-19.

Payne, D., 2000. Poor ambulance response causes 700 deaths annually in Ireland. *British Medical Journal,* 321(7270).

Schott, M., 2019. *Random Forest Algorithm for Machine Learning.* [Online]
Available at: https://medium.com/capital-one-tech/random-forest-algorithm-for-machine-learning-c4b2c8cc9feb
[Accessed 26 April 2020].

Schull, M. J., Morrison, L. J., Vermeulen, M. & Redelmeier, D. A., 2003. Emergency department overcrowding and ambulance transport delays for patients with chest pain. *Canadian Medical Association Journal,* 168(3), pp. 277-283.

Spangler, D., Hermansson, T., Smekal, D. & Blomberg, H., 2019. A validation of machine learning-based risk scores in the prehospital setting. *PLOS ONE,* 14(12), p. e0226518.

Trzeciak, S. & Rivers, E. P., 2006. Emergency department overcrowding in the United States: an emerging threat to patient safety and public health. *Emergency Medicine Journal,* 20(5), pp. 402-405.

Univeristy of Regina, n.d. *Overview of the KDD Process.* [Online]
Available at: http://www2.cs.uregina.ca/~dbd/cs831/notes/kdd/1_kdd.html
[Accessed 6 April 2020].

Young, V., Rochon, E. & Mihailidis, A., 2016. Exploratory analysis of real personal emergency response call conversations: considerations for personal emergency response spoken dialogue systems. *Journal of NeuroEngineering and Rehabilitation,* 13(1).

Zhan, Z.-Y.et al., 2020. Effects of hourly precipitation and temperature on ambulance response time. *Environmental Research,* Volume 181, p. 108946.

7. Appendix

7.1. Acronyms, Definitions & Abbreviations

Abbreviation / Acronym	Definition
SVM	Support Vector Machine
DFB	Dublin Fire Brigade
NAS	National Ambulance Service (Ireland)
HSE	Health Service Executive
FDNY	Fire Department of New York
EMS	Emergency Medical Service
KDD	Knowledge Discovery in Databases
YoY	Year over Year
YTD	Year to Date
KPI	Key Performance Indicator
Omega	Minor Illness or Injury
Alpha	Non-Serious or Non-Life-Threatening Injury
Bravo	Serious but Non-Life-Threatening (Urgent)
Charlie	Serious but Non-Life-Threatening (Immediate)
Delta	Life-Threatening Illness or Injury (Excluding Cardiac or Respiratory Arrest)
Echo	Life-Threatening Cardiac or Respiratory Arrest

7.2. Full Data Description FDNY EMS

Column Name	Description
CAD_INCIDENT_ID	An incident identifier comprising the julian date and a 4 character sequence number starting at 1 each day.
INCIDENT_DATETIME	The date and time the incident was created in the dispatch system
INITIAL_CALL_TYPE	The call type assigned at the time of incident creation.
INITIAL_SEVERITY_LEVEL_CODE	The segment(priority) assigned at the time of incident creation.
FINAL_CALL_TYPE	The call type at the time the incident closes.
FINAL_SEVERITY_LEVEL_CODE	The segment(priority) assigned at the time the incident closes.
FIRST_ASSIGNMENT_DATETIME	The date and time the first unit is assigned.
DISPATCH_RESPONSE_SECONDS_QY	The time elapsed in seconds between the incident_datetime and the first_assignment_datetime.
FIRST_ACTIVATION_DATETIME	The date and time the first unit gives the signal that it is enroute to the location of the incident.
FIRST_ON_SCENE_DATETIME	The date and time the first unit signals that it has arrived at the location of the incident.

INCIDENT_RESPONSE_SECONDS_QY	The time elapsed in seconds between the incident_datetime and the first_on_scene_datetime.
INCIDENT_TRAVEL_TM_SECONDS_QY	The time elapsed in seconds between the first_assignment_datetime and the first_on_scene_datetime.
FIRST_TO_HOSP_DATETIME	The date and time the first unit gives the signal that it is enroute to the hospital.
FIRST_HOSP_ARRIVAL_DATETIME	The date and time the first unit signals that it has arrived at the hospital.
INCIDENT_CLOSE_DATETIME	The date and time the incident closes in the dispatch system.
HELD_INDICATOR	Indicates that for some reason a unit could not be assigned immediately
INCIDENT_DISPOSITION_CODE	A code indicating the final outcome of the incident. See incident dispositions.
BOROUGH	The borough of the incident location.
INCIDENT_DISPATCH_AREA	The dispatch area of the incident.
ZIPCODE	The zip code of the incident.
POLICEPRECINCT	The police precinct of the incident.
CITYCOUNCILDISTRICT	The city council district.
COMMUNITYDISTRICT	The community district.
COMMUNITYSCHOOLDISTRICT	The community school district.
CONGRESSIONALDISTRICT	The congressional district.
REOPEN_INDICATOR	Indicates that at some point the incident was closed but then reopened.
SPECIAL_EVENT_INDICATOR	Indicates that the incident was a special event such as the NYC Marathon.
STANDBY_INDICATOR	Indicates that the units were assigned to stand by incase they were needed.
TRANSFER_INDICATOR	Indicates that the incident was created for the transportation of a patient from one facility to another.
LATITUDE	Co-Ordinate generated using ZIPCODE
LONGITUDE	Co-Ordinate generated using ZIPCODE

7.3. Full Data Types for FDNY EMS

Column	Data Type
CAD_INCIDENT_ID	int
INCIDENT_DATETIME	Factor
INITIAL_CALL_TYPE	Factor
INITIAL_SEVERITY_LEVEL_CODE	int
FINAL_CALL_TYPE	Factor
FINAL_SEVERITY_LEVEL_CODE	int
FIRST_ASSIGNMENT_DATETIME	Factor
DISPATCH_RESPONSE_SECONDS_QY	int
FIRST_ACTIVATION_DATETIME	Factor
FIRST_ON_SCENE_DATETIME	Factor
INCIDENT_RESPONSE_SECONDS_QY	int
INCIDENT_TRAVEL_TM_SECONDS_QY	int
FIRST_TO_HOSP_DATETIME	Factor
FIRST_HOSP_ARRIVAL_DATETIME	Factor
INCIDENT_CLOSE_DATETIME	Factor
HELD_INDICATOR	Factor
INCIDENT_DISPOSITION_CODE	int
BOROUGH	Factor

INCIDENT_DISPATCH_AREA	Factor
ZIPCODE	int
POLICEPRECINCT	int
CITYCOUNCILDISTRICT	int
COMMUNITYDISTRICT	int
COMMUNITYSCHOOLDISTRICT	int
CONGRESSIONALDISTRICT	int
REOPEN_INDICATOR	Factor
SPECIAL_EVENT_INDICATOR	Factor
STANDBY_INDICATOR	Factor
TRANSFER_INDICATOR	Factor
Longitude	int
Latitude	int

