Seattle 911 Police Department Incident Response Data Analysis Final Report

Khaliun Bat-Ochir

May 25th, 2021

Contents

1 Introduction

This is the Seattle 911 Police Department Incident Response Data Analysis Final Report for Capstone Course from Data Science Professional Certificate Program offered by HarvardX and online initiative by Harvard University. Report is prepared by Khaliun Bat-Ochir.

Final Report includes methodology and results of the analysis. Also, this report includes project environment description used for building and running the code.

1.1 Project Goal

This is a Machine Learning project. Main goal of the project consists in training an algorithm that predicts Event Clearance Code of **Seattle Police Department 911 Incident Response** data set.

Sub goal of the project consists in:

- Analyzing **Seattle Police Department 911 Incident Response** data to obtain the best features from the data set and improve interpretability
- Choosing the best model for training
- Obtaining best tune to achieve highest accuracy, specificity, sensitivity values in the confusion matrix

1.2 Project Data and preprocessing for download

Project data is based on **Seattle Police Department 911 Incident Response** cleaned data set from *Kaggle*[1]. The original data file was downloaded and processed using *Seattle911DataPrepScript.sh*[2] bash script to pre-process the data to obtain intended amount of lines of data and also replace the header. Resulting data file was then uploaded to *Github Repository*[3] and is available for download.

1.3 Project Files

Project data and script files and folders are uploaded *Seattle911PDProject Github Repository*[4]). Uploaded Files Include:

- Data File - "data/SeattlePD911IR_80_MB.zip" available for download and used for the project script
- RMarkdown Report File - "Seattle911PDProject_FinalReport.Rmd" which is the current report file
- R Script File - "Seattle911PDProject_Script.R" that includes codes and comments for model fitting and training
- PDF Report - "Seattle911PDProject_FinalReport.pdf"

Github repository also holds additional files and folders used for the project and knitting the report.

[1] https://www.kaggle.com/sohier/seattle-police-department-911-incident-response
[2] https://github.com/khaliunb/Seattle911PDProject/blob/fe35003ed71cdae22966a168b7728724403d3a7c/Seattle911DataPrepScript.sh
[3] https://github.com/khaliunb/Seattle911PDProject/tree/main/data
[4] https://github.com/khaliunb/Seattle911PDProject.git

1.4 Project Environment

Codes for the project were built and tested using:

- R version 3.6 and
- RStudio Version 1.3.1073
- Linux Ubuntu 20.04

2 Method

2.1 Pre-processing the data

Code downloads and populates the data file into **S911IR** data set.

Additional features related to timing had been mutated into the data set we are using. Here is the list descriptions of data headers.

colDesc	colName
CAD CDW ID	CAD_CDW_ID
CAD Event Number	CAD_EN
General Offense Number	GON
Event Clearance Code	ECC
Event Clearance Description	ECD
Event Clearance SubGroup	ECSG
Event Clearance Group	ECG
Event Clearance Date	ECDt
Hundred Block Location	HBL
District/Sector	Dist_Sec
Zone/Beat	Zone_Beat
Census Tract	Census_Tract
Longitude	Longitude
Latitude	Latitude
Incident Location	ILoc
Initial Type Description	ITDesc
Initial Type Subgroup	ITSG
Initial Type Group	ITG
At Scene Time	ASTm
Event Clearance Date Converted to Full data time format	EC_DateTime
At Scene Time Converted to Full data time format	AS_DateTime
Timespan in minutes between At Scene Time and Event Clearance Date Time	AS_TimeSpan

Basic overview of data, revealed that we probably have incomplete records before June of 2010. Therefore we are trimming the original data to records between 1st of July, 2010 and 31st December, 2017.

Overview of data revealed data record number drop between 2013 and 2014 year. The data is probably missing. Therefore, We are removing this part of data from the data set.

Also, for large amount of data for Initial Type description/subgroup/group and At Scene time records are missing.

We have chosen the features for our prediction. NAs are not permitted in random forest predictors. But the ITDesc, ITSG and ITG field NAs are valuable. We have to replace NAs within those columns by further processing the *S911IR* data set. Therefore, we are correcting the data by changing ITDesc, ITSG, ITG NAs to "UNKNOWN" character values.

Here is how ITDesc and other initial type features' NAs looked like:

```
S911IR%>%filter(is.na(ITDesc)|is.na(ITSG)|is.na(ITG))%>%
  group_by(ITDesc,ITSG,ITG)%>%
  summarise(n=n())%>%
  select(ITDesc,ITSG,ITG,n)%>%
  knitr::kable()
```

ITDesc	ITSG	ITG	n
NA	NA	NA	144483

And here is how ITDesc and other initial type features' NAs looked like after NA Replacement with "UNKNOWN" character string:

```
#We are using this code to replace the fields
S911IR<-S911IR%>%replace_na(list(ITDesc="UNKNOWN",ITSG="UNKNOWN",ITG="UNKNOWN"))

S911IR%>%filter(is.na(ITDesc)|is.na(ITSG)|is.na(ITG))%>%
  group_by(ITDesc,ITSG,ITG)%>%
  summarise(n=n())%>%
  select(ITDesc,ITSG,ITG,n)%>%
  knitr::kable()
```

ITDesc	ITSG	ITG	n
UNKNOWN	UNKNOWN	UNKNOWN	144483

Then, we trim the data columns by selecting all the features used for predictions and remove all NAs from the resulting data set by using this code:

```
S911IR<-S911IR%>%drop_na()
```

2.2 Data Set for training

We will take a glimpse at **S911IR** data set:

```
## Rows: 353,415
## Columns: 15
```

```
## $ CAD_CDW_ID <dbl> 765130, 2190614, 1046025, 391547, 945147, 108938, 838723...
## $ ECC        <chr> "280", "245", "245", "244", "161", "065", "200", "244", ...
## $ ECD        <chr> "SUSPICIOUS PERSON", "DISTURBANCE, OTHER", "DISTURBANCE,...
## $ ILoc       <chr> "(47.467179311, -122.318224863)", "(47.660794519, -122.3...
## $ Longitude  <dbl> -122.3182, -122.3651, -122.3498, -122.3746, -122.3163, -...
## $ Latitude   <dbl> 47.46718, 47.66079, 47.61646, 47.67781, 47.66568, 47.605...
## $ ITDesc     <chr> "UNKNOWN", "DUI - DRIVING UNDER INFLUENCE", "DISTURBANCE...
## $ EC_Year    <dbl> 2012, 2014, 2015, 2011, 2012, 2010, 2012, 2010, 2016, 20...
## $ EC_Quarter <int> 1, 4, 1, 3, 4, 4, 3, 3, 1, 3, 3, 1, 2, 2, 1, 2, 2, 3, 1,...
## $ EC_Month   <dbl> 3, 11, 3, 7, 10, 11, 7, 9, 3, 8, 8, 2, 4, 4, 2, 6, 5, 7,...
## $ EC_Day     <int> 30, 3, 21, 15, 28, 2, 1, 26, 16, 24, 10, 28, 25, 8, 29, ...
## $ EC_Weekday <dbl> 6, 2, 7, 6, 1, 3, 1, 1, 4, 4, 2, 3, 2, 1, 2, 7, 2, 2, 7,...
## $ ECDn       <int> 1, 2, 2, 3, 4, 5, 6, 3, 2, 7, 8, 9, 10, 9, 11, 12, 7, 7,...
## $ ITDescN    <int> 1, 2, 3, 1, 1, 1, 1, 1, 3, 1, 4, 5, 6, 1, 7, 8, 9, 9, 10...
## $ ILocN      <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 1...
```

Resulting data set has *353415 rows* and *15 columns*.

For training purposes, data is divided into **train_set** and **test_set**:

- **train_set** - 80% of the complete **Seattle Police Department 911 Incident Response** data set
- **test_set** - 20% of the complete **Seattle Police Department 911 Incident Response** data set

Furthermore, **train_set** data set had been matched with **test_set** data set and all recurring data had been removed from **test_set** data set.

Summary of **train_set** data set:

```
##    CAD_CDW_ID          ECC                ECD                ILoc
##  Min.   :  15758   Length:79950       Length:79950       Length:79950
##  1st Qu.: 711870   Class :character   Class :character   Class :character
##  Median :1270954   Mode  :character   Mode  :character   Mode  :character
##  Mean   :1246268
##  3rd Qu.:1845570
##  Max.   :2258718
##    Longitude         Latitude        ITDesc             EC_Year
##  Min.   :-122.4   Min.   :47.45   Length:79950       Min.   :2010
##  1st Qu.:-122.3   1st Qu.:47.59   Class :character   1st Qu.:2011
##  Median :-122.3   Median :47.61   Mode  :character   Median :2014
##  Mean   :-122.3   Mean   :47.62                      Mean   :2014
##  3rd Qu.:-122.3   3rd Qu.:47.66                      3rd Qu.:2016
##  Max.   :-122.2   Max.   :47.78                      Max.   :2017
```

```
##    EC_Quarter      EC_Month          EC_Day        EC_Weekday
##  Min.   :1.00   Min.   : 1.000   Min.   : 1.00   Min.   :1.000
##  1st Qu.:2.00   1st Qu.: 4.000   1st Qu.: 8.00   1st Qu.:2.000
##  Median :3.00   Median : 7.000   Median :16.00   Median :4.000
##  Mean   :2.59   Mean   : 6.719   Mean   :15.63   Mean   :4.064
##  3rd Qu.:3.00   3rd Qu.: 9.000   3rd Qu.:23.00   3rd Qu.:6.000
##  Max.   :4.00   Max.   :12.000   Max.   :31.00   Max.   :7.000
##       ECDn           ITDescN           ILocN
##  Min.   :  1.00   Min.   :  1.00   Min.   :    1
##  1st Qu.:  4.00   1st Qu.:  1.00   1st Qu.: 3105
##  Median : 12.00   Median :  8.00   Median :10565
##  Mean   : 18.66   Mean   : 16.96   Mean   :17360
##  3rd Qu.: 26.00   3rd Qu.: 23.00   3rd Qu.:26864
##  Max.   :113.00   Max.   :164.00   Max.   :71071
```

