2022

Analyzing Weather Impact on Ambulance Response Times

Final Project Report

Assignment 6

GUIDED BY:

PROF. THILANKA MUNASINGHE

SUBMITTED BY:

SHREY JAIN

662046332

JAINS6@RPI.EDU

Table of Contents

1. Abstract and Introduction

In this world of rapid development, having a perfect health directly means leading a productive life. While prevention is better than cure, there are some cases where some serious, unexpected things happen which require immediate action. In those scenarios it is crucial to provide immediate responses to such emergencies as they could turn out be a lifesaver. Another such lifesaving action is to provide the pre-hospital care to any patient present in an ambulance or waiting for one. Therefore, having an accurate arrival time of ambulance could be very helpful.

New York City is the centre of all sorts of emergencies, because it is one of very crowded city in the United States of America. The first dataset is the weather dataset which is of the Manhattan borough in NYC. Emergency medical services data for the same borough is used to analyze trends and ultimately predict response time.

While the EMS data is sufficient to roughly predict the response times for situations that fall in similar categories, however, the aim of the project is also to highlight how co-related weather data is with EMS response data and how much performance gains do we get when we use additional weather attributes for model development.

This project covers process of building a machine learning model that predicts a response time of any ambulance in relation with the weather on that day. It also covers details about the datasets in focus, what are the steps that are to be followed to get the data ready for any model building or evaluation. This report also includes some of the plots which helps identify the relation between variables and finally contains the results and the conclusion drawn out from them.

All the references and resources used for making this project are included in the end of this project report.

2. Data Description and Exploratory Data Analysis

The project works on two datasets:

- EMS Dataset: This dataset is provided by Fire Department of New York City (FDNY). This dataset contains data for many regions hence it has 11863759 rows and 32 columns.
- Weather Dataset: This dataset is provided by Global Historical Climatology Network (GHCN). Since this dataset only contains weather data of Manhattan Borough, hence, it has comparatively very less rows as compared to EMS dataset. It only contains 3288 rows and 34 columns.

Weather on a particular date is used to link with the response times for emergencies that occurred on that date using the EMS datasets. Datasets are joined using INCIDENT_DT column of EMS dataset and DATE column of Weather dataset. The target column to be predicted using various machine learning regression models is INCIDENT_RESPONSE_SECONDS_QY which is present in the EMS dataset.

Before jumping to applying regression models it is very crucial to get data prepared, which means, applying data pre-processing techniques to make data ready for model development. The following data preparation steps (Fig. 2.1) are applied:

- Quality Assessment: High level checks to determine if the data meets the required quality standards.
- Data Cleaning: Fixing incorrect, corrupted entries in the dataset.
- Data Munging: Modified or changing dataset beyond its original state.
- Exploratory Data Analysis (EDA): Performed initial investigation on the data to discover patterns, trends and detect abnormality in data.
- Model Preparation: Applied 4 regression models on the pre-processed dataset.
- Model Evaluation: Evaluated the accuracy and performance of all these models and determined which works best for given set of data.

Fig. 2.1: Data pre-processing steps

On further analyzing the data, there are some key useful functions, which can help us give high help summary of the data variables. Example: *summary*, *fivenum, dim, etc*

- The summary function provides us very crucial information on the dependent variable that I want to predict. It seems there are about 0.3M N/A values in the dataset, just in the response time column itself.

```
> summary(a$incident_response_seconds_qy)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
    0.0   302.0   425.0   536.8   603.0 32392.0  392188 
```

Fig. 2.2: Summary of response time column

```
> dim(ems_data)
[1] 11863759       32
> ems_data_without_na <- na.omit(ems_data)
> dim(ems_data_without_na)
[1] 8130147      32
> 
```

Fig. 2.3: Dimensions of EMS data with and without N/A values

3. Analysis

The following section contains various graphs and plots and how each of these helped in identifying trends and patterns in the data. Fig. 3.1 shows a bar chart highlighting the number of EMS incidents that happened every year ranging from 2008-2016.