7.4. Full Data Description for DFB Ambulance

Column Name	Description
Date	Date of Incident
DFB Station Area	DFB Station Area incident occurred
DFB_Station_Area_ID	ID for DFB Station Area incident occurred
Hospital_Code	Adult A&E Catchment Area Dublin
Hospital_Code_ID	ID for Adult A&E Catchment Area Dublin
DISTRICT	Location / Sub Area / Townland
TOC	Time of call
ORD	Time Order (1st appliance)
MOB	Time MOB (1st appliance)
IA	Time IN ATTENDANCE (at Scene)
LS	Leaving Scene
AH	At Hospital
MAV	Mobile & Available
CD	Closing Down
Criticality_Code	Level of severity
Criticality_Code_ID	Unique ID for level of severity
TOC_ORD_Mins	Time in munutes between TOC & ORD
ORD_MOB_Mins	Time in munutes between ORD & MOB
MOB_IA_Mins	Time in munutes between MOB & IA
IA_LS_Mins	Time in munutes between IA & LS
LS_AH_Mins	Time in munutes between LS & AH
AH_MAV_Mins	Time in munutes between AH & MAV
MAV_CD_Mins	Time in munutes between MAV & CD
TOC_IA_Mins	Time in munutes between TOC & IA
ORD_IA_Mins	Time in munutes between ORD & IA
LS_CD_Mins	Time in munutes between LS & CD
TOC_CD_Mins	Time in munutes between TOC & CD

7.5. DFB Full Data Type

Column	Data Type
Date	Factor
DFB_Station	Factor
DFB_Station_ID	int
Hospital_Code	Factor
Hospital_Code_ID	int
District	Factor
Description	Factor
TOC	Factor
ORD	Factor
MOB	Factor
IA	Factor
LS	Factor
AH	Factor
MAV	Factor
CD	Factor
Criticality_Code	Factor
Criticality_ID	int
TOC_ORD_Mins	int
ORD_MOB_Mins	int
MOB_IA_Mins	int
IA_LS_Mins	int
LS_AH_Mins	int
AH_MAV_Mins	int
MAV_CD_Mins	int
TOC_IA_Mins	int
ORD_IA_Mins	int
LS_CD_Mins	int
TOC_CD_Mins	int

7.6. Descriptive Statistics for Response Time per Category (DFB)

Descriptives

	Criticality_ID			Statistic	Std. Error
TOC_IA_Mins	Omega	Mean		38.51	2.686
		95% Confidence Interval for Mean	Lower Bound	33.22	
			Upper Bound	43.81	
		5% Trimmed Mean		34.05	
		Median		25.00	
		Variance		1392.126	
		Std. Deviation		37.311	
		Minimum		5	
		Maximum		243	

		Range	238	
		Interquartile Range	35	
		Skewness	2.195	.175
		Kurtosis	6.271	.348
	Alpha	Mean	35.29	1.308
		95% Confidence Interval for Mean	Lower Bound	32.73	
			Upper Bound	37.86	
		5% Trimmed Mean	30.30	
		Median	22.00	
		Variance	1266.733	
		Std. Deviation	35.591	
		Minimum	2	
		Maximum	249	
		Range	247	
		Interquartile Range	27	
		Skewness	2.490	.090
		Kurtosis	7.308	.179
	Bravo	Mean	25.75	.894
		95% Confidence Interval for Mean	Lower Bound	24.00	
			Upper Bound	27.51	
		5% Trimmed Mean	22.34	
		Median	17.00	
		Variance	604.192	
		Std. Deviation	24.580	
		Minimum	3	
		Maximum	245	
		Range	242	
		Interquartile Range	18	
		Skewness	3.412	.089
		Kurtosis	17.545	.178
	Charlie	Mean	28.83	.909
		95% Confidence Interval for Mean	Lower Bound	27.05	
			Upper Bound	30.62	
		5% Trimmed Mean	25.33	
		Median	20.00	
		Variance	687.176	
		Std. Deviation	26.214	
		Minimum	1	
		Maximum	235	
		Range	234	

		Interquartile Range	22	
		Skewness	2.951	.085
		Kurtosis	12.662	.169
	Delta	Mean	21.93	.405
		95% Confidence Interval for Mean Lower Bound	21.14	
		95% Confidence Interval for Mean Upper Bound	22.72	
		5% Trimmed Mean	19.20	
		Median	15.00	
		Variance	392.436	
		Std. Deviation	19.810	
		Minimum	0	
		Maximum	225	
		Range	225	
		Interquartile Range	14	
		Skewness	3.412	.050
		Kurtosis	17.589	.100
	Echo	Mean	11.29	.688
		95% Confidence Interval for Mean Lower Bound	9.93	
		95% Confidence Interval for Mean Upper Bound	12.66	
		5% Trimmed Mean	10.58	
		Median	9.00	
		Variance	40.234	
		Std. Deviation	6.343	
		Minimum	4	
		Maximum	40	
		Range	36	
		Interquartile Range	7	
		Skewness	1.981	.261
		Kurtosis	5.403	.517

7.7. Normality Result for Response Time per Category (DFB)

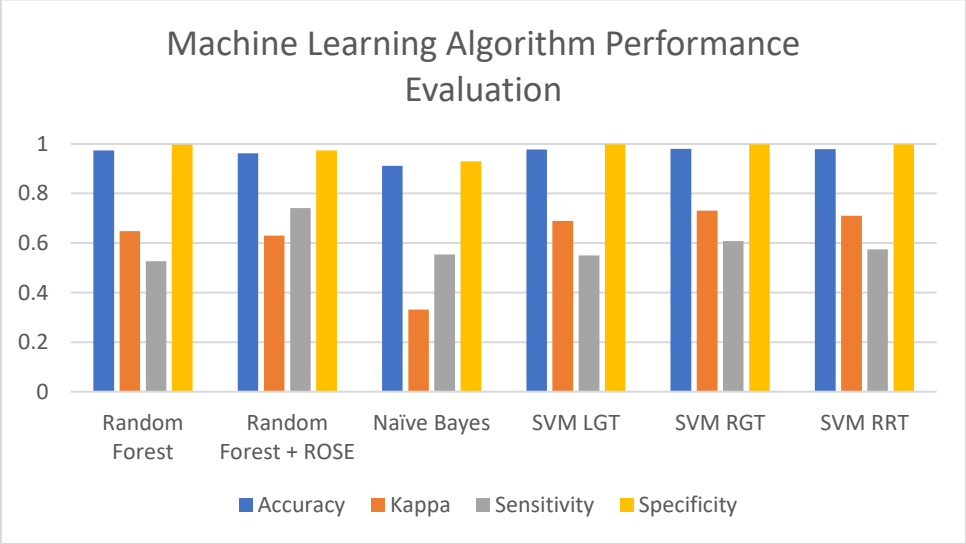
Tests of Normality

		Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	Criticality_ID	Statistic	df	Sig.	Statistic	df	Sig.
TOC_IA_Mins	Omega	.199	193	.000	.761	193	.000
	Alpha	.205	740	.000	.707	740	.000
	Bravo	.210	756	.000	.665	756	.000
	Charlie	.194	832	.000	.706	832	.000
	Delta	.210	2394	.000	.671	2394	.000
	Echo	.169	85	.000	.817	85	.000

a. Lilliefors Significance Correction

7.8. All Machine Learning Model Performance's

Model	Accuracy	Kappa	Sensitivity	Specificity
Random Forest	0.9742	0.6486	0.52655	0.99667
Random Forest + ROSE	0.9624	0.6304	0.74075	0.97338
Naïve Bayes	0.912	0.3321	0.55427	0.92985
SVM Linear Grid Tuned	0.9774	0.6894	0.55	0.9989
SVM Radial Grid Tuned	0.9796	0.731	0.6083	0.9983
SVM Random Radial Tuned	0.9786	0.7102	0.575	0.9989



7.9. October Monthly Reflective Journal

My Achievements

This month, I received the data that I requested from the Dublin Fire Brigade for my project. When I received the data, I arranged a meeting my supervisor to show the data and see what he that and how I could go on using this data. Unfortunately, he felt that although there were 150,000 rows of ambulance call data, there wasn't enough numeric values or distinct categorical data to be able to perform any descriptive or statistical analysis. This was a big setback for me but was determined to find more data to be able to stick with this topic.