Summary of **test_set** data set:

```
##    CAD_CDW_ID          ECC                ECD                ILoc
##  Min.   :  15762   Length:20050       Length:20050       Length:20050
##  1st Qu.: 714260   Class :character   Class :character   Class :character
##  Median :1277068   Mode  :character   Mode  :character   Mode  :character
##  Mean   :1248664
##  3rd Qu.:1844685
##  Max.   :2258698
##    Longitude         Latitude        ITDesc             EC_Year
##  Min.   :-122.4   Min.   :47.46   Length:20050       Min.   :2010
##  1st Qu.:-122.3   1st Qu.:47.59   Class :character   1st Qu.:2011
##  Median :-122.3   Median :47.61   Mode  :character   Median :2014
##  Mean   :-122.3   Mean   :47.62                      Mean   :2014
##  3rd Qu.:-122.3   3rd Qu.:47.66                      3rd Qu.:2016
##  Max.   :-122.2   Max.   :47.78                      Max.   :2017
##    EC_Quarter       EC_Month          EC_Day        EC_Weekday
##  Min.   :1.000   Min.   : 1.000   Min.   : 1.00   Min.   :1.00
##  1st Qu.:2.000   1st Qu.: 4.000   1st Qu.: 8.00   1st Qu.:2.00
##  Median :3.000   Median : 7.000   Median :16.00   Median :4.00
##  Mean   :2.595   Mean   : 6.727   Mean   :15.62   Mean   :4.08
##  3rd Qu.:3.000   3rd Qu.: 9.000   3rd Qu.:23.00   3rd Qu.:6.00
##  Max.   :4.000   Max.   :12.000   Max.   :31.00   Max.   :7.00
##       ECDn           ITDescN           ILocN
##  Min.   :  1.00   Min.   :  1.00   Min.   :    1
```

```
##  1st Qu.:  4.00   1st Qu.:  1.00   1st Qu.: 2965
##  Median : 12.00   Median :  8.00   Median :10394
##  Mean   : 18.79   Mean   : 16.88   Mean   :17368
##  3rd Qu.: 26.00   3rd Qu.: 22.00   3rd Qu.:26898
##  Max.   :115.00   Max.   :166.00   Max.   :71025
```

2.3 Questions Posed

Before we handle any of the data, we have three questions to answer:

- What Are We Predicting?
- What Features Will We Use for Predictions?
- How Are We Gonna Predict?

From my perspective, the linear model is not applicable here. The best way to approach this problem is to cluster and train the data with knn algorithm and see what happens. However, we need to develop a way to make it more interpretable.

By this point, we are looking at more definite description of our project's problem. We need to experiment with data, to test out different assumptions. General questions we needed to answer:

1. What if we only group the full data features into subsets of groups?
2. How many groups the data should the features be divided into?
3. Should we consider regularization?
4. What happens if we feed the train the clustered group features random forest algorithm, and what is the best tune?
5. How much accuracy can we get with predicting the event clearance?

From the data set, we assume that the features can be divided into location, timing, and situation groups. From here, we answer to more detailed questions:

- Which are the event clearance groups/subgroup/description that occur the most?
- Which are the most frequently occurring event clearance group and incident type group pair?/Is there relation between incident type group/subgroup/description and event clearance group/subgroup/description?
- Is there relation between location and event clearance description/subgroup/description?
  - Is there certain prevalence in overall number of event clearance description/subgroup/description in certain location?
- Is there relation between timing and event clearance description/subgroup/description?
  - Was there increase/decrease in event clearance description/subgroup/description overall occurrence over time?
  - During which season event clearance description/subgroup/description overall occurrence increases/decreases?
  - On which day of the week event clearance description/subgroup/description overall occurrence increases/decreases?
  - During which hour of the day event clearance description/subgroup/description overall occurrence increases/decreases?

These questions clarify the answer which of the event clearance group subgroup/description features we are predicting.

Also we are considering recurrence of certain groups of events clearance and incident types:

- Is there relation between number of recurrence of certain incident type group/subgroup/description in data and event clearance group/subgroup/description?
- Is there relation between number of recurrence of certain event clearance group/subgroup/description in data and timing of the incident type group/subgroup/description?
- Is there relation between number of recurrence of certain event clearance group/subgroup/description in data and location of the incident type group/subgroup/description?
- Is there certain pattern of increase/decrease recurrence of certain event clearance group/subgroup/description in data in relation to location?
  - Is there certain prevalence in number of certain event clearance description/subgroup/description in certain location?
- Is there certain pattern of increase/decrease recurrence of certain event clearance group/subgroup/description in data in relation to timing?
  - Was there increase/decrease in certain event clearance description/subgroup/description occurrence over time?
  - During which season certain event clearance description/subgroup/description occurrence increases/decreases?
  - On which day of the week certain event clearance description/subgroup/description occurrence increases/decreases?
  - During which hour of the day certain event clearance description/subgroup/description occurrence increases/decreases?

3 Analysis and Results

Note: Results section contains plots and summary results of the analysis process. Plots had been saved and are stored in *images* folder.

This section demonstrates the results for **S911IR** Project data before we pre-processed (removed any of data or cleaned from NAs).

Due to the fact data was likely missing, it is probably best to compare the analysis results for both Project data we are using and the complete 380MB data downloaded from Kaggle. This will help is see if the data was properly sampled, and also answer the question "What happens if we use the incomplete data set?". Plots have been aligned side by side for better comparison.

Lists we represent here in the analysis results have been prepared using the Project data we are using.

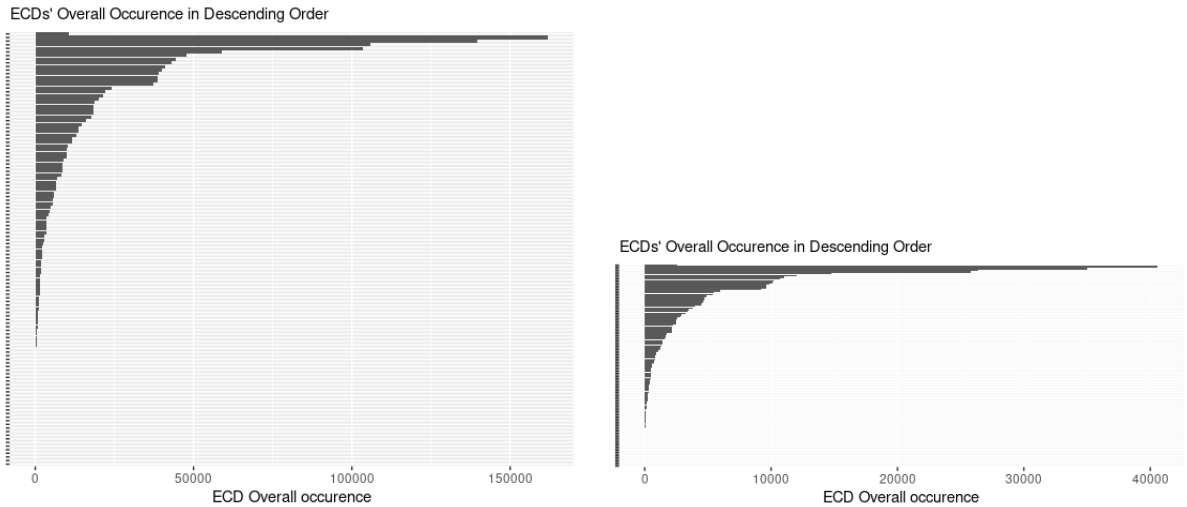
3.1 Questions answered

3.1.1 Basic Question: What are we predicting?

We are predicting **ECD** (Event Clearance Description) value. From data, we see the incident description and the event clearance descriptions rarely match for events viewed as "criminal" and "suspicious". We also see that traffic incidents have clearer descriptions. This may relate to the fact that people's eyes recognize "suspicious" may be viewed differently depending on circumstances. We may view it as additional motivation for predicting the event clearance prior to dispatching police department.

3.1.2 Question: What is the overall picture of overall occurence of event clearance description? Also, do some ECDs prevail over others on daily basis?

We will view overall occurence plots side by side to get a clear idea.

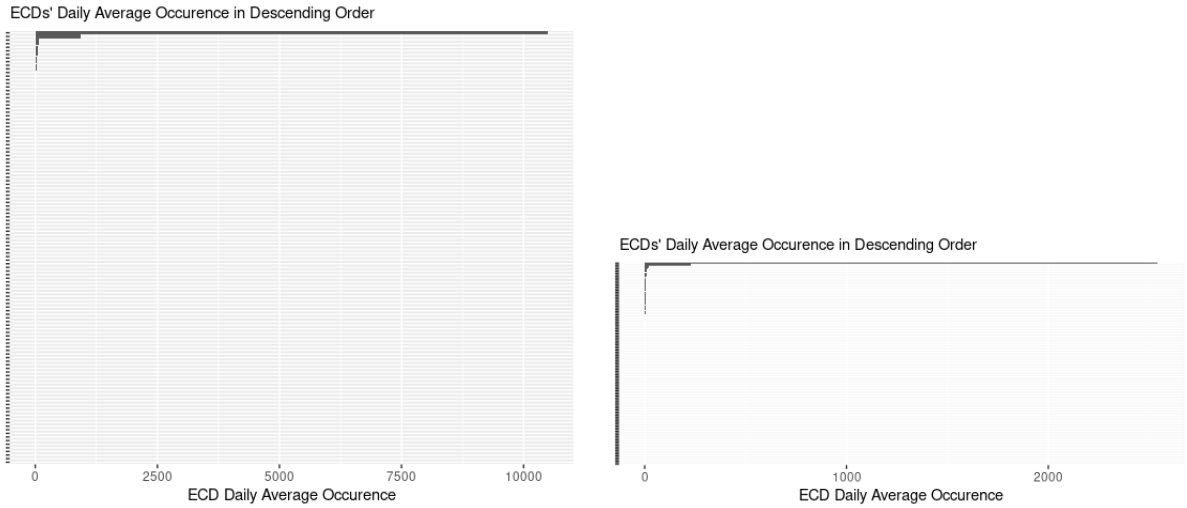


(a) Full Kaggle Data

(b) Project Data

Figure 1: Seattle 911 Incident Response: Event Clearance Description Overall Occurence

We can also see that the picture changes when we consider daily averages.