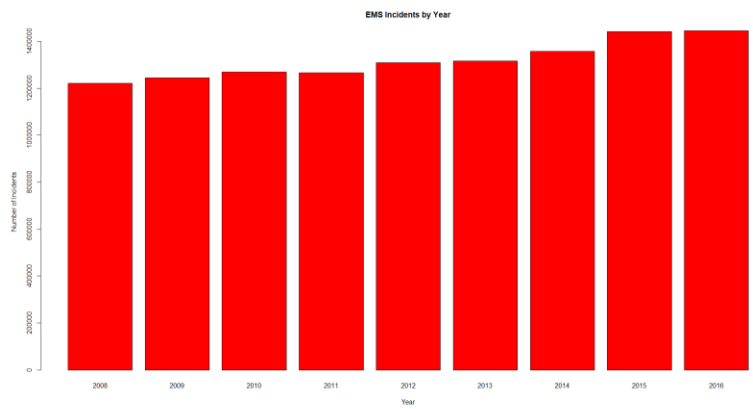


Fig. 3.1: Bar chart highlighting number of incidents by year

It can be inferred from Fig. 3.1 that; number of incidents have a positive co-relation with the year. With each year, the incidents have gradually increase in number.

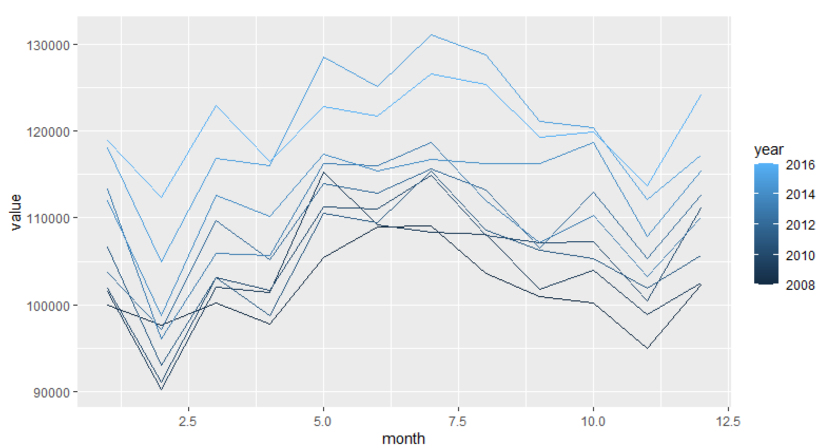


Fig. 3.2: GG-Plot highlighting number of incidents by month

Additionally, from Fig. 3.2, it can be inferred that, usually in the month of February and November, the incidents are comparatively lesser. On the other hand, there is a rise in number of incidents during summers/fall, in the month of June, July, and August.

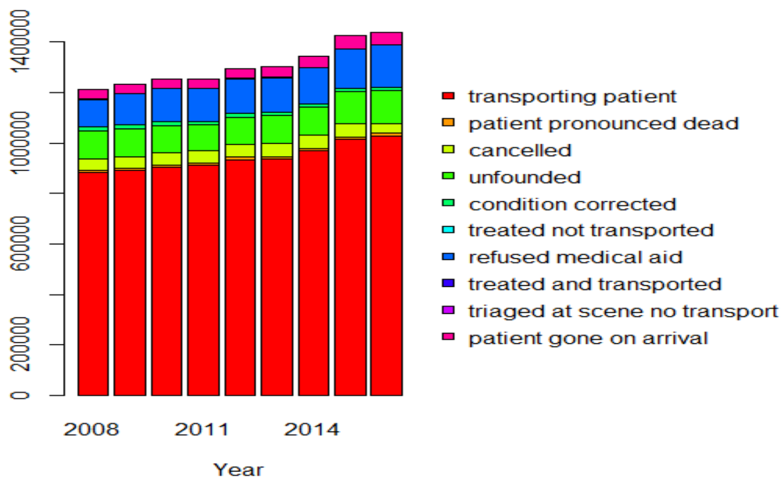


Fig. 3.3: Bar chart highlighting total incidents with time split in different stages

It can also be seen from Fig. 3.3 that, most of the time goes into transporting the patient. As we know that the response time is split across various stages which starts from the time the call was placed to the point where the patient was taken to the doctor. Transporting patients roughly takes about 70% of the total response time in case of an emergency.