I spent some time looking for data like the data I received, and on data.gov, I found EMS Incident Response data for New York. I opened the data in Excel, and it reached the maximum number of rows, however, I was confident there was enough data in this dataset for the supervisor to say it will work. Once again, I arranged to meet the supervisor, showed him the data, and he was impressed with the results. I was advised to use the New York data as my primary dataset, with the Dublin Fire Brigade dataset as a secondary, and to use to reference and compare.

My Reflection

At first, I was disappointed that the Dublin Fire Brigade data would not be enough to proceed with the project and felt this was a major setback. I spent up to 2 weeks thinking of what else I could do my project on, and with the project proposal deadline looming, I needed an idea soon. I started stressing and was finding it increasingly difficult to find data on something that I felt interested in. I felt determined and wanted to stick with my original idea, so I went looking online for hours until I came across data that I could use to relate to the data I received. I was thinking of ideas along the lines of:

- A&E Department Waiting Times
- How weather can affect the duration of an ambulance
- How traffic can delay an ambulance from being able to arrive to the scene sooner

When looking for the supporting data, I could not find any data that I felt would be enough to support the primary data. When I found the New York EMS data, there was a sigh of relief, because I knew that the quality of the data would be good enough to be the primary data and use the Dublin data as a comparison.

One issue that I am finding with the data is loading it into R Studio. Due to the large volume of data, it is taking over 30-45 minutes for the data to load in, and whilst I have a module that uses R Studio a lot, I would be constantly refreshing the environment which would delete the loaded csv every time.

Intended Changes

Next month, I plan on getting a start on the data cleansing, and maybe some data analysis techniques. As well as this, I would like to have an environment set up that will be dedicated to just having the New York EMS data loaded into the R Studio.

With the deadline of some Continuous Assessments for other modules, and a test after reading week, I feel that I have pushed my final year project off to one side for now. I plan on changing that this month and assigning more time to focusing on the project to relieve some of the pressure that will undoubtedly be encountering closer to the deadline.

Supervisor Meetings

Date of Meeting: 4.10.19

Items discussed:

- Dataset received from Dublin Fire Brigade

- What can be done for the project with the current data
- What should be included in the video interview

Action Items:

- Find alternatives or more data for the primary dataset
- Record the video interview and mention what data analysis techniques will be used on the data

Date of Meeting: 18.10.19

Items discussed:

- New York dataset that I found
    - Excel will not load all the rows, its maxing at 1 million rows
- My project proposal

Action Items:

- Load the csv into R Studio to find the total number of rows
- Update the project proposal to mention that the New York data will be the primary dataset while the Dublin Fire Brigade data will be the secondary dataset and used for comparing

7.10. November Monthly Reflective Journal

My Achievements

This month, I had to reduce the number of rows that were in the NYC dataset. While it would have been much better to perform analysis on the larger dataset, the hardware in my device was just unable to cope with the large volume of data and was taking in excess of 45-60 minutes just to load the data into R studio. The dataset has been compressed from 8.5 million rows to 4.5 million. This should make it less hardware intense when performing analysis.

I have still been performing different analysis techniques with the data and will import 3 or 4 visual representations into my mid-point presentation, and continue to create more to display my findings much clearer

I have also created my first draft of my mid-point presentation. Once I am happy with the draft, I am going to arrange a meeting with my supervisor and will request feedback and will make necessary changes to ensure it is ready to be presented

My Reflection

I found this month to be quite tough to progress on the project. There were some deadlines that needed to be met in November for other modules and I gave them priority over the final

project. Looking back, I would have made changes to how I should have approached it, but it is in the past and will need to do extra in December to catch up.

Intended Changes

For the month of December, my intentions are to get much more complete than I did this month. I want to finalize the visual representations that will be going into the project presentation.

Due to the size reduction in the main dataset, I did not get much data cleansing done as mentioned last month. However, the bit I did, I had to cancel the process due to the duration of time it was taking. So for December, I plan on preparing a mice function to perform on the data in R, and once I am happy that it will work, I am going to run the imputation throughout the duration it will take, which could be 15-20 hours.

I will also be removing entire rows where if there are any null values in a cell where there should be a date or time. My idea behind this is, although it is possible to predict, I want the time date data to be as accurate as possible and do not want any errors or outliers when performing time series analysis with the data.

Supervisor Meetings

Date of Meeting: November 11th, 2019

Items discussed:

- NYC dataset size
- The best option is to reduce the size
- Prepare for the mid-point presentation
- Current ETL progress

Action Items:

- Have a PowerPoint complete
    - Show the idea
    - How I came up with the idea
    - Dataset Overview
    - Visual Representations
- Display code snippets
- Have charts and descriptive's completed

7.11. December Monthly Reflective Journal

The focus of this month was to prepare the midpoint presentation and to begin studying for the winter exams that are being held between January 3rd and 11th. I began creating the presentation

inserting some information about the data, what analysis I wanted to perform, and some preliminary analysis that I thought would be interesting. I created the first draft of the presentation and when I was happy, I arranged to meet with Dr. Eugene O'Loughlin to show him the first draft and seek feedback for what I currently have. Some suggestions that he made were:

- Change the colour on the pie chart
- Add some more detail
- Get some descriptive's from the data
- Add the map I created for New York

I left the meeting with excellent feedback and immediately introduced the suggestions. I was happy that the final draft was complete and uploaded the finished presentation. I was practicing the presentation for the days in the build up to the presentation, trying to prepare for any questions that could arise.

My presentation was held on the 18th of December where I presented some preliminary results. I done the presentation and answered the questions asked to the best of my ability. I felt that it went very well.

Now that the presentation was done, its then time to start focusing on studying for the exams. I tried to use a strategy that I would not usually consider effective for me to study as I feel I learn better when I do practical work. Instead, this time I decided to write notes on paper. Although I was not enjoying making the notes and was feeling like I was wasting my time, I continued writing the notes. I felt the more I done it, the easier it began to get.

As the exams are in next month, I do not plan on making any progress on the project until February earliest. I will arrange to meet with Dr. Eugene O'Loughlin when lectures resume after the exam period to seek feedback from the midpoint presentation.

7.12. January Monthly Reflective Journal

Like December, I did not plan to get much progress done on the project. The Winter exams were my main priority to ensure I do the best I can to achieve my goal of a 1st class honours. I had 4 exams and I felt confident after each one that they went very well from my perspective. It was just a waiting game until February to get the results.