(a) Full Kaggle Data (b) Project Data

Figure 2: Seattle 911 Incident Response: Event Clearance Description Daily Average Occurence

From the plot we see some bars that significantly prevail over others. So, which are the event clearance description that occur the most? And do they also prevail on daily basis? We are considering the project data in detail.

Project data for Large Occurence ECDs:

```
S911IR%>%group_by(date(EC_DateTime),ECD)%>%summarise(daily = n())%>%
    ungroup()%>%group_by(ECD)%>%
    summarise(overall=sum(daily),daily_avg=mean(daily),daily_median=median(daily))%>%
    filter(overall>20000)%>%arrange(desc(overall))%>%
    select(ECD,overall,daily_avg,daily_median)%>%
  knitr::kable()
```

ECD	overall	daily_avg	daily_median
SUSPICIOUS PERSON	40587	19.08181	19
DISTURBANCE, OTHER	35039	16.53563	16
TRAFFIC (MOVING) VIOLATION	26387	12.46434	12
PARKING VIOLATION (EXCEPT ABANDONED VEHICLES)	25803	12.24051	12

We can also see group of bars in project data plot that have occurences between 8'000 and 20'000 (in full data version, these numbers are between 20'000 and 50'0000). And do they what is their average occurence on daily basis? Which are the event clearance description that form the group? We are considering the project data in detail.

Project data for Middle Occurence ECDs

```
S911IR%>%group_by(date(EC_DateTime),ECD)%>%summarise(daily = n())%>%
    ungroup()%>%group_by(ECD)%>%
    summarise(overall=sum(daily),daily_avg=mean(daily),daily_median=median(daily))%>%
    filter(overall>=8000 & overall<=20000)%>%
    arrange(desc(overall))%>%
    select(ECD,overall,daily_avg,daily_median)%>%
```

ECD	overall	daily_avg	daily_median
MOTOR VEHICLE COLLISION	9629	12.297573	12
SHOPLIFT	9612	4.732644	5
NOISE DISTURBANCE	9223	4.634673	4

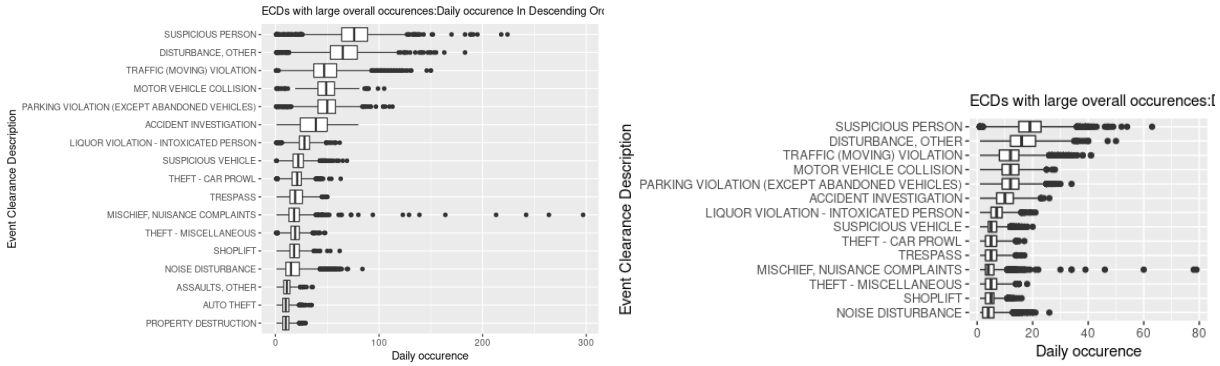
Those ECDs that are present, but can be viewed as separate occurences should also be noted. So, what are the ECDs that have barely existent occurences? We are considering the project data in detail and picking out that have least number of occurences. But we are considering the fact that data is probably missing between 2013 and 2014. Therefore we are considering years between July, 2010- February 2013 and July, 2014 - December 2017.

```
S911IR%>%filter(!((EC_DateTime>make_date(year=2014,month=2,day=1))
                  &(EC_DateTime<make_date(year=2014,month=7,day=1))))%>%
  group_by(date(EC_DateTime),ECD)%>%summarise(daily = n()) %>%
  ungroup()%>%group_by(ECD)%>%
  summarise(overall=sum(daily),daily_avg=mean(daily),daily_median=median(daily))%>%
  filter(overall<50)%>%arrange(overall)%>%select(ECD,overall,daily_avg,daily_median)%>%
  knitr::kable()
```

ECD	overall	daily_avg	daily_median
HARBOR - BOATING UNDER THE INFLUENCE	1	1.000000	1
HARBOR - MARINE FIRE	1	1.000000	1
HARBOR - VESSEL RECOVERY	2	1.000000	1
SOAP (STAY OUT OF AREA OF PROSTITUTION) ORDER VIOLATION	4	1.000000	1
TRAFFIC - SCHOOL ZONE ENFORCEMENT	6	1.000000	1
LIQUOR VIOLATIONS (BUSINESS)	7	1.166667	1
TRAFFIC - BICYCLE VIOLATION	8	1.000000	1
HARBOR - VESSEL ABANDONED	10	1.000000	1
HARBOR - VESSEL THEFT	11	1.000000	1
PORNOGRAPHY	14	1.000000	1
AWOL	15	1.000000	1
GAMBLING	15	1.000000	1
TRAFFIC - COMMUNITY TRAFFIC COMPLAINT (CTC)	15	1.071429	1
LOST PERSON	18	1.000000	1
ASSAULTS, GANG RELATED	20	1.000000	1
HARBOR - BOAT ACCIDENT	20	1.000000	1
CRISIS COMPLAINT - PICK-UP OR TRANSPORT	23	1.000000	1
HARBOR - ASSIST BOATER (NON EMERGENCY)	25	1.000000	1
DEMONSTRATION MANAGEMENT (Control tactics used)	26	1.083333	1

ECD	overall	daily_avg	daily_median
HARBOR - CODE VIOLATION	31	1.033333	1
NARCOTICS WARRANT SERVICE	33	1.031250	1
PURSUIT	34	1.000000	1
MENTAL PERSON PICK-UP OR TRANSPORT	36	1.058823	1
TRAFFIC CONTROL (SPECIAL EVENTS)	46	1.069767	1
HARBOR - DEBRIS, NAVIGATIONAL HAZARDS	47	1.021739	1

We are using boxplot to see more clear picture. And here we are considering group of ECDs that have overall occurence of more than 8'000. For full data, the trimming point for "Large overall" occurences had been above 20'000. But for Project data it had been 8'000 for this plot. We are comparing the Project data and Full Kaggle data results side by side. But the overall picture is the same.



(a) Full Kaggle Data (b) Project Data

Figure 3: Seattle 911 Incident Response: ECD Quantile Daily Average Occurence

3.1.3 Question: Do Initial Type Descriptions match Event Clearance Description, or do people commonly mistake the symptoms of the situation very often?

Here, we are bringing in the Initial Type Description to see whether they match the resulting Event Clearance Description. We are comparing the Project data and Full Kaggle data results side by side.

For this Dot plot, alpha opaqueness represents high daily average Initial Type Description.

From the Project data plot we see clearly that:

- "NA"s in ITDesc occur most frequently. But it is not a missing data. It is due to failure of callers to describe the incident. Operators at 911 do not initially know what happened.
- ECD "Suspicious person","Disturbance, Other" is most prevalent for most reasons. Which we can attribute to the fact that this type of description is applicable to most circumstances.
- ECD "PARKING VIOLATIONS (EXCEPT ABANDONED VEHICLES)" have the least daily average for ECD, but the most daily average for ITDesc. Which means while this type of ECD happens less, people identify the reason for this ECD most clearly.