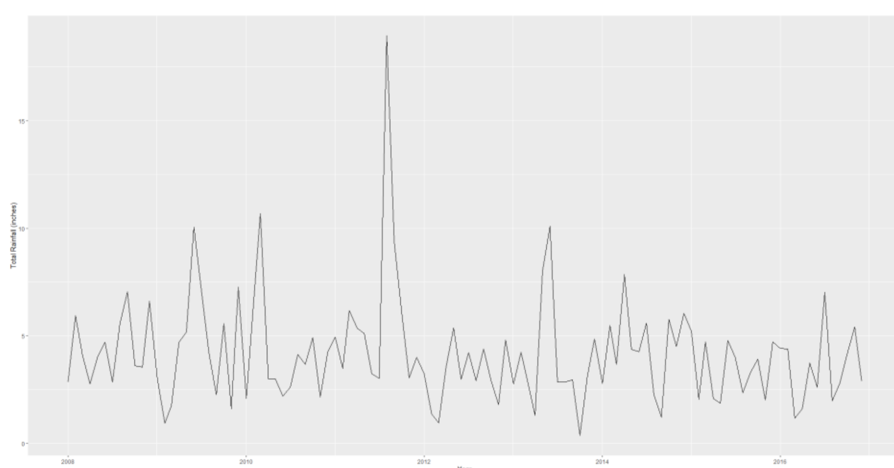


Fig. 3.4: Chart highlighting total rainfall by year

On a separate note, for the Weather data Fig. 3.4, depicts the total rainfall over the year, somewhere in 2011, the rainfall was at an abnormally high peak, which could potentially be an outlier.

➢ Outlier & Cleaning Stats:

- ❖ Outlier detection and removal were done using the IQR method.
- ❖ Removed about 0.22M outliers from the Weather and EMS dataset.
- ❖ Removed about 3.7M rows with N/A values from EMS dataset.
- ❖ Removed 3 columns from Weather dataset as they had majority of N/A values.
- ❖ Converted Date Format & String to numbers.

4. Model Development and Application

Before applying models, the datasets were merged using the data column. The final merged dataset obtained had about 0.36M rows. This number is after all data filtering and cleaning were done and dataset was ready for model building. Models performed significantly better on datasets without outliers and N/A values. Four regression models were applied which are Linear Regression, Support Vector Machine, Random Forest, and finally K-Nearest Neighbours. My additional hypothesis is that Hyper tuning will turn out to be an essential tool optimizing the model parameters and maximizing the performances of all these four models.

➢ Linear Regression

Before performing the linear regression, character literals were converted to integers, N/A values were removed, and attributes that have only one level (constants) were removed. After all the above removal, the final merged dataset had about 360K rows. Fig. 4.1 and Fig. 4.2 depicts histograms of the frequency of the values corresponding to the response times with and without N/A values respectively.

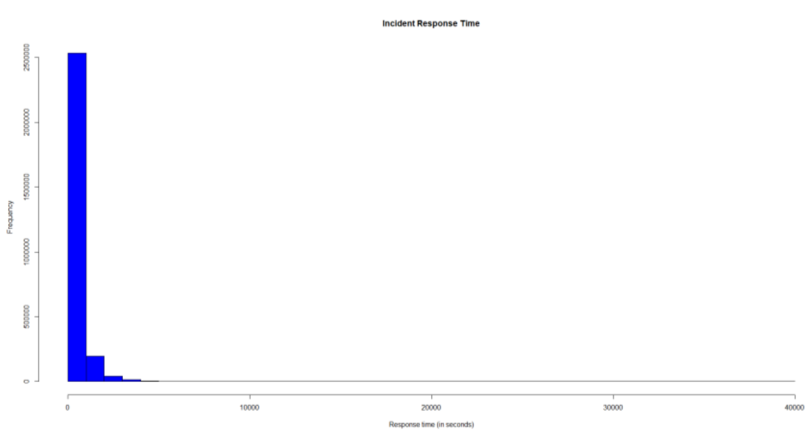


Fig. 4.1: Histogram highlighting frequency of response times in the merged dataset with outliers