I did make time to arrange a meeting with Dr. Eugene O'Loughlin to seek some feedback from the midpoint presentation of my project that I had presented in December. The overall feedback that he gave was with some areas to touch up. Both markers seemed very interested in the

idea when I was presenting it and my result gave me the motivation to keep pushing on with this idea.

We chatted for a while about where my project currently is, and what I need to do to make sure I keep on track. He mentioned that without machine learning, my result will not be as good as I would expect and that I should investigate implementing some small models just to get familiar with the concept. I should also start doing some research on the topic that I will reference when I begin writing the final report.

Other than that, there was not much done in January. Looking ahead to February, I would like to go back to my data and see where I could potentially add some more data to the dataset using the information that is already in the data, just manipulating what I have.

7.13. February Monthly Reflective Journal

One of the key issues with the data I was provided for Dublin is there is a latitude & longitude, but it is in Ordinance Survey Ireland (OSI) format, not traditional latitude and longitude. I found a link on the OSI website that I was able to convert the co-ordinated into degrees, minutes & seconds location points. I then found another website that was able to convert these into the traditional latitude and longitude. I decided to try automating this using Python & Selenium. I spent a good length of time in February trying to get to grips with learning the programming language. I felt I had a great opportunity to use this language to help me use the data that I had to produce more data for additional analysis.

I met with my supervisor, Dr. Eugene O'Loughlin, to demonstrate what I had working. He seemed impressed with what I had so far but warned me that it might not be possible to proceed with the automation. He advised me to check the policies that the companies behind the websites have as they may not allow for the use of automation to collect and retrieve data. I took this information and after the meeting I went researching. The OSI website did not explicitly say that the use of automation was restricted from their website, but it also never said that it was permitted. I rang OSI if they were able to provide a converter instead of me triggering tens of thousands of requests to their website and they said none was available. I did not request permission to perform the web automation as I felt I already knew the answer. The 2nd conversion website would not have been an issue, as they also listed the formula that is used to do the conversion.

I accepted that I would not be able to implement this into my project, so now I had time towards the end of the month to make more data out of what was given to me, using a different method. As most of the data was times, I decided to get the time difference between all the stages,

then some of the more relevant stages. I then converted the time difference to integers so I will be able to do some further analysis on the times.

I also received my winter exam results, which I was extremely happy with and can proudly say that I am on track to achieving the 1st class honours that I would like to graduate with!

Looking ahead for March, I am going to begin the report and getting the initial analysis started and look into some machine learning algorithms that I have been learning these past few weeks in the Data & Web Mining lecture that is teaching some very interesting predictive analysis modelling that I will seek to implement into my project.

7.14. March Monthly Reflective Journal

March had started off going well. I was beginning to get some good progress on returning to the original data and adding more options to the data. I finished adding more columns of data that I felt could be relevant to future analysis. I was then told that there was a special offer for an annual subscription for full access to DataCamp. I had used DataCamp in the past when I was beginning to learn R. However, because it was the Free version, I only had access to the 1st chapter, which was very limited, but I felt it made it easier to learn. I decided to purchase the DataCamp subscription and will begin learning all the modules that are on offer.

However, not everything went according to plan. The main device that I have used for the duration of my project, a Dell XPS, ended up breaking. This device has an Intel i7 & 16GB of RAM. For the analysis I was doing, it was good enough to get the work done, but was taking a long time to complete the processes in R. The laptop breaking is a major inconvenience for me. Having spent several hours of troubleshooting, I finally identified that the battery was the issue and I promptly ordered a replacement.

In the meantime, I used a desktop PC that I have at home, however, it is not as near powerful as the laptop, as it only has 8GB of RAM. I spent hours trying to run some small analysis models and R Studio kept crashing. All these issues left me very nervous for the progress on my project. To try increase productivity and reduce wait times, I decided the best option for me was to upgrade my desktop PC to 32GB of RAM, which with minimal analysis so far, seems to have boosted the speed

When the battery for the laptop finally arrived, I replaced it and discovered that the hard drive in the laptop had corrupted and I had complete data loss. This has proven to be very costly for me as I was not backing up my data, or my R files for the data analysis. My plan for the remaining

duration of the project is to use the upgraded desktop PC for doing the heavy lifting work and the laptop for using the less resource intensive work.

If this has not been enough pain, Ireland was impacted with Covid-19 which has seen college closing. All exams have been cancelled and lectures have moved online. I have now been given 2 additional continuous assessments, so managing time has been rather challenging.

I had 2 meetings with my Dr. Eugene O'Loughlin this month. One was right before the college closed and the 2nd was on the 26th of March where the same things were discussed just added additional information about how things will proceed going forward. Some items discussed are:

- Begin running Machine Learning models on my data
- Get starting on the literature review
- Perform initial descriptive and statistical analysis

This month did not go according to plan. However, things can only start to look up. I am going to carry over the work that I had planned for March into April, as well as make some significant progress on the technical report, to the point where it is or very nearly is complete draft pending changes.

7.15. April Monthly Reflective Journal

This month was by far my best month in terms of progression. I achieved everything that I set out in March what I was going to do. My goal was to get the first draft of my technical report complete so I had 10 days to make any changes without stressing about it not being complete.

The first two weeks of April were devoted to completing the final assessments, which was two additional reports that needed to be complete for the two modules that I was supposed to have an exam in. the exams were cancelled due the global pandemic, and the reports were the substitute. As well as working on these, I was performing some analysis and data cleansing for this project.

I had weekly progress meetings with Dr. Eugene O'Loughlin where we discussed what I have achieved since our last meeting, asked some questions that I had and he advised me what I should have done and where I should be by the time we next spoke.

I essentially started from scratch for the project, as I was not backing up my files or using GitHub as effectively as I should have and losing the data on the hard drive on the laptop in March set me back. I was not going to let this get in my way, and I spent as much time as possible getting back to the point that I was at. Once the two reports were submitted, I focused all my efforts on wanting to complete the first draft as soon as possible.

I completed the data cleansing and preparation and loaded the data into Tableau to create the visualisations. In the background, I was deciding which machine learning models to use and what I want to use them to try and predict.

My workplace gave me permission to use my desk on my days off to work on the project, so I was working on the report during the day time while using my laptop, then when I got home I would use the now more powerful computer to do the hardware intensive work such as running the machine learning models. I used the for loop that was created to obtain the latitude and longitude for all the NYC calls in the data I had on a sample of 80,000 calls.

When I completed the three machine learning models, I decided to revisit the loop getting the co-ordinates for the 80,000 calls, and instead get them for the 3 million calls. This process took over 22 hours to complete, 12 hours longer than anticipated. I was still working in the background on the report, pushing to reach my goal.

On April 30th exactly, I had my first draft of the document ready for proof reading, with 10 days remaining to make the minor changes, and to discuss with Dr. Eugene O'Loughlin on our weekly meeting that is arranged for May 1st.