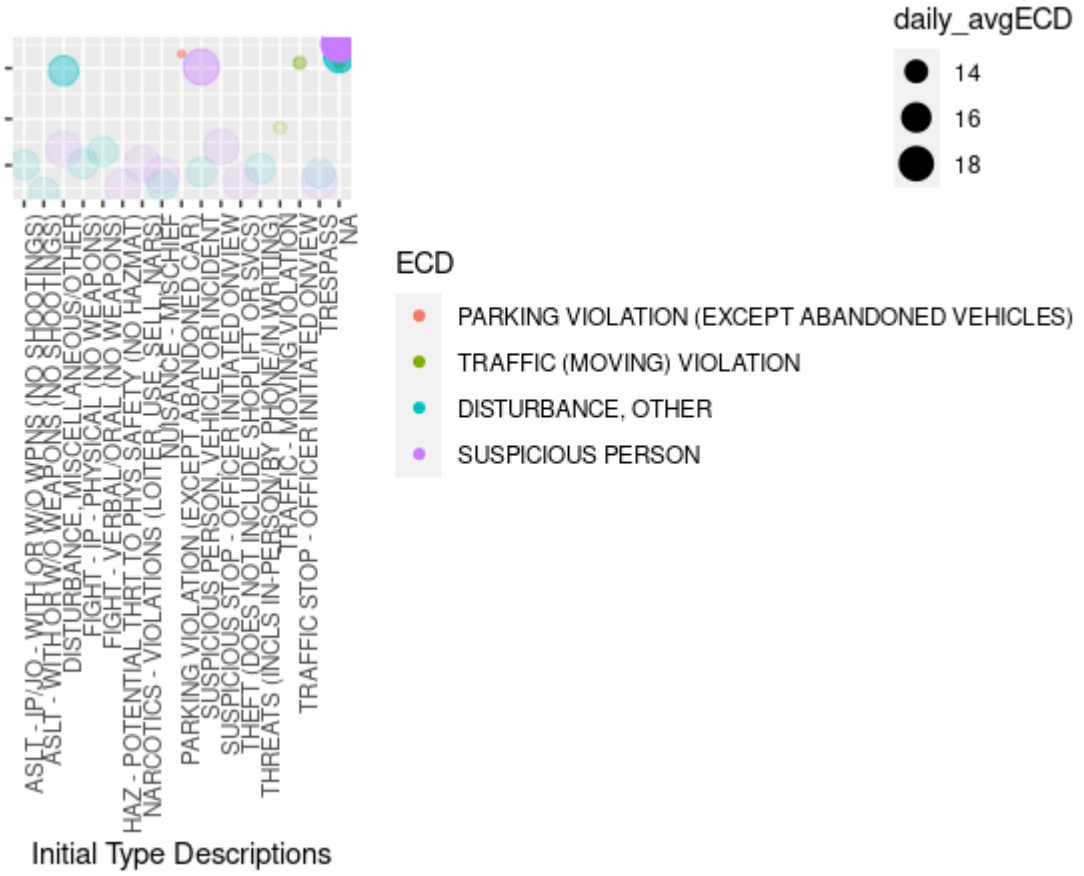


Figure 4: Seattle 911 Incident Response: Event Clearance Description vs Initial Type Description - Project Data

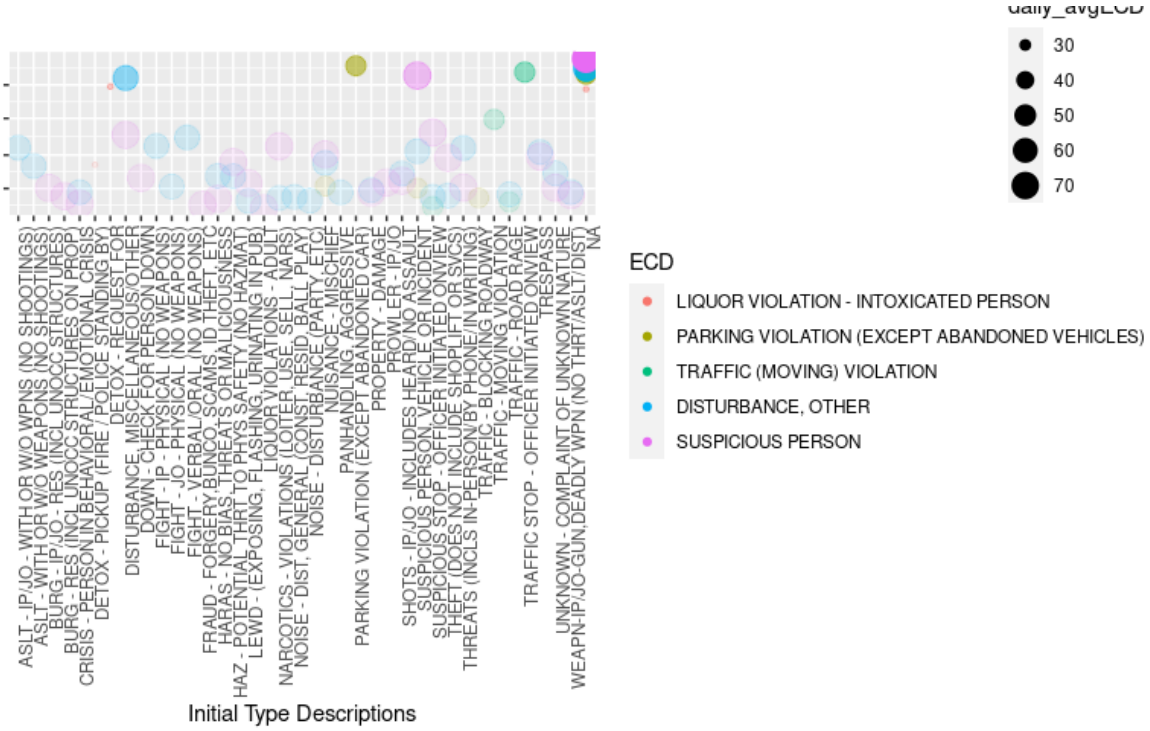


Figure 5: Seattle 911 Incident Response: Event Clearance Description vs Initial Type Description - Full Kaggle Data

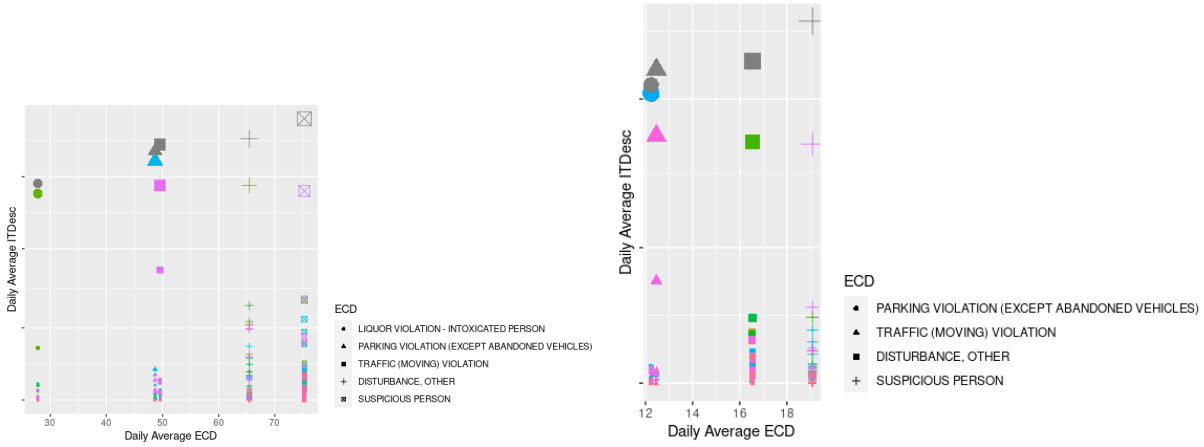
From full data, we see that:

- DETOX - REQUEST FOR result in DISTURBANCE, OTHER
- PARKING VIOLATIONS (EXCEPT ABANDONED VEHICLES), TRAFFIC - MOVING VIOLATION in ITDesc match the ECD directly

And here we see another more insightful aspect:

- SHOTS - IP/JO - INCLUDES HEARD/NO ASSAULT results in Event Clearance Description of SUSPICIOUS PERSON This may describe the degree of fear of guns and shots in Seattle citizens, and the fact that these rarely are founded assumptions.

Here is another version of the plot to see whether there are relationship between the ECD and ITDesc. Again, we are comparing the Project data and Full Kaggle data results side by side.



(a) Full Kaggle Data (b) Project Data

Figure 6: Seattle 911 Incident Response: Event Clearance Description vs Initial Type Description

For this version of the plot, color represents ITDesc groups, while ECDs are represented by shape. For full data, the trimming point for "Large ECD" occurences had been above 50'000. But for Sample data it had been 20'000 for this plot. We are comparing the results side by side. But the overall picture is the same.

3.1.4 Question: Is there relation between timing and event clearance description?

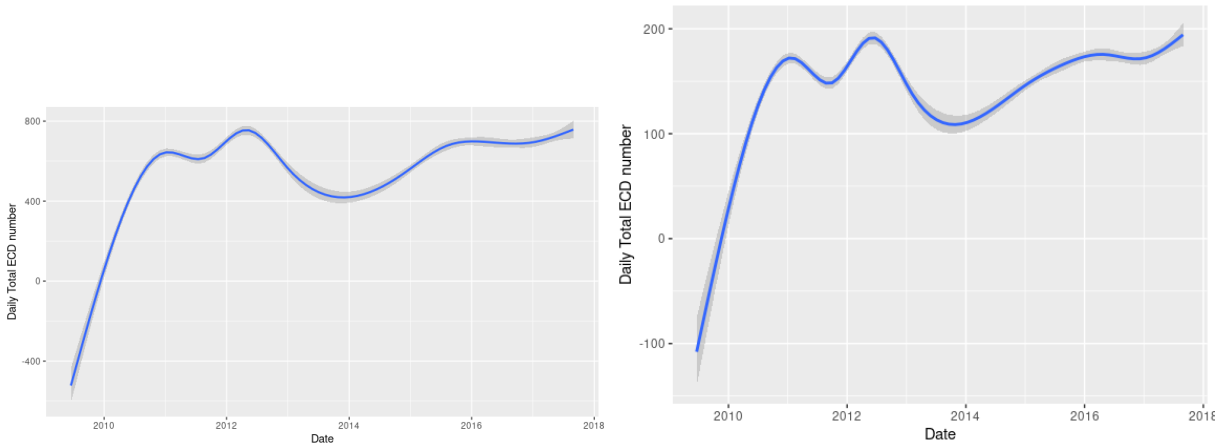
3.1.4.1 Was there increase/decrease in event clearance description overall occurrence over time?