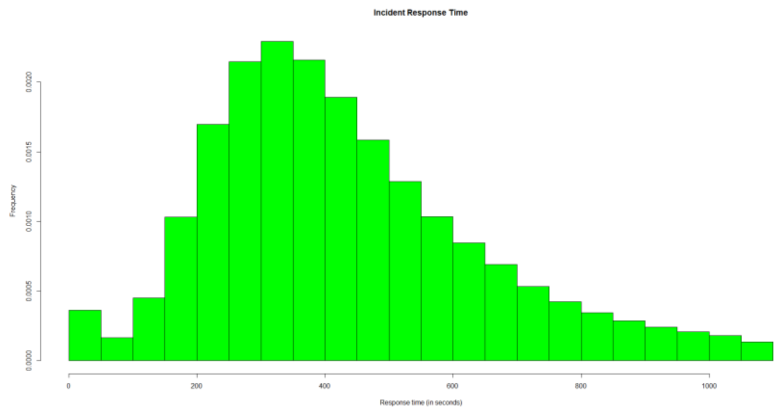


Fig. 4.2: Histogram highlighting frequency of response times in the merged dataset without outliers

This proves one of our initial hypotheses that outlier removal is very important when building any regression model.

Moving on, while applying linear regression the following columns (as represented in Fig. 4.3) were removed as they had constant values and in R if we build a regression model on columns with constant values it gives error.

```
final_dataset_without_outliers <- select(final_dataset_without_outliers, -WT03)
final_dataset_without_outliers <- select(final_dataset_without_outliers, -WT04)
final_dataset_without_outliers <- select(final_dataset_without_outliers, -valid_incident_rspns_time_indc)
final_dataset_without_outliers <- select(final_dataset_without_outliers, -valid_dispatch_rspns_time_indc)
final_dataset_without_outliers <- select(final_dataset_without_outliers, -WT14)
```

Fig. 4.3: Five columns that had constant values and were removed before building regression model

The final RMSE value of linear regression is depicted in Fig. 4.4. This value is very low, meaning that linear regression performed well.

```
> sqrt(mean(model_linear$residuals^2))
[1] 5.846136e-12
```