My plans for May are to:

- Create the R Shiny dashboard of the interactive map of NYC Calls and publish it
- Start creating the presentation and prepare to record and upload it
- Upload the finished technical report and code on May 10th
- Create a poster for the project showcase

7.16. Project Code

```
# Final Year Project

# Carl O'Beirne

# Analysis of Ambulance Responses



############



############ Packages ############

library("data.table")
```

```
library("zipcode")

library("caret")

library("randomForest")

library("ROSE")

library("leaflet")

library("tidyverse")

library("ggthemes")

library("leaflet.extras")

library("magrittr")

library("e1071")

library("mlbench")


############ Read Datasets ############

DFB_EMS_Data <- fread("Data/DFB_EMS_Data.csv", sep = ",", stringsAsFactors = T, na.strings= "")

NYC_EMS_Data <- fread("Data/NYC_EMS_Data.csv", sep = ",", stringsAsFactors = T, na.strings = "") # No longer required to be read

################################################


############ Reading Cleaned Datasets ############

NYC_EMS_Data <- fread("Data/NYC_EMS_Data.csv", header = T, sep = ",", stringsAsFactors = T, na.strings = "")

DFB_EMS_Data <- read.csv("Data/NEW_DFB_EMS_Data.csv", header = T, sep = ",", stringsAsFactors = T, na.strings = "")

NYC_EMS_MapSample <- fread("Data/NYC_EMS_MapData.csv", header = T, sep = ",", stringsAsFactors = T, na.strings = "")
```

```
##############################################

############ Data Cleansing DFB ############

str(DFB_EMS_Data)

DFB_EMS_Data$Date <- as.POSIXct(DFB_EMS_Data$Date, format = "%d/%m/%Y") # Changning data format to readable R date format

# Removing irrelevant columns

DFB_EMS_Data <- DFB_EMS_Data[, -c(2,19,20,21,23,25,27,29,31,33,35,37,39,41)]

# Checking for NA values

sapply(DFB_EMS_Data, function(x) sum(is.na(x))) # Make sure there are no NA values

# Removing Obs with NA values in particular variables

DFB_EMS_Data <- DFB_EMS_Data[complete.cases(DFB_EMS_Data[, c(4,6,10:12,14)]), ] # Removing obs with blanks

DFB_EMS_Data$Date <- as.POSIXct(paste(DFB_EMS_Data$Date,DFB_EMS_Data$TOC))

DFB_EMS_Data <- subset(DFB_EMS_Data, TOC_CD_Mins <= 360 & TOC_IA_Mins <= 360 & AH_MAV_Mins <= 360) # Removing data where call length > 6 hours

# Saving cleaned file for faster readoing

write.csv(DFB_EMS_Data, "Data/NEW_DFB_EMS_Data.csv", row.names = F)
```

```
##############################################

############ Data Cleansing NYC ############

str(NYC_EMS_Data)

sapply(NYC_EMS_Data, function(x) sum(is.na(x))) # Make sure there are no NA values

# Dropping level in borough - set as unknown and there are only 5 boroughs

NYC_EMS_Data$BOROUGH <- droplevels(NYC_EMS_Data$BOROUGH, "UNKNOWN")

# Removing final call type unknown

NYC_EMS_Data$FINAL_CALL_TYPE <- droplevels(NYC_EMS_Data$FINAL_CALL_TYPE,'UNKNOW')

# Changing date format

NYC_EMS_Data$INCIDENT_DATETIME <- as.POSIXct(NYC_EMS_Data$INCIDENT_DATETIME, format =
"%m/%d/%Y %I:%M:%S %p")

# NYC2019 <- NYC_EMS_Data[ NYC_EMS_Data$INCIDENT_DATETIME >= as.POSIXct("2019-01-01
00:00:00") & NYC_EMS_Data$INCIDENT_DATETIME <= as.POSIXct("2019-12-31 23:59:59"), ] # Only
created to get the number of calls made in 2019 for results in report

# Reducing data to just 2017/2018

NYC_EMS_Data <- NYC_EMS_Data[ NYC_EMS_Data$INCIDENT_DATETIME >= as.POSIXct("2017-01-
01 00:00:00") & NYC_EMS_Data$INCIDENT_DATETIME <= as.POSIXct("2018-12-31 23:59:59"), ]

# Changing T/F values to Y/N for consistency

NYC_EMS_Data$HELD_INDICATOR <- gsub('false', 'N', NYC_EMS_Data$HELD_INDICATOR)

NYC_EMS_Data$HELD_INDICATOR <- gsub('true', 'Y', NYC_EMS_Data$HELD_INDICATOR)

NYC_EMS_Data$HELD_INDICATOR <- as.factor(NYC_EMS_Data$HELD_INDICATOR)

NYC_EMS_Data$VALID_DISPATCH_RSPNS_TIME_INDC <- gsub('false', 'N',
NYC_EMS_Data$VALID_DISPATCH_RSPNS_TIME_INDC)

NYC_EMS_Data$VALID_DISPATCH_RSPNS_TIME_INDC <- gsub('true', 'Y',
NYC_EMS_Data$VALID_DISPATCH_RSPNS_TIME_INDC)
```

```
NYC_EMS_Data$VALID_DISPATCH_RSPNS_TIME_INDC <-
as.factor(NYC_EMS_Data$VALID_DISPATCH_RSPNS_TIME_INDC)


NYC_EMS_Data$VALID_INCIDENT_RSPNS_TIME_INDC <- gsub('false', 'N',
NYC_EMS_Data$VALID_INCIDENT_RSPNS_TIME_INDC)

NYC_EMS_Data$VALID_INCIDENT_RSPNS_TIME_INDC <- gsub('true', 'Y',
NYC_EMS_Data$VALID_INCIDENT_RSPNS_TIME_INDC)

NYC_EMS_Data$VALID_INCIDENT_RSPNS_TIME_INDC <-
as.factor(NYC_EMS_Data$VALID_INCIDENT_RSPNS_TIME_INDC)


NYC_EMS_Data$REOPEN_INDICATOR <- gsub('false', 'N', NYC_EMS_Data$REOPEN_INDICATOR)

NYC_EMS_Data$REOPEN_INDICATOR <- gsub('true', 'Y', NYC_EMS_Data$REOPEN_INDICATOR)

NYC_EMS_Data$REOPEN_INDICATOR <- as.factor(NYC_EMS_Data$REOPEN_INDICATOR)


NYC_EMS_Data$SPECIAL_EVENT_INDICATOR <- gsub('false', 'N',
NYC_EMS_Data$SPECIAL_EVENT_INDICATOR)

NYC_EMS_Data$SPECIAL_EVENT_INDICATOR <- gsub('true', 'Y',
NYC_EMS_Data$SPECIAL_EVENT_INDICATOR)

NYC_EMS_Data$SPECIAL_EVENT_INDICATOR <-
as.factor(NYC_EMS_Data$SPECIAL_EVENT_INDICATOR)


NYC_EMS_Data$STANDBY_INDICATOR <- gsub('false', 'N', NYC_EMS_Data$STANDBY_INDICATOR)

NYC_EMS_Data$STANDBY_INDICATOR <- gsub('true', 'Y', NYC_EMS_Data$STANDBY_INDICATOR)

NYC_EMS_Data$STANDBY_INDICATOR <- as.factor(NYC_EMS_Data$STANDBY_INDICATOR)


NYC_EMS_Data$TRANSFER_INDICATOR <- gsub('false', 'N', NYC_EMS_Data$TRANSFER_INDICATOR)

NYC_EMS_Data$TRANSFER_INDICATOR <- gsub('true', 'Y', NYC_EMS_Data$TRANSFER_INDICATOR)

NYC_EMS_Data$TRANSFER_INDICATOR <- as.factor(NYC_EMS_Data$TRANSFER_INDICATOR)


#Checking for NA Values

sapply(NYC_EMS_Data, function(x) sum(is.na(x))) # Make sure there are no NA values


NYC_EMS_Data <- NYC_EMS_Data[complete.cases(NYC_EMS_Data[, c(7,10,11,13:17,19,20,22:27)]),
] # Removing obs with blanks
```

```
# Removing irrelevant columns
NYC_EMS_Data <- NYC_EMS_Data[, -c(8,12)]

# Saving Cleaned File for quicker reading
write.csv(NYC_EMS_Data, "Data/NYC_EMS_Data.csv", row.names = F)

##############################################

########### Functions ###########