We are building Timeline plot to see this.

From the plot, we see sharp drop in daily total number around 2014. So what happened here?

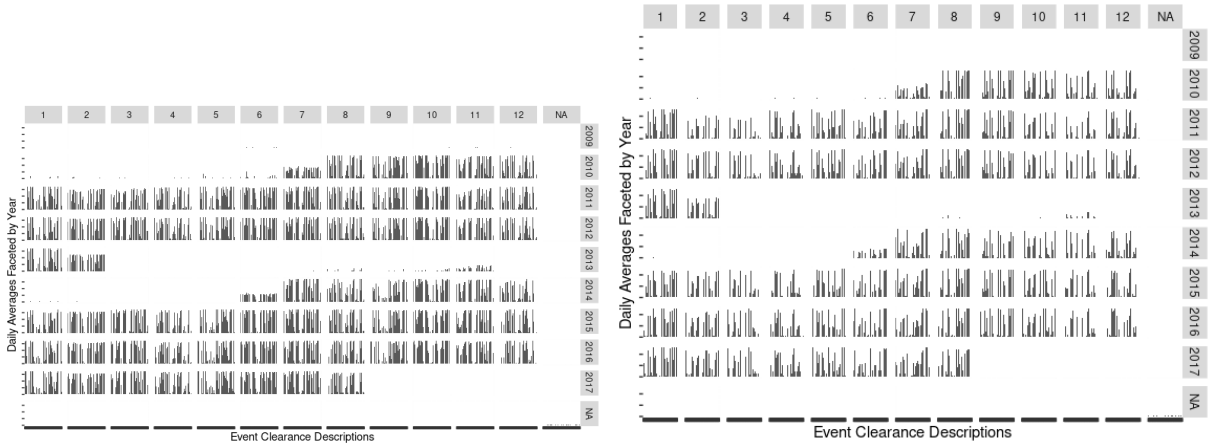
We investigate by detailing the plot by ECD and faceting daily averages by year versus month.

And here we see data largely missing between March of 2013 to June of 2014. And we also see that data after 1st of September, 2017 is missing. Therefore we have trimmed the Project data after that point.



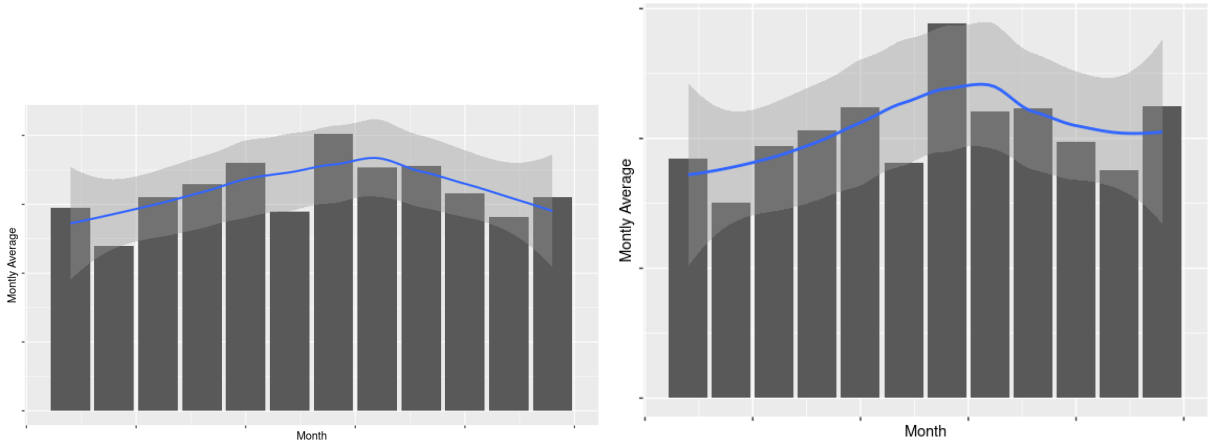
(a) Full Kaggle Data (b) Project Data

Figure 7: Seattle 911 Incident Response: ECD Timeline



(a) Full Kaggle Data (b) Project Data

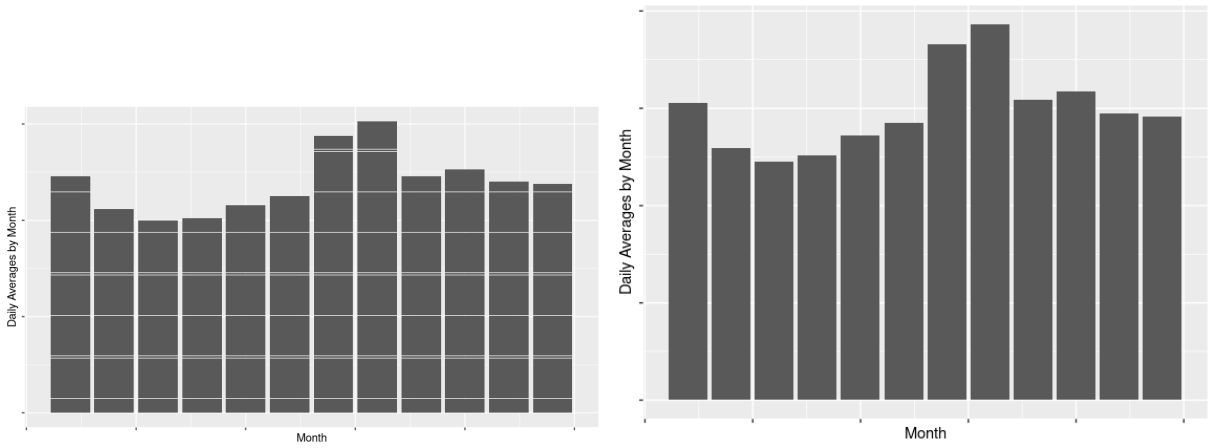
Figure 8: Seattle 911 Incident Response: ECD Timeline Faceted by Year and Month



(a) Full Kaggle Data (b) Project Data

Figure 9: Seattle 911 Incident Response: ECD Monthly Average

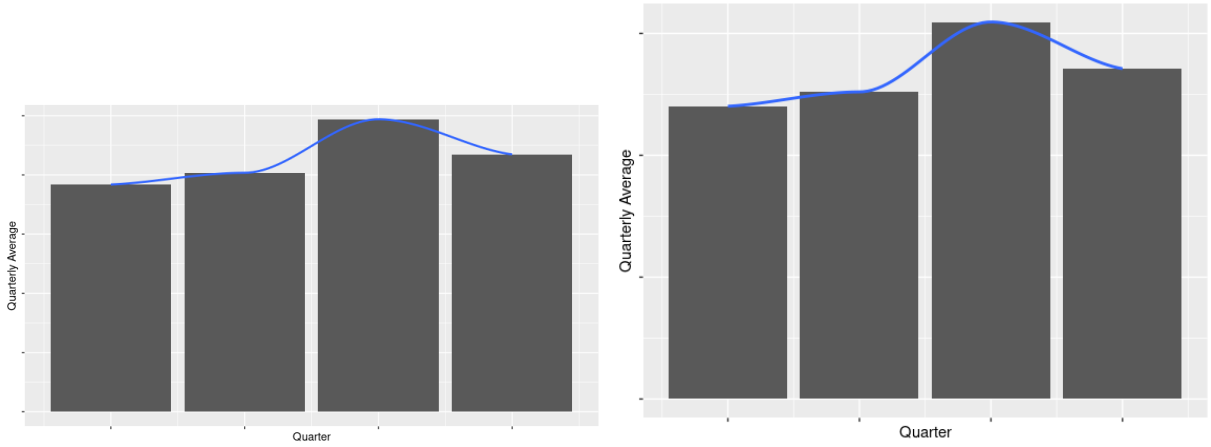
3.1.4.2 During which month event clearance description overall occurrence increases/decreases? We also want to see what happens on a daily basis.



(a) Full Kaggle Data (b) Project Data

Figure 10: Seattle 911 Incident Response: ECD Daily average by Month

From the histogram, we see surge of daily averages in July and August.



(a) Full Kaggle Data (b) Project Data

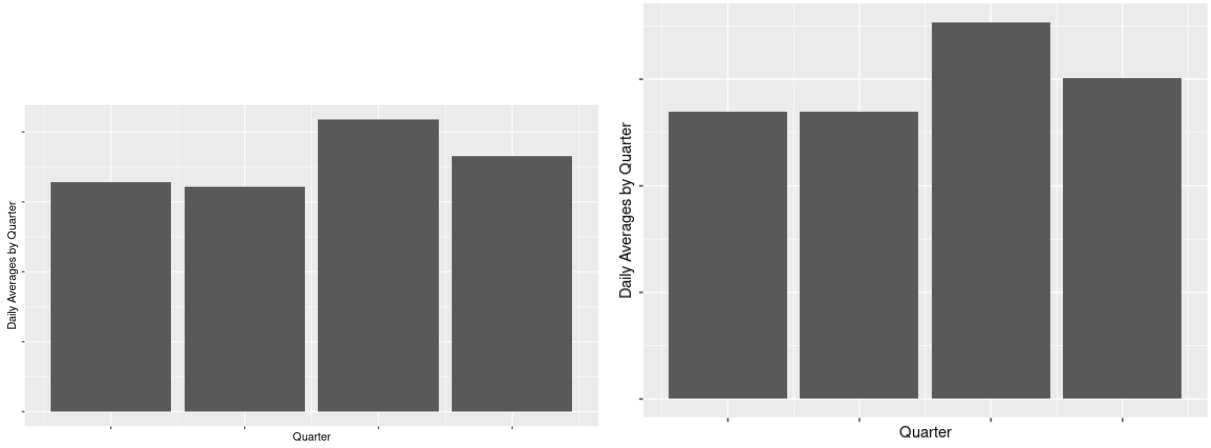
Figure 11: Seattle 911 Incident Response: ECD Quarterly Average

3.1.4.3 During which quarter event clearance description overall occurrence increases/decreases? We also want to see what happens on a daily basis.