Fig. 4.4: RMSE value of Linear Regression

The linear regression model was then built using the lm.fit() function. Fig. 4.5 on the next page describes the output of the summary of the linear regression model.

```
Call:
lm(formula = incident_response_seconds_qy ~ ., data = final_dataset_without_outliers)

Residuals:
       Min         1Q     Median         3Q        Max 
-1.254e-10 -1.400e-13 -1.000e-14  1.000e-13  1.635e-09 

Coefficients:
                               Estimate Std. Error    t value Pr(>|t|)    
(Intercept)                   9.153e-12  4.371e-12  2.094e+00 0.036264 *  
initial_severity_level_code   1.613e-12  3.222e-14  5.006e+01  < 2e-16 ***
final_severity_level_code    -1.135e-13  3.211e-14 -3.533e+00 0.000411 ***
dispatch_response_seconds_qy  1.000e+00  3.446e-16  2.901e+15  < 2e-16 ***
incident_travel_tm_seconds_qy 1.000e+00  9.745e-17  1.026e+16  < 2e-16 ***
held_indicator1              -1.730e-13  1.201e-13 -1.440e+00 0.149954    
incident_disposition_code    -3.681e-16  4.018e-15 -9.200e-02 0.927003    
zipcode10023                 -1.204e-13  2.591e-13 -4.650e-01 0.642190    
zipcode10024                 -1.597e-13  2.638e-13 -6.060e-01 0.544841    
zipcode10025                 -1.932e-13  2.791e-13 -6.920e-01 0.488900    
policeprecinct                2.744e-15  2.192e-14  1.250e-01 0.900374    
citycouncildistrict           9.501e-14  2.794e-14  3.400e+00 0.000673 ***
communitydistrict            -5.421e-14  4.284e-14 -1.265e+00 0.205753    
communityschooldistrict      -6.830e-14  2.415e-13 -2.830e-01 0.777342    
congressionaldistrict        -2.479e-14  1.850e-14 -1.340e+00 0.180232    
reopen_indicator1            -1.921e-14  8.580e-13 -2.200e-02 0.982136    
special_event_indicator1     -3.013e-13  2.080e-12 -1.450e-01 0.884852    
standby_indicator1           -1.756e-13  3.620e-13 -4.850e-01 0.627572    
transfer_indicator1           1.055e-15  5.850e-12  0.000e+00 0.999856    
initial_call_type            -1.691e-16  9.466e-16 -1.790e-01 0.858230    
final_call_type              -2.339e-16  9.080e-16 -2.580e-01 0.796701    
month.x02                    -3.289e-13  9.320e-14 -3.529e+00 0.000418 ***
month.x03                    -3.819e-13  9.577e-14 -3.988e+00 6.67e-05 ***
month.x04                    -4.716e-13  1.112e-13 -4.240e+00 2.24e-05 ***
month.x05                    -5.075e-13  1.220e-13 -4.158e+00 3.21e-05 ***
month.x06                    -5.798e-13  1.392e-13 -4.165e+00 3.12e-05 ***
month.x07                    -5.949e-13  1.512e-13 -3.934e+00 8.37e-05 ***
month.x08                    -5.833e-13  1.469e-13 -3.971e+00 7.17e-05 ***
month.x09                    -5.495e-13  1.353e-13 -4.061e+00 4.89e-05 ***
month.x10                    -4.796e-13  1.164e-13 -4.119e+00 3.81e-05 ***
month.x11                    -4.179e-13  1.077e-13 -3.880e+00 0.000105 ***
month.x12                    -3.017e-13  9.402e-14 -3.209e+00 0.001333 ** 
AWND                         -1.010e-15  1.411e-14 -7.200e-02 0.942919    
FMTM                         -1.343e-16  4.161e-17 -3.227e+00 0.001252 ** 
PGTM                          1.536e-16  4.004e-17  3.836e+00 0.000125 ***
PRCP                         -4.540e-14  5.677e-14 -8.000e-01 0.423872    
SNOW                          6.191e-15  3.298e-14  1.880e-01 0.851090    
SNWD                         -2.191e-15  8.798e-15 -2.490e-01 0.803342    
TMAX                         -7.462e-16  4.010e-15 -1.860e-01 0.852394    
TMIN                          6.558e-15  4.898e-15  1.339e+00 0.180580    
WDF2                         -8.042e-16  3.494e-16 -2.302e+00 0.021356 *  
WDF5                          6.327e-16  3.568e-16  1.773e+00 0.076187 .  
WSF2                         -1.939e-14  1.476e-14 -1.314e+00 0.188835    
WSF5                          1.278e-14  8.604e-15  1.485e+00 0.137508    
WT01                          8.975e-14  1.086e-13  8.260e-01 0.408756    
WT02                         -3.023e-14  1.142e-13 -2.650e-01 0.791278    
WT05                         -2.266e-13  6.437e-14 -3.520e+00 0.000431 ***
WT06                          1.435e-14  4.431e-13  3.200e-02 0.974157    
WT07                         -1.859e-14  8.649e-14 -2.150e-01 0.829807    
WT08                         -1.072e-13  5.923e-14 -1.811e+00 0.070185 .  
WT09                          9.257e-14  2.128e-13  4.350e-01 0.663557    
WT11                         -9.827e-14  6.405e-13 -1.530e-01 0.878061    
WT13                          5.908e-14  1.019e-13  5.800e-01 0.561952    
WT16                          1.460e-13  6.320e-14  2.310e+00 0.020908 *  
WT17                         -5.647e-14  2.120e-13 -2.660e-01 0.789920    
WT18                         -1.718e-13  9.972e-14 -1.723e+00 0.084823 .  
WT19                         -1.359e-13  1.272e-13 -1.069e+00 0.285150    
WT22                         -6.672e-14  2.308e-13 -2.890e-01 0.772478    
month.y                      -1.340e-16  5.028e-17 -2.665e+00 0.007689 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Fig. 4.4: Summary of the Linear Regression Model

➢ Support Vector Regression

In case of support vector machine, the same dataset without outliers was used which was used for Linear Regression. Before applying models, the datasets were merged using the data column. Models performed significantly better on datasets without outliers and N/A values. My hypothesis is again that Hyper tuning will turn out to be an essential tool optimizing the model parameters and maximizing the performances of Support Vector Regression.

The SVR was built on a subset of the entire data since it was taking exceptionally long to build the model on the entire dataset. The following are the lines of code used for this model (Fig. 4.5).