# Obtain Latitude & Longitude from Zip Code
zipcode <- read.csv("Data/NYC_ZipCodes.csv")

for (i in 1:nrow(NYC_EMS_MapSample)){
   if(length(zipcode$zip[NYC_EMS_MapSample$ZIPCODE[i] == zipcode$zip]) == 1){
      NYC_EMS_MapSample$Latitude[i] <- zipcode$latitude[NYC_EMS_MapSample$ZIPCODE[i] ==
zipcode$zip]
      NYC_EMS_MapSample$Longitude[i] <- zipcode$longitude[NYC_EMS_MapSample$ZIPCODE[i] ==
zipcode$zip]
      print(paste("Row: ",i, "Zip Code:
",NYC_EMS_MapSample$ZIPCODE[i],NYC_EMS_MapSample$Latitude[i],NYC_EMS_MapSample$Lon
gitude[i], "Status: ", TRUE, Sys.time()))
   }else{
      NYC_EMS_MapSample$Latitude[i] <- NA
      NYC_EMS_MapSample$Longitude[i] <- NA
      print(paste("Row: ",i, "Zip Code:
",NYC_EMS_MapSample$ZIPCODE[i],NYC_EMS_MapSample$Latitude[i],NYC_EMS_MapSample$Lon
gitude[i], "Status: ", FALSE, Sys.time()))

   }
}

write.csv(NYC_EMS_MapSample, "Data/NYC_EMS_MapData.csv", row.names = F)# Saving the file
rm(i)
rm(zipcode)
```

```
# Function to find the mode for any data
findMode <- function(x){
   uniqueVals <- unique(x)
   uniqueVals[which.max(tabulate(match(x, uniqueVals)))]
}

##############################################

############ Start of Analysis ############

############ Model 1 - RF to Predict whether the incident will be HELDUP in NYC ############

#NYC RF Sample
set.seed(546513) # Reproducability
index <- sample(1:nrow(NYC_EMS_Data), 300000, replace = F)
NYC_EMS_RFSample <- NYC_EMS_Data[index, ]

sapply(NYC_EMS_RFSample, function(x) sum(is.na(x))) # Make sure there are no NA values



str(NYC_EMS_RFSample)

NYC_EMS_RFSample <- NYC_EMS_RFSample[, -c(1:5,7:10,13:15,20)] # Removing variables with
factors > 53 levels & unrelated variables

# Create Train & Test Data
set.seed(325146) # Reproducability

index <- sample(1:nrow(NYC_EMS_RFSample), 0.75*nrow(NYC_EMS_RFSample), replace = F )
nycTrainRF <- NYC_EMS_RFSample[index, ]
nycTestRF <- NYC_EMS_RFSample[-index, -4]

actualHeldUpRF <- NYC_EMS_RFSample[-index, 4]
```

```
rm(index)

nyc_rf_model <- randomForest(HELD_INDICATOR ~., nycTrainRF)

varImpPlot(nyc_rf_model)

nyc_rf_pred <- predict(nyc_rf_model, nycTestRF)

heldupCMRF <- confusionMatrix(nyc_rf_pred, actualHeldUpRF, positive = "Y")

heldupCMRF

rm(nyc_rf_pred)
rm(nycTestRF)
rm(nycTrainRF)
rm(nyc_rf_model)
rm(actualHeldUpRF)

# ROSE (Random Over Sampling Examples)
# Oversampling with Rose
set.seed(69745)
index <- sample(2, nrow(NYC_EMS_RFSample), replace = T, prob = c(0.75,0.25))
roseTrain <- NYC_EMS_RFSample[index == 1, ]
roseTest <- NYC_EMS_RFSample[index == 2, -4 ]

actualHeldUpROSE <- NYC_EMS_RFSample[index == 2, 4]

rm(index)

table(roseTrain$HELD_INDICATOR) # See how many obs there are for each class
```

```
heldup <- ovun.sample(HELD_INDICATOR ~., data=roseTrain, method = "over", N = 428594)$data # Oversample the most frequent

table(heldup$HELD_INDICATOR)

#Creating Model
rfTrainRose <- randomForest(HELD_INDICATOR ~., heldup)

#Evaluate Model with Test Data
rosePred <- predict(rfTrainRose, roseTest)

heldupRoseCM <- confusionMatrix(rosePred, actualHeldUpROSE, positive = "Y")

heldupRoseCM

rm(rosePred)
rm(roseTest)
rm(roseTrain)
rm(rfTrainRose)
rm(heldup)
rm(actualHeldUpROSE)
rm(NYC_EMS_RFSample)

########### ML Model 2 - NB to Predict whether the incident will be held up ###########
# NYC NB Sample
set.seed(215145) # Reproducability
index <- sample(1:nrow(NYC_EMS_Data), 300000, replace = F)
NYC_EMS_NBSample <- NYC_EMS_Data[index, ]

sapply(NYC_EMS_NBSample, function(x) sum(is.na(x))) # Make sure there are no NA values

str(NYC_EMS_NBSample)
NYC_EMS_NBSample <- NYC_EMS_NBSample[, -c(1:5,7:9,10,13:15)]
```

```
# Creating Train & Test Set
set.seed(234158) # Reproducability

index <- sample(1:nrow(NYC_EMS_NBSample), 0.75*nrow(NYC_EMS_NBSample), replace = F )
nycTrainNB <- NYC_EMS_NBSample[index, ]
nycTestNB <- NYC_EMS_NBSample[-index, -4]

actualHeldUpNB <- NYC_EMS_NBSample[-index, 4]

rm(index)

nb_model <- naiveBayes(HELD_INDICATOR ~., nycTrainNB)

nb_pred <- predict(nb_model, nycTestNB)

heldupNBCM <- confusionMatrix(nb_pred, actualHeldUpNB, positive = "Y")

heldupNBCM

rm(nb_pred)
rm(actualHeldUpNB)
rm(nycTestNB)
rm(nycTrainNB)
rm(NYC_EMS_NBSample)
rm(nb_model)

##########################################################################################

########### ML Model 3 - SVM to Predict whether the incident will be held up ###########

##########################################################################################
#####
```

```
####### N.B. - THE CODE FOR SVM MACHINE LEARNING MODEL (LINE 271-447) IS ADAPTED FROM LAB FROM COLLEGE LAB LECTURED BY MR. NOEL COSGRAVE OF THE NATIONAL COLLEGE OF IRELAND ######

################################################################################################

NYC_EMS_SVMSample <- NYC_EMS_Data

sapply(NYC_EMS_SVMSample, function(x) sum(is.na(x))) # Make sure there are no NA values

# NYC SVM Sample

set.seed(65451) # Reproducability

index <- sample(1:nrow(NYC_EMS_Data), 20000, replace = F)

NYC_EMS_SVMSample <- NYC_EMS_Data[index, ]

rm(index)

str(NYC_EMS_SVMSample)

NYC_EMS_SVMSample_Numbers <- NYC_EMS_SVMSample[, c(6,8,11,12,16,17,20:25)] # Keeping Just Numeric Columns for SVM

rm(NYC_EMS_SVMSample)

str(NYC_EMS_SVMSample_Numbers)

sapply(NYC_EMS_SVMSample_Numbers, function(x) sum(is.na(x)))

set.seed(65451)

index <- createDataPartition(

  NYC_EMS_SVMSample_Numbers$HELD_INDICATOR,

  p = .75,

  list = F

)

svm_train <- NYC_EMS_SVMSample_Numbers[index, ]

svm_test <- NYC_EMS_SVMSample_Numbers[-index, ]
```

```
rm(index)

# Linear Based SVM with Tuning Grid #
Cost <- 2^c(1:8)
Cost

set.seed(65451)
svm_gt_control <- trainControl(
   method = "cv",
   number = 10,
   summaryFunction = defaultSummary
)
set.seed(65451)
svm_linear_grid <- expand.grid(
   C = Cost
)
set.seed(65451)
svm_lgt_model1 <- train(
   HELD_INDICATOR ~.,
   data = svm_train,
   method = "svmLinear",