We can see definite surge of daily average number of incidents in 3rd quarter.

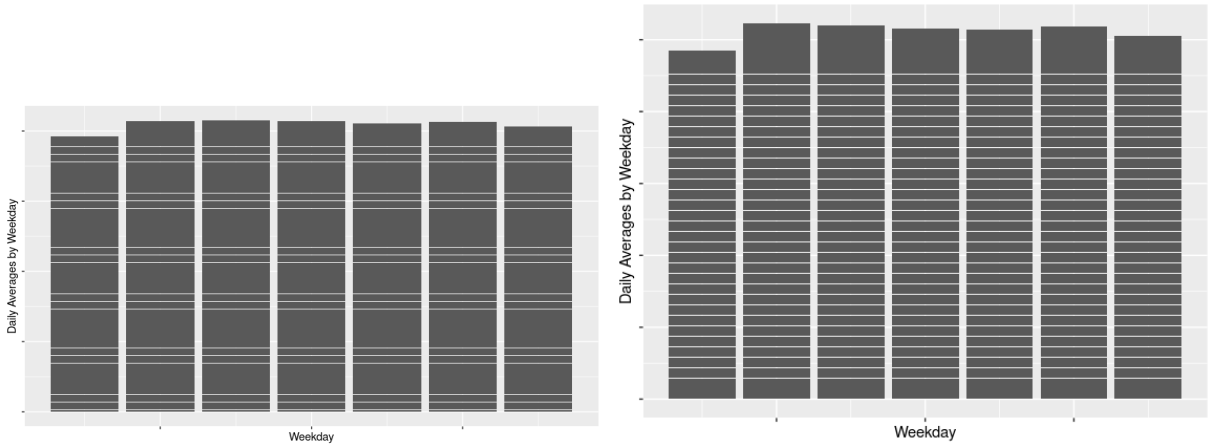
3.1.4.4 On which day of the week event clearance description overall occurrence increases/decreases? According to the plot, none of the weekdays can be viewed as special.

3.1.4.5 During which hour of the day event clearance description overall occurrence increases/decreases? Hourly averages are practically the same. But in the sampled data, there is slight



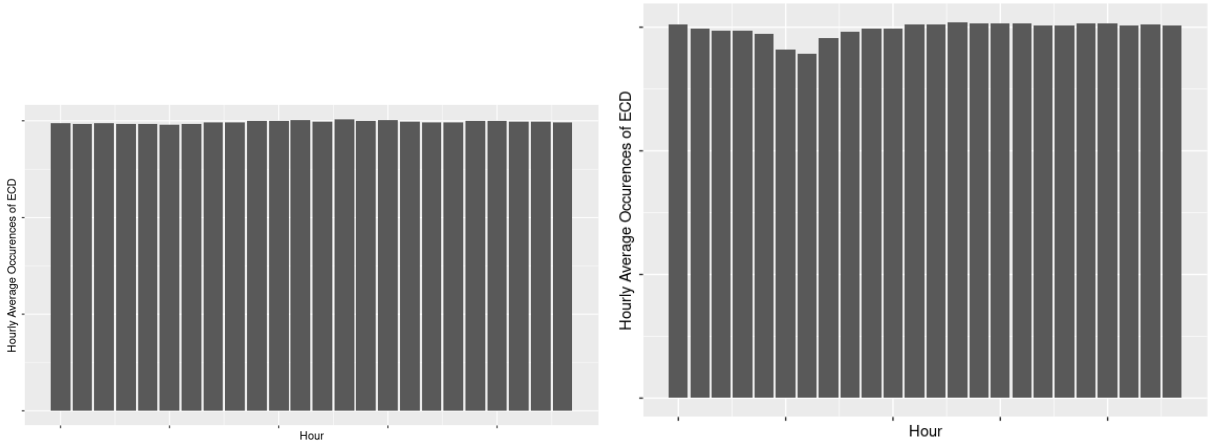
(a) Full Kaggle Data (b) Project Data

Figure 12: Seattle 911 Incident Response: ECD Daily average by Quarter



(a) Full Kaggle Data (b) Project Data

Figure 13: Seattle 911 Incident Response: ECD Daily average by Weekday



(a) Full Kaggle Data (b) Project Data

Figure 14: Seattle 911 Incident Response: ECD Daily average by Hour

drop between 7AM and 8AM in daily average occurence. It can be due to sampling process of Project data.

3.1.5 Question: Is there relation between location and event clearance description?

3.1.5.1 Is there certain prevalence in overall number of event clearance description in certain location? We are now proceeding with the cleaned data prepared for training. Are there certain locations where calls to 911 prevail? We are further considering the yearly median for each location. And we are going to determine the value for average yearly Incident Response encounter in Seattle for median by month of each location.

```
loc_yearly_median<-S911IR%>%group_by(ILoc,EC_Year)%>%
    summarize(count=as.integer(n()))%>%select(ILoc,EC_Year,count)%>%
    group_by(ILoc)%>%summarise(year_med=median(count))%>%select(ILoc,year_med)
seattle_loc_med_year_mean<-mean(loc_yearly_median$year_med)
```

[1] 2.537033

Now we are ready to determine which locations exceed yearly norm for average location incident. From there, we are able to determine the monthly averages for these locations and filter out above average locations by months.

```
locMatrix<-S911IR%>%group_by(ILoc,EC_Year,EC_Month)%>%summarise(count=as.integer(n()))%>%
    group_by(ILoc,EC_Month,)%>%summarize(month_median=median(count))%>%
    select(ILoc,EC_Month,month_median)%>%spread(key=EC_Month,value=month_median,fill = 0)%>%
    mutate(month_sum = sum(c_across(where(is.numeric))))
locMatrix<-locMatrix%>%filter(month_sum>=seattle_loc_med_year_mean)%>%
    arrange(desc(month_sum))%>%select(-"month_sum")
locMatrix<-as.matrix(locMatrix)
rownames(locMatrix)<- locMatrix[,1]
locMatrix<- locMatrix[,-1]
mode(locMatrix)<-"integer"
locMatrix[1:4,]%>%knitr::kable(align="c")
```

	1	2	3	4	5	6	7	8	9	10	11	12
(47.600464242, -122.330807499)	79	55	70	56	64	42	40	41	36	64	42	71
(47.600464, -122.33081)	64	59	67	58	57	63	59	48	14	0	55	82
(47.602413, -122.331085)	44	42	63	45	48	38	32	38	7	0	23	39
(47.60046424, -122.3308075)	46	31	32	33	43	25	34	24	22	33	36	34

```
locMatrix <- sweep(locMatrix, 2, colMeans(locMatrix, na.rm = TRUE))
mode(locMatrix)<-"integer"
#Replace all the negative values with 0
locMatrix<-pmax(locMatrix,0)
locMatrix<-locMatrix[rowSums(locMatrix)>0,]

locMatrix_Sums<-as.matrix(rowSums(locMatrix))
locMatrix_Sums <- sweep(locMatrix_Sums, 1, rowMeans(locMatrix, na.rm = TRUE))
conLocs<-as.list(rownames(locMatrix_Sums>0))
```

Here we have locations that have persistent occurences of ECDs by average and median. Now, let us see whether same ECDs occur at these locations. We count the groups of ECDs that have occurence for each of those locations and average the results by ECD.

Let us see where exactly these points lie on the map by using Latitude and Longitude. And how intense they look. But we are not using every location. Just the top 1000 locations by ocurrence of ECDs

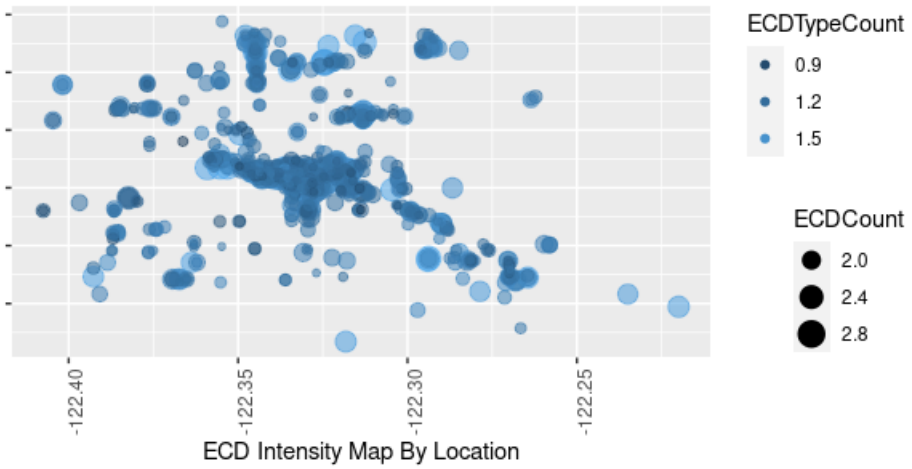


Figure 15: Seattle 911 Incident Response: ECD Intensity Map

We can see that at outskirts of the city, same ECDs occur in larger number.

But what happens if we facet the same plot by month? But we have to choose the most intense locations for this. We are choosing 100 top intense locations.

We can do the same for weekdays. Here we are considering daily totals averaged by Weekdays.

3.2 Chi-squared Tests

As all of the data are non-continuous, we are turning to Pearson's Chi-Squared test for Count Data to determine the correlation between ECD~ITDesc, ECD~EC_Month, ECD~EC_Weekday, ECD~Iloc pairs separately.

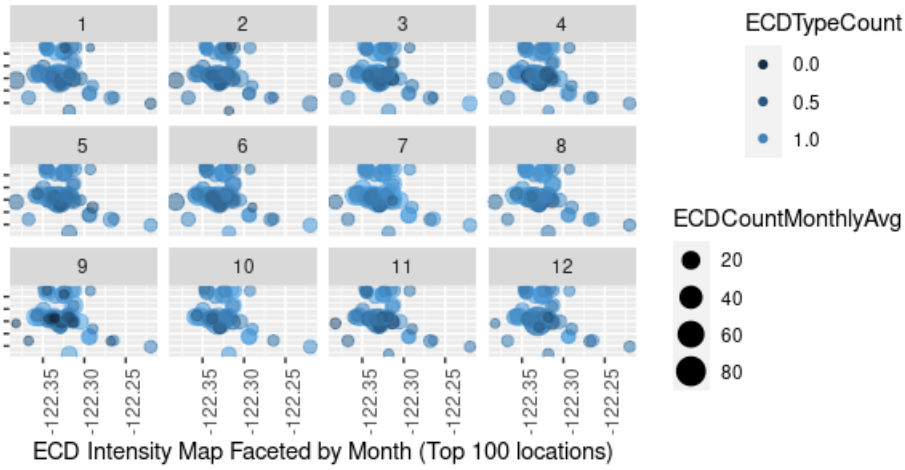


Figure 16: Seattle 911 Incident Response: ECD Intensity Map Faceted by Month

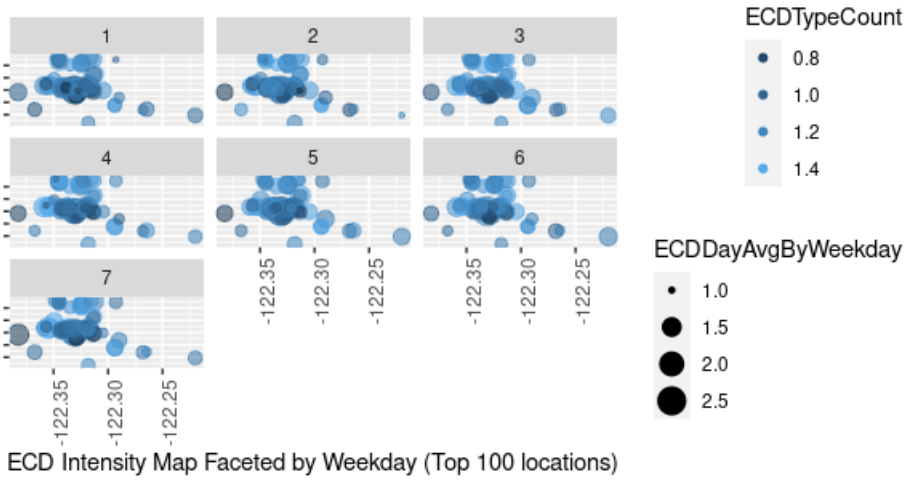


Figure 17: Seattle 911 Incident Response: ECD Intensity Map Faceted by Month

Chi-test for ECD~ILoc yielded pvalue of 1, while all the other pairs yielded 0.

```
#ECD~ILoc pair results
set.seed(1)
ECDILocMatrix<-S911IR%>%group_by(ECD,ILoc)%>%summarise(count=as.integer(n()))%>%select(ECD,ILoc,coun
ECDILocMatrix<-as.matrix(ECDILocMatrix)
rownames(ECDILocMatrix)<- ECDILocMatrix[,1]
ECDILocMatrix<- ECDILocMatrix[,-1]
mode(ECDILocMatrix)<-"integer"
rownamesECDILocMatrix<- rownames(ECDILocMatrix)
ECDILocMatrix<- ECDILocMatrix[,-1]
ECDILocMatrix<- sweep(ECDILocMatrix, 1, rowMeans(ECDILocMatrix, na.rm = TRUE))
ECDILocMatrix<- sweep(ECDILocMatrix, 2, colMeans(ECDILocMatrix, na.rm = TRUE))
ECDILocMatrix<-pmax(ECDILocMatrix,0)
ECD_ITDescChi<-chisq.test(ECDILocMatrix)
ECD_ITDescChi$p.value
```

[1] 1

But we are refraining from any conclusion on this point.

3.3 KMeans clustering

We will use kmeans clustering and we will see the results on the map.

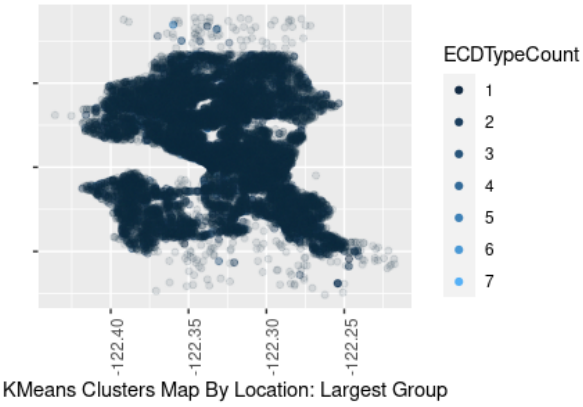
```
#ECD~ILoc pair results
rownames(ECDILocMatrix)<- rownamesECDILocMatrix

k <- kmeans(ECDILocMatrix, centers = 100, nstart=25)
summary(k)%>%knitr::kable()
```

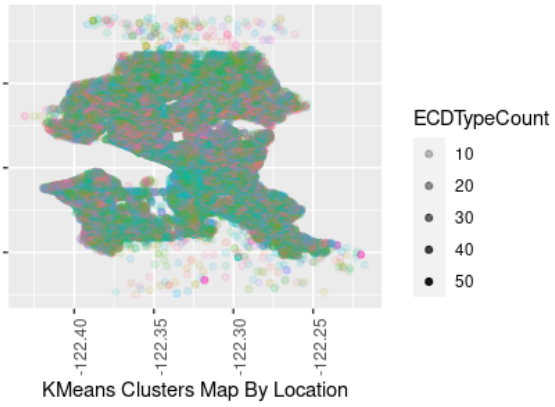
	Length	Class	Mode
cluster	71072	-none-	numeric
centers	11500	-none-	numeric
totss	1	-none-	numeric
withinss	100	-none-	numeric
tot.withinss	1	-none-	numeric
betweenss	1	-none-	numeric
size	100	-none-	numeric
iter	1	-none-	numeric
ifault	1	-none-	numeric

```
#ECD~ILoc pair kmeans clustering results
S911IR%>%distinct(ILocGroup,ILoc)%>%group_by(ILocGroup)%>%
  summarise(locations=n())%>%
  arrange(desc(locations))%>%knitr::kable()
```

ILocGroup	locations	ILocGroup	locations	ILocGroup	locations
14	18032	30	1991	76	1083
52	4438	59	1981	22	1012
94	4336	11	1832	13	992
35	3253	55	1728	4	894
57	2356	36	1574	29	886
93	2288	18	1506	31	872
88	2221	81	1436	23	862
2	2129	61	1215	...	...



(a) Larges Group



(b) Other 99 Groups

Figure 18: Seattle 911 Incident Response: KMeans Clusters Map By Location

3.4 Final model performance

Further, we will train the **S911IR** data with randomForest and knn method.

3.4.1 Random Forest: Growing the tree

For tuning and running, we have prepared the **train_set** and **test_set** with variety of subsamples starting with 10'000 and ending with 100'000.

Here is the script:

```
S911IR_trainsmall<-sample_n(S911IR,100000)


test_index <- createDataPartition(y = S911IR_trainsmall$ECD, times = 1,
                                  p = 0.2, list = FALSE)
train_set <- S911IR_trainsmall[-test_index,]
test_set <- S911IR_trainsmall[test_index,]
```

We have tuned random forest model for variety of features and number of features. With analysis, we have determined that ECD is largely dependent on location, than any other feature. Therefore, we started with location features and added features with each run and with variety of nTree, mTry and nSamp parameters for the training. At final, we have chosen 8 features for the random forest. We are not using **ILocGroup** feature calculated by kmeans clustering in the final model.

```
train_set<-train_set%>%select(ECD,Latitude,Longitude,
                              ITDescN,EC_Year,EC_Quarter,
                              EC_Month,EC_Day,EC_Weekday)
test_set<-test_set%>%select(ECD,Latitude,Longitude,
                            ITDescN,EC_Year,EC_Quarter,
                            EC_Month,EC_Day,EC_Weekday)
```

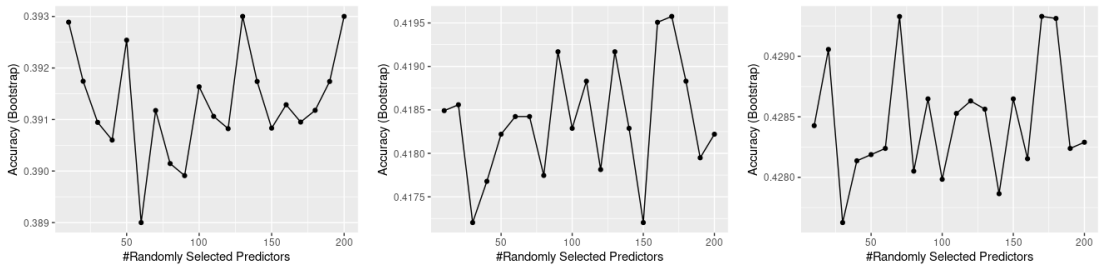
And following is the result for tuning and importance of final features chosen.

```
train_rf <-  train(train_set[, -1], factor(train_set[,1]),
                   method = "rf",
                   nTree = 500,
                   tuneGrid = data.frame(mtry = seq(10, 200, 10)),
                   nSamp = 10000)


train_rf$bestTune%>%knitr::kable()
```

Sample	10'000	50'000	100'000
mtry	130	170	170

Final training script ran 1:20 hours for 100'000 data.



(a) Sample 10.000 (b) Sample 50.000 (c) Sample 100.000

Figure 19: Random Forest Parameter Tuning

```
varImp(train_rf)
```

	10'000	50'000	100'000
ITDescN	100.00	100.00	100.00
Latitude	50.60	49.79	48.84
Longitude	49.55	49.69	48.43
EC_Day	35.14	38.84	41.14
EC_Month	19.71	20.60	21.23
EC_Weekday	19.04	19.39	21.11
EC_Year	12.98	12.81	11.93
EC_Quarter	0.00	0.00	0.00

We see that we can drop the EC_Quarter least important feature.

Now we are fitting the random tree with *ntree=500* and with a tuned parameter *mtry=170*. This script ran about 3 minutes.

```
library(randomForest)
fit_rf <- randomForest(train_set[, -1], factor(train_set[,1]), ntree = 500, mtry = 170)
importance(fit_rf)
```

	MeanDecreaseGini
Latitude	10822.851
Longitude	10606.341
ITDescN	14806.409
EC_Year	3148.856
EC_Quarter	1668.354
EC_Month	4617.753
EC_Day	7936.642

	MeanDecreaseGini
EC_Weekday	4841.357

```
varImpPlot(fit_rf,type=2)
```

And with a plot, we see that variable importance order is the same as varImp() function results with train_rf.

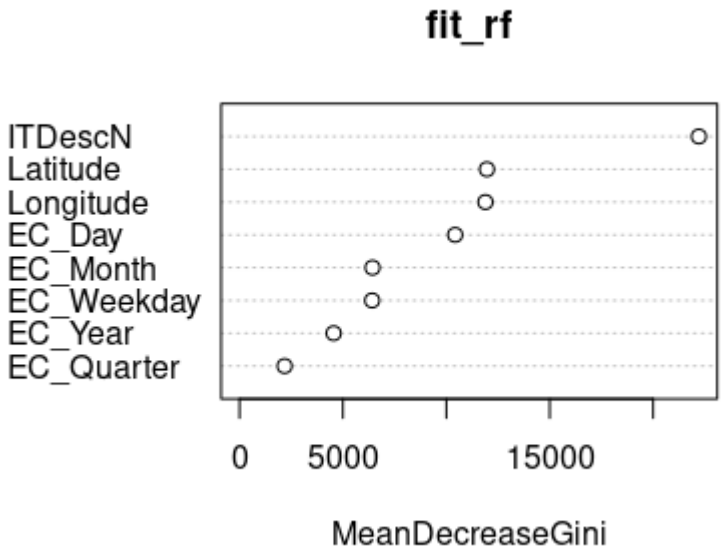


Figure 20: Random Forest Variable Importance

```
y_hat_rf <- predict(fit_rf, test_set[ ,-1])
cm <- confusionMatrix(y_hat_rf, factor(test_set[,1]))
cm$overall
```

	x
Accuracy	0.4417398
Kappa	0.4100457
AccuracyLower	0.4348497
AccuracyUpper	0.4486469
AccuracyNull	0.1135275
AccuracyPValue	0.0000000
McnemarPValue	NaN

3.4.2 Knn algorithm

Tuning for 8 Features and k=seq(10,100,10). Training algorithm ran 67 minutes for this tuning range. Although, this tuning yielded inconclusive results.

```
train_knn <- train(train_set[ ,-1], factor(train_set[,1]),
                   method = "knn",
                   tuneGrid = data.frame(k = seq(10, 300, 10)))
train_knn$bestTune%>%knitr::kable()
```

	k
2	410

Tried again with 3 Features and k=seq(10,100,10). Final training script ran 192 minutes for 100'000 data.

```
train_set<-train_set%>%select(ECD,Latitude,Longitude,ITDescN)
test_set<-test_set%>%select(ECD,Latitude,Longitude,ITDescN)

train_knn <- train(train_set[ ,-1], factor(train_set[,1]),
                   method = "knn",
                   tuneGrid = data.frame(k = seq(10, 300, 10)))
train_knn$bestTune%>%knitr::kable()
max(train_knn$results$Accuracy)%>%knitr::kable()
```

	k	x
5	50	0.4391884

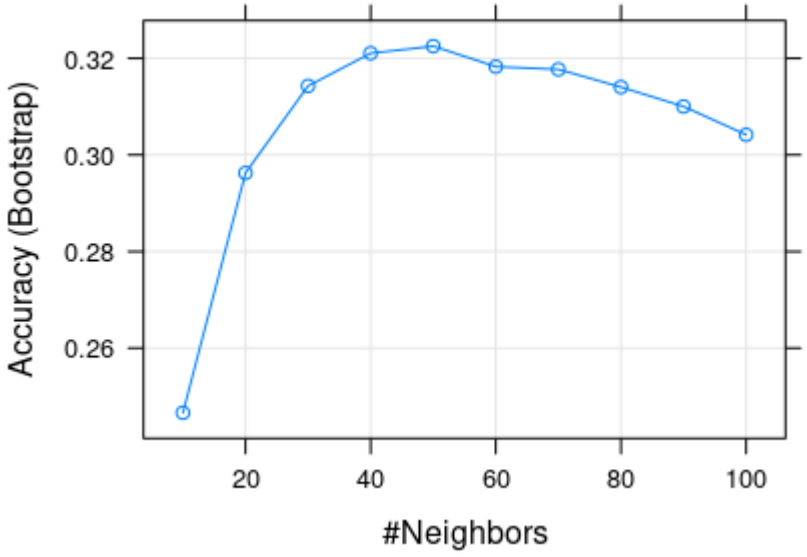


Figure 21: knn Parameter Tuning

4 Conclusion

We have trained basic and not-so-precise machine-learning algorithm. Accuracy results should be improved further for better use in real life. For now, we have not progressed above 0.44. It is not a good result.

4.1 About the Similarity between 911 calls and 103 calls

There is a similarity between records of US 911 and Mongolian 102 (Police Hotline), 103 (Medical Emergency Number) records. People describe the symptoms of the situation, and the operator spends some time evaluating the situation by asking questions. In many of the cases, both people in distress and the call receiving operator fail to describe the situation properly.

With ECD predicting algorithm:

- Time of the call would improve
- Precision of questions posed by operator could be improved
- Properly recognized prank calls will not affect the life-threatening or other serious situation calls

4.2 Tests and evaluation to be done

The result predictions should be tested with real life situations. Further examination of data must be done regularly for improvement.

4.3 Technical requirements for the execution

Mobile technology, government regulations restrict Mongolia from tracking the phone calls to its source. Prior to analyzing and training the data, we would need to improve the tracking ability for 102 and 103

operators.

As mobile phones dominate the field, and people have unlimited access to smart phones, the calls should be considered to be done from mobile phones. If such project should be initiated, we would need access to mobile operators' VLR info, and we would need the VLR info to be gathered regularly or on demand.

Even better solution would be creating an app for mobile phones that uses improved machine-learning algorithm. But we still would need the trained prediction run on the 102, 103 operators' side. Therefore the predicting function should be offered as an API available for developers to use.

4.4 Plan for further use of data

I will continue on with this project and try out new models to improve accuracy. Further, I am considering simulating phone numbers and call Timestamp field based on AS_DateTime field. I think it will help prepare for real-life data and will add a possibility to predict emergency levels of calls. Also it will help achieve one of this Project's initial goals - which is to recognize prank calls - that was sadly not achieved for now.

Contact Information

If you have any questions regarding the project, please feel free to contact me at any of my emails: *khaliun83@yahoo.com*[5], *khaliun@spoon.mn*[6]; or feel free to visit my *Linkedin Profile*[7]

References

Books

- Irizarry (2021)
- Xie, Dervieux, & Riederer (2020)

Articles

- Boehmke (2021)

Manuals

- R Core Team (2020)
- Wickham et al. (2021)
- Wickham et al. (2020)
- Wickham (2019)
- Datacamp team (2020)

Boehmke, B. (2021). UC Business Analytics R Programming Guide: K-Means Cluster Analysis. Retrieved from https://uc-r.github.io/kmeans_clustering

Datacamp team. (2020). *Predict.lm: Predict Method for Linear Model Fits.* Datacamp. Retrieved from https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/predict.lm

Irizarry, R. A. (2021). *Introduction to Data Science: Data Analysis and Prediction Algorithms with R.* CRC Press. Retrieved from https://rafalab.github.io/dsbook/

R Core Team. (2020). *R: A Language and Environment for Statistical Computing.* Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/

Wickham, H. (2019). *Tidyverse: Easily Install and Load the Tidyverse.* Retrieved from https://CRAN.R-project.org/package=tidyverse

Wickham, H., Chang, W., Henry, L., Pedersen, T. L., Takahashi, K., Wilke, C., Woo, K., Yutani, H. & Dunnington, D. (2020). *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics.* Retrieved from https://CRAN.R-project.org/package=ggplot2

[5] mailto:khaliun83@yahoo.com
[6] mailto:khaliun@spoon.mn
[7] https://www.linkedin.com/in/khaliun-bat-ochir-334925b4/

Wickham, H., François, R., Henry, L. & Müller, K. (2021). *Dplyr: A Grammar of Data Manipulation.* Retrieved from https://CRAN.R-project.org/package=dplyr

Xie, Y., Dervieux, C. & Riederer, E. (2020). *The R Series: RMarkdown Cookbook.* CRC Press Tayler; Francis Group A Chapman And Hall Book. Retrieved from https://bookdown.org/yihui/rmarkdown-cookbook/