```
#Regression with SVM
set.seed(7)
train <- sample(dim(final_no_outlier)[1], 1800)
modelsvm = svm(incident_response_seconds_qy~SNOW+PRCP, subset = train, data = final_no_outlier)

#Predict using SVM regression
predYsvm = predict(modelsvm, final_no_outlier)
```

Fig. 4.5: SVM model building code

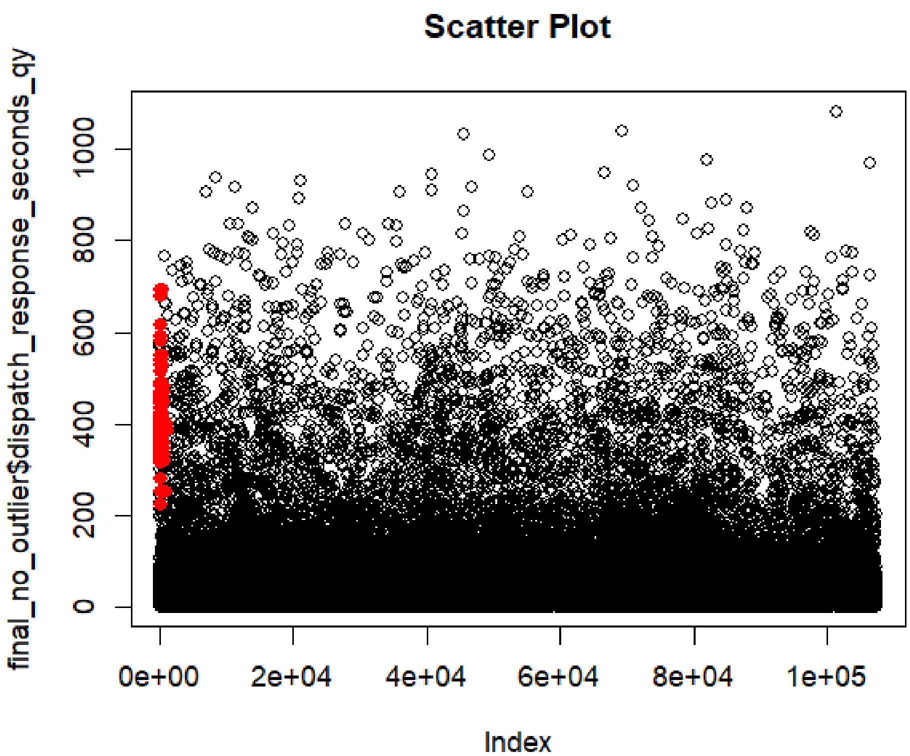


Fig. 4.6: Scatter plot highlighting predicted values vs actual values for Precipitation

Fig. 4.6 depicts the scatter plot for precipitation column, the performance Is really bad for the SVM model, which is primarily because of the less size of the training dataset. Fig. 4.7 gives the summary of the SVM model. After building linear regression and SVM, I can see that linear regression not only took less time to build but also had very less RMSE value and beats the performance of SVR model. The RMSE value of SVM came out to be around 300.

```
Call:
svm(formula = incident_response_seconds_qy ~ SNOW + PRCP, data = final_no_outlier, subset = trair


Parameters:
   SVM-Type:  eps-regression
 SVM-Kernel:  radial
       cost:  1
      gamma:  0.5
    epsilon:  0.1


Number of Support Vectors:  1632
```

Fig. 4.7: SVM Model Summary

➢ Random Forest

Random Forest Regressor is the third model of choice for analyzIng the weather impact. Fig. 4.8 depicts the summary of the Random Forest model.

```
Call:
 randomForest(formula = incident_response_seconds_qy ~ SNOW +      PRCP, data = final_no_outlier, ntree = 1000, keep.forest = FALSE,
rtance = TRUE)
               Type of random forest: regression
                     Number of trees: 1000
No. of variables tried at each split: 1