   trControl = svm_gt_control,
   preProc = c("center", "scale", "nzv"),
   verbose = F,
   tuneGrid = svm_linear_grid
)

svm_lgt_model1

Cost = 2^seq(0,2,0.1)
Cost
```

```
set.seed(65451)
svm_linear_grid2 <- expand.grid(
   C = Cost
)

set.seed(65451)
svm_lgt_model2 <- train(
   HELD_INDICATOR ~.,
   data = svm_train,
   method = "svmLinear",
   trControl = svm_gt_control,
   preProc = c("center", "scale", "nzv"),
   verbose = F,
   tuneGrid = svm_linear_grid2
)
svm_lgt_pred

set.seed(65451)
svm_lgt_pred <- predict(
   svm_lgt_model2,
   svm_test[, -5]
)

svm_lgt_pred <- confusionMatrix(
   data = svm_pred,
   reference = svm_test[, 5],
   positive = "Y"
)

svm_cm
```

```
# Radial Based SVM with Tuning Grid #
set.seed(65451)
svm_rbf_grid <- expand.grid(
   C = 2^seq(3,5,0.1),
   sigma = 2^c(-25,-20,-15,-1,-5,0)
)
set.seed(65451)
sigma_svm_model <- train(
   HELD_INDICATOR ~.,
   data = svm_train,
   method = "svmRadial",
   trControl = svm_gt_control,
   preProc = c("center", "scale", "nzv"),
   verbose = F,
   tuneGrid = svm_rbf_grid
)

sigma_svm_model$bestTune

set.seed(65451)
svm_rbf_pred <- predict(
   sigma_svm_model,
   svm_test[, -5]
)

svm_rbf_cm <- confusionMatrix(
   svm_rbf_pred,
   svm_test[, 5],
   positive = "Y"
)

svm_rbf_cm
```

```
# Radial Based SVM with Random Tuned #
set.seed(65451)
svm_rndm_control <- trainControl(
   method = "cv",
   number = 10,
   summaryFunction = defaultSummary,
   search = "random"
)

set.seed(65451)
svm_rndm_model <- train(
   HELD_INDICATOR ~.,
   data = svm_train,
   method = "svmRadial",
   trControl = svm_rndm_control,
   preProc = c("center", "scale", "nzv"),
   verbose = F,
   tuneLenght = 60
)

svm_rndm_model$bestTune

set.seed(65451)
svm_rndm_pred <- predict(
   svm_rndm_model,
   svm_test[, -5]
)

svm_rndm_cm <- confusionMatrix(
   svm_rndm_pred,
   svm_test[, 5],
```

```
   positive = "Y"
)

svm_rndm_cm



rm(svm_rndm_control)
rm(svm_rndm_pred)
rm(svm_rndm_model)
rm(svm_rbf_grid)
rm(svm_rbf_pred)
rm(sigma_svm_model)
rm(svm.control)
rm(svm_train)
rm(svm_test)
rm(svm_linear_grid)
rm(svm_linear_grid2)
rm(svm_model1)
rm(svm_model2)
rm(svm_pred)
rm(Cost)
rm(NYC_EMS_SVMSample_Numbers)

#############################################################################

NYC_EMS_MapData$Latitude <- as.numeric(NYC_EMS_MapData$Latitude)
NYC_EMS_MapData$Longitude <- as.numeric(NYC_EMS_MapData$Longitude)

# Mapping
NYC_EMS_MapData_Sample <- sample(1:nrow(NYC_EMS_MapData), 4000, replace = F)
NYC_EMS_MapData_Sample <- NYC_EMS_MapData[NYC_EMS_MapData_Sample, ]
```

```
# FDNY EMS Calls in Cluster Map
leaflet() %>%
  addTiles() %>%
  addCircleMarkers(lat = NYC_EMS_MapData_Sample$Latitude,
                   lng = NYC_EMS_MapData_Sample$Longitude,
                   popup = NYC_EMS_MapData_Sample$FINAL_CALL_TYPE,
                   clusterOptions = markerClusterOptions())

# FDNY EMS Calls Map
leaflet() %>%
  addTiles() %>%
  addMarkers( lat = NYC_EMS_MapData_Sample$Latitude,
                   lng = NYC_EMS_MapData_Sample$Longitude,
                   popup = NYC_EMS_MapData_Sample$FINAL_CALL_TYPE)

leaflet() %>%
  addTiles() %>%
  addHeatmap( lat = NYC_EMS_MapData$Latitude,
              lng = NYC_EMS_MapData$Longitude,
              blur = 25,
              radius = 15)

rm(NYC_EMS_MapSample)
rm(NYC_EMS_MapData)

# Analysis of Call types & Call duration
NYC_CallSev_Times <- NYC_EMS_Data[, c(6,14)]
DFB_CallSev_Times <- DFB_EMS_Data[, c(1, 16,17, 25, 28)]

index <- sample(1:nrow(NYC_CallSev_Times), 15000, replace = F)
NYC_CallSev_Times <- NYC_CallSev_Times[index, ]
```

```
index <- sample(1:nrow(DFB_CallSev_Times), 5000, replace = F)

DFB_CallSev_Times <- DFB_CallSev_Times[index, ]


rm(index)


table(DFB_CallSev_Times$Criticality_Code)


fwrite(NYC_CallSev_Times, "Data/NYC_CallSev_Times.csv", row.names = F)

fwrite(DFB_CallSev_Times, "Data/DFB_CallSev_Times.csv", row.names = F)


DFB_2017 <- DFB_EMS_Data

DFB_2017 <- DFB_EMS_Data[ DFB_2017$Date >= as.POSIXct("2017-01-01") & DFB_2017$Date <=
as.POSIXct("2017-12-31"), ]


sd(DFB_CallSev_Times[DFB_CallSev_Times$Criticality_Code == 'E', ]$TOC_IA_Mins)


##############################################


# Top 5 Final Call Types by Borough FDNY EMS


NYC_TOP_5_CALL <- NYC_EMS_Data %>%
   group_by(BOROUGH, FINAL_CALL_TYPE) %>%
   summarise(count = n()) %>%
   top_n(n = 5, wt = count)


ggplot(NYC_TOP_5_CALL, aes(x = FINAL_CALL_TYPE, y = count)) +
   geom_col() +
   ggtitle(label = "Top 5 Call Categories",
           subtitle = "by Borough") +
   facet_grid(~BOROUGH, scales = "free")+
   theme_economist_white()
```

```
# NYC Response Times

NYC_Response_Times <-
data.frame(NYC_EMS_Data$INCIDENT_DATETIME,NYC_EMS_Data$FINAL_SEVERITY_LEVEL_CODE,
NYC_EMS_Data$INCIDENT_RESPONSE_SECONDS_QY,
NYC_EMS_Data$INCIDENT_TRAVEL_TM_SECONDS_QY)


NYC_Response_Times$NYC_EMS_Data.INCIDENT_RESPONSE_SECONDS_QY <-
round(NYC_Response_Times$NYC_EMS_Data.INCIDENT_RESPONSE_SECONDS_QY / 60, 2)


NYC_Response_Times$NYC_EMS_Data.INCIDENT_TRAVEL_TM_SECONDS_QY <-
round(NYC_Response_Times$NYC_EMS_Data.INCIDENT_TRAVEL_TM_SECONDS_QY / 60, 2)


summary(NYC_Response_Times)


table(NYC_Response_Times$NYC_EMS_Data.FINAL_SEVERITY_LEVEL_CODE)

mean(NYC_Response_Times$NYC_EMS_Data.INCIDENT_RESPONSE_SECONDS_QY)


NYC_Response_Times <- subset(NYC_Response_Times,
NYC_EMS_Data.INCIDENT_RESPONSE_SECONDS_QY <= 360)


write.csv(NYC_Response_Times, "Data/NYC_Response_Times.csv", row.names = F)


########### Analysis Results ###########

# Confusion Matrices

# Random Forest

heldupCMRF


# Random Forest w/ Rose

heldupRoseCM


# Naive Bayes

heldupNBCM


# SVM Linear Grid Tuned

svm_cm
```

```
# SVM Radial Grid Tuned

svm_rbf_cm


# SVM Radial Random Tuned

svm_rndm_cm


index <- sample(1:nrow(NYC_EMS_Data), 80000, replace = F)

NYC_Map <- NYC_EMS_Data[index, ]

write.csv(NYC_Map, "Data/NYC_EMS_MapData.csv", row.names = F)
```

7.17. R Shiny Application Code

```
library(rsconnect)

library(shiny)

library(rlang)

library(shinydashboard)

library(leaflet)

library(data.table)

library(DT)


NYC_EMS_MapData <-
fread("https://raw.githubusercontent.com/CarlOBeirne/Ambulance_Response_FYP/master/NYC_EMS_MapData.csv", header = T, sep = ",")


NYC_EMS_MapData$PopUp <- as.character(paste("Call ID: ",
NYC_EMS_MapData$CAD_INCIDENT_ID, " | Call Reason: ", NYC_EMS_MapData$FINAL_CALL_TYPE, " | Borough: ", NYC_EMS_MapData$BOROUGH))


NYC_EMS_Disp_Code <-
fread("https://raw.githubusercontent.com/CarlOBeirne/Ambulance_Response_FYP/master/NYC_EMS_Disposition_Desc.csv", header = T, sep = ",")

NYC_EMS_Call_Desc <-
fread("https://raw.githubusercontent.com/CarlOBeirne/Ambulance_Response_FYP/master/NYC_EMS_Call_Desc.csv", header = T, sep = ",")
```

```
NYC_EMS_MapData_Sample <- sample(1:nrow(NYC_EMS_MapData), 2000, replace = F)
NYC_EMS_MapData_Sample <- NYC_EMS_MapData[NYC_EMS_MapData_Sample, ]

NYC_EMS_MapData_Table <- NYC_EMS_MapData_Sample[, c(1,2,5,6,17,18,20,30,31)]
colnames(NYC_EMS_MapData_Table)[1] <- "Incident_ID"
colnames(NYC_EMS_MapData_Table)[2] <- "Incident_Date"
colnames(NYC_EMS_MapData_Table)[3] <- "Incident_Reason"
colnames(NYC_EMS_MapData_Table)[4] <- "Incident_Severity"
colnames(NYC_EMS_MapData_Table)[5] <- "Disposition_Code"
colnames(NYC_EMS_MapData_Table)[6] <- "Borough"
colnames(NYC_EMS_MapData_Table)[7] <- "Zipcode"
colnames(NYC_EMS_MapData_Table)[8] <- "Latitude"
colnames(NYC_EMS_MapData_Table)[9] <- "Longitude"


# Define UI for application
ui <- dashboardPage(
   skin = "red",
   # Application title
   dashboardHeader(title = "EMS Calls NYC"),
   dashboardSidebar(
      sidebarMenu(
         menuItem("Interactive Map", tabName = "NYCMap"),
         menuItem("Data Descriptions", tabName = "DataDescs")
      )
   ),
   dashboardBody(
      tabItems(
         tabItem(
            tabName = "NYCMap",
            fluidRow(
               box(
```

```
            width = 12,
            leafletOutput(
              outputId = "nycMap"
            )
          )

        ),
        fluidRow(
          box(
            width = 12,
            dataTableOutput(
              outputId = "SummaryTable"
            )
          )
        )
      ),
      tabItem(
        tabName = "DataDescs",
        fluidRow(
          box(
            titlePanel(
              h2("Disposition Code Descriptions", align = "center")
            ),
            width = 6,
            dataTableOutput(
              outputId = "CallReasonDesc"
            )
          ),
          box(
            titlePanel(
              h2("Call Type Descriptions", align = "center")
            ),
```

```
                width = 6,
                dataTableOutput(
                  outputId = "CallDispDesc"

                )
              )
            )
          )
        )
    )
)

# Server function to produce leaflet map and tables using DT
server <- function(input, output) {
  output$nycMap <- renderLeaflet(
    leaflet() %>%
      addProviderTiles(providers$OpenStreetMap) %>%
      addMarkers( lat = NYC_EMS_MapData_Sample$Latitude,
                  lng = NYC_EMS_MapData_Sample$Longitude,
                  popup = NYC_EMS_MapData_Sample$PopUp,
      )
  )

  output$SummaryTable <- renderDataTable(
    NYC_EMS_MapData_Table
  )

  output$CallReasonDesc <- renderDataTable(
    NYC_EMS_Disp_Code
  )

  output$CallDispDesc <- renderDataTable(
```

```
      NYC_EMS_Call_Desc
    )
}

# Run the application
shinyApp(ui = ui, server = server)
```

7.18. Additional URLS

GitHub Repo - https://bit.ly/3fpKjOJ

R Shiny Dashboard - https://bit.ly/2LeFqd7

Tableau Public Dashboard - https://tabsoft.co/3fv5kHK