# New York Fire Incident Reporting System (NYFIRS)

## Analysis of Fire Department Activity from 2013 to 2018

by **Gino Parages**

**“This dataset contains detailed information on incidents handled by FDNY Fire (non-EMS) units and includes fire, medical and non-fire emergencies. The data is collected in the New York Fire Incident Reporting System (NYFIRS), which is structured by the FDNY to provide data to the National Fire Incident Reporting System (NFIRS). NFIRS is a modular all-incident reporting system designed by the U.S. Fire Administration. After responding to an incident, FDNY officers complete one or more of the NFIRS modules, depending upon the type of incident. The information in these modules describes the kind of incident responded to, where it occurred, the resources used to mitigate it. Although NFIRS was designed specifically to understand the nature and causes of fire, as well as civilian fire casualties and firefighter injuries, it has been expanded to collect basic information on all incidents to which fire units respond.”**

https://opendata.cityofnewyork.us/

**From this data it is possible to see the amount of time the New York City Fire Department spends protecting citizens of each of the five boroughs from the dangers of daily life.**

**Across the board we can expect an average of ~40,000 hours per year spent answering calls and solving problems for those in need. Staten Island is separate in this due to the significantly lower population, however their needs are still understood in the charts included.**

**There is also a correlation in the type of alarm dispatchers have called and the amount of time required to solve these issues.**

This chart represents the total NYC breakdown of alarms and average time (in minutes) spent each time this type of alarm is called.

This metric is calculated as:
**Last Unit Departure - Incident Time = Total Incident Duration**

**Reference below to understand what units are sent to the call location when this type of alarm is called.**
http://www.fdnewyork.com/aa.asp
https://en.wikipedia.org/wiki/New_York_City_Fire_Department

**All of the below numbers are what the total response is filled out to. (i.e. 8 engines on a second alarm means the 4 that were already there plus 4 more.) Full list of codes and descriptions:** http://www.fdnewyork.com/10code.asp

**First Alarm (signal 1-1)**
First alarm (signal 1-1) is response transmitted “box after initial” (Upon additional information or sources received at dispatch operations, dispatchers will fill the optimum assignment compared to the minimum response. Response assignment varies depending on the nature of the reported emergency. This is not a signal that there is a working fire or emergency, A “10–75” or signal 7-5 (announced as an “all hands”) used by a responding unit or chief is confirmation of a fire or emergency)

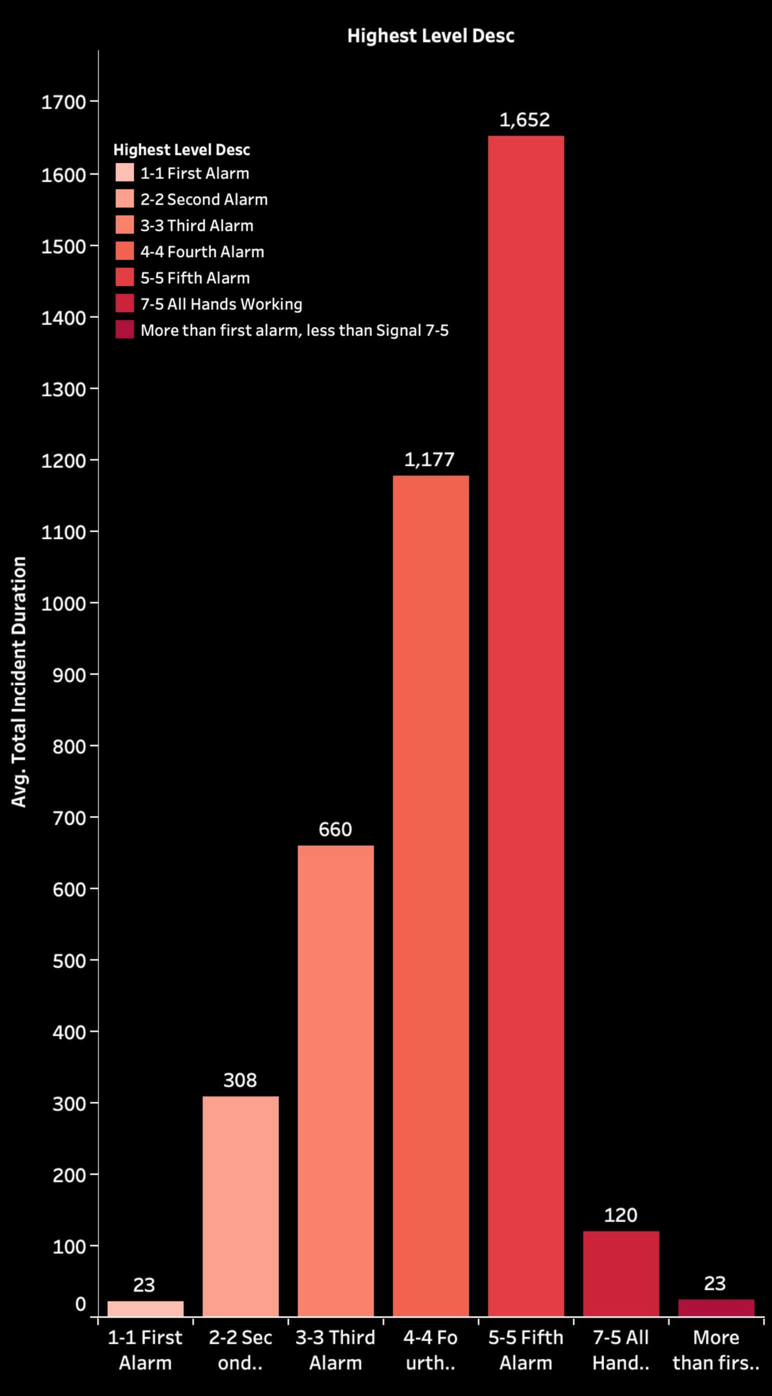
**Second alarm (signal 2-2)**
8 engines
5 ladders
5 battalion chiefs
1 rescue
1 squad
1 deputy chief
1 RAC unit
1 satellite
safety battalion
SOC battalion
1 tactical support unit
field comm
field comm battalion
communications unit

**Third alarm (signal 3-3)**
12 engines
7 ladders
6 battalion chiefs
1 rescue
1 squad
1 deputy chief
1 RAC unit
1 satellite
safety battalion
SOC battalion
1 tactical support unit
field comm
field comm battalion
communications unit
mask service unit
Air Recon Chief (on Bklyn box)

**Fourth alarm (signal 4-4)**
16 engines
9 ladders
6 battalion chiefs
1 rescue
1 squad
1 deputy chief
1 RAC unit
1 satellite
safety battalion
SOC battalion
1 tactical support unit
field comm
field comm battalion
communications unit
mobile command unit
planning section chief
Air Recon Chief (on Bklyn box)

**Fifth alarm (signal 5-5)**
20 engines
11 ladders
6 battalion chiefs
1 rescue
1 squad
1 deputy chief
1 RAC unit
1 satellite
safety battalion
SOC battalion
1 tactical support unit
field comm
field comm battalion
communications unit
mobile command unit
planning section chief
Air Recon Chief (on Bklyn box)

**Signal 7-5 (using all hands)**
4 engines
3 ladders