          Mean of squared residuals: 43022.23
                    % Var explained: 0.07
```

Fig. 4.8: Random Forest Summary

The performance of Random Forest is judged based on the mean squared residuals which came out to be very high. Due to the limitations of the hardware on which I was working on, I was not able to build a random forest model with all the important attributes. However, I used some of the attributes which had the highest weights in the linear regression model. Next Fig. 4.9 depicts a GG-plot showing the weightage of SNOW and PRCP column in the model.

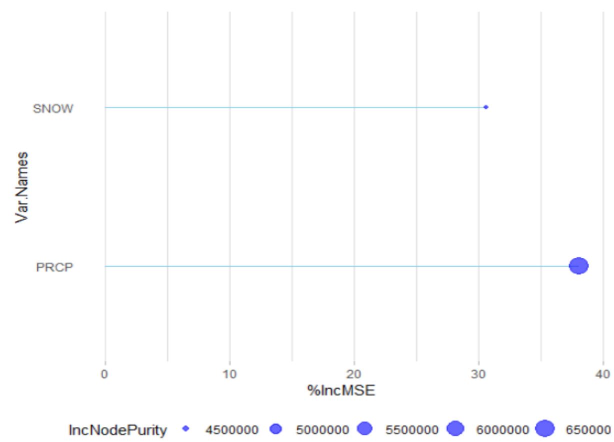
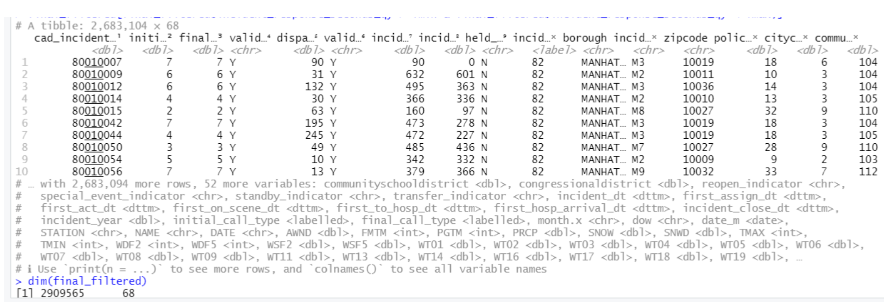


Fig. 4.9: Importance plot for Random Forest

We can see that the scatter plot depicts especially for precipitation column that the performance Is bad for the SVM model, which is primarily because of the less size of the training dataset. Fig. 4.9 gives the summary of the Random Forest model. After building linear regression and Random Forest, I can see that linear regression not only took less time.

➢ KNN

The fourth and final model for my project is the K-nearest neighbour regressor. While KNN is usually suited for classification problems, my aim for choosing this was to validate how well KNN performs for regression problems. While I was not even aware at first that KNN could be used as a regressor too. Therefore, it my fourth choice of model for this problem. So far, only Linear Regression performed well will very low RMSE value and a high accuracy. I assume KNN would also not perform that well.



```
# A tibble: 2,683,104 x 68
   cad_incident…¹ initi…² final…³ valid…⁴ dispa…⁵ valid…⁶ incid…⁷ incid…⁸ held_…⁹ incid…˟ borough incid…˟ zipcode polic…˟ cityc…˟ commu…˟
            <dbl>   <dbl>   <dbl> <chr>     <dbl> <chr>     <dbl>   <dbl> <chr>   <label> <chr>   <chr>   <chr>     <dbl>   <dbl>   <dbl>
 1       80010007       7       7 Y            90 Y            90       0 N            82 MANHAT… M3      10019        18       6     104
 2       80010009       6       6 Y            31 Y           632     601 N            82 MANHAT… M2      10011        10       3     104
 3       80010012       6       6 Y           132 Y           495     363 N            82 MANHAT… M3      10036        14       3     104
 4       80010014       4       4 Y            30 Y           366     336 N            82 MANHAT… M2      10010        13       3     105
 5       80010015       2       2 Y            63 Y           160      97 N            82 MANHAT… M8      10027        32       9     110
 6       80010042       7       7 Y           195 Y           473     278 N            82 MANHAT… M3      10019        18       3     104
 7       80010044       4       4 Y           245 Y           472     227 N            82 MANHAT… M3      10019        18       3     105
 8       80010050       3       3 Y            49 Y           485     436 N            82 MANHAT… M7      10027        28       9     110
 9       80010054       5       5 Y            10 Y           342     332 N            82 MANHAT… M2      10009         9       2     103
10       80010056       7       7 Y            13 Y           379     366 N            82 MANHAT… M9      10032        33       7     112
# … with 2,683,094 more rows, 52 more variables: communityschooldistrict <dbl>, congressionaldistrict <dbl>, reopen_indicator <chr>,
#   special_event_indicator <chr>, standby_indicator <chr>, transfer_indicator <chr>, incident_dt <dttm>, first_assign_dt <dttm>,
#   first_act_dt <dttm>, first_on_scene_dt <dttm>, first_to_hosp_dt <dttm>, first_hosp_arrival_dt <dttm>, incident_close_dt <dttm>,
#   incident_year <dbl>, initial_call_type <labelled>, final_call_type <labelled>, month.x <chr>, dow <chr>, date_m <date>,
#   STATION <chr>, NAME <chr>, DATE <chr>, AWND <dbl>, FMTM <int>, PGTM <int>, PRCP <dbl>, SNOW <dbl>, SNWD <dbl>, TMAX <int>,
#   TMIN <int>, WDF2 <int>, WDF5 <int>, WSF2 <dbl>, WSF5 <dbl>, WT01 <dbl>, WT02 <dbl>, WT03 <dbl>, WT04 <dbl>, WT05 <dbl>, WT06 <dbl>,
#   WT07 <dbl>, WT08 <dbl>, WT09 <dbl>, WT11 <dbl>, WT13 <dbl>, WT14 <dbl>, WT16 <dbl>, WT17 <dbl>, WT18 <dbl>, WT19 <dbl>, …
# ℹ Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
> dim(final_filtered)
[1] 2909565      68
```

Fig. 4.10: KNN Regression Dataset Summary

For KNN, I split the data in testing and training with a 30-70 split. There were many columns in the dataset which were not numeric and were giving me error while building the model. After performing the pre-processing, I did not get those errors again. Also, outlier removal helped with the performance of the model.

```
 incident_response_seconds_qy      AWND               PRCP               SNOW              SNWD               TMAX
 Min.   :-0.85848             Min.   :-2.303723   Min.   :-0.354295   Min.   :-0.11013   Min.   :-0.227383   Min.   :-2.6654574
 1st Qu.:-0.37550             1st Qu.:-0.743416   1st Qu.:-0.354295   1st Qu.:-0.11013   1st Qu.:-0.227383   1st Qu.:-0.8121132
 Median :-0.17879             Median :-0.193203   Median :-0.354295   Median :-0.11013   Median :-0.227383   Median : 0.0600488
 Mean   :-0.00005             Mean   :-0.000018   Mean   :-0.000065   Mean   : 0.00002   Mean   :-0.000032   Mean   :-0.0000071
 3rd Qu.: 0.10588             3rd Qu.: 0.541784   3rd Qu.:-0.223034   3rd Qu.:-0.11013   3rd Qu.:-0.227383   3rd Qu.: 0.8777007
 Max.   :50.94527             Max.   : 7.066329   Max.   :14.898174   Max.   :31.63250   Max.   : 9.556725   Max.   : 2.1859437
      TMIN
 Min.   :-3.0164941
 1st Qu.:-0.8022275
 Median : 0.0356031
 Mean   :-0.0000303
 3rd Qu.: 0.8734337
 Max.   : 2.0703345
```

5. Conclusion and Discussion

To conclude, the primary aim of the project was to apply the concepts of Data Analytics to identify if Weather has any impact on the EMS response times. The following conclusion can be drawn from the work done as part of this project.

- Weather data and EMS response times are not entirely co-related, meaning it does not assert the initial hypothesis I have made that weather has impact on the ambulance response time.
- As part of preprocessing:
  - Removed about 0.22M outliers from the Weather and EMS dataset.
  - Removed about 3.7M rows with N/A values from EMS dataset.
  - Removed 3 columns from Weather dataset as they had majority of N/A values.
  - Converted Date Format & String to numbers.
- All the four regression models namely, Linear Regression, SVM, Random Forest, and KNN were built on the same dataset and evaluated basis their accuracy, root mean square error, deviation from the actual expected result, and other evaluation techniques.
- Linear Regression had the highest accuracy and least mean square error amongst other 4 models which means that it is best suited to predict ambulance response times for our problem.
- Hyper tuning turned out to be an essential tool to optimize model parameters and maximize the performances of all the four models.
- The aim of the project was to analyze the impact of Weather on EMS response times using the concepts taught by Prof. Thilanka Munasinghe in his Data Analytics class.

6. References

- R Basics: https://towardsdatascience.com/r-basics-everything-you-need-to-know-to-get-started-with-r-10c8e566d7b3
- GGplot in R: https://www.youtube.com/watch?v=E_KCbC6sBv0
- Random Forest in R: https://hackernoon.com/random-forest-regression-in-r-code-and-interpretation
- KNN in R: https://www.datatechnotes.com/2020/10/knn-regresion-example-in-r.html
- Data munging in R: https://www.toptal.com/r/boost-your-data-munging-with-r
- InterQuartile Range in R: https://www.datatechnotes.com/2020/10/knn-regresion-example-in-r.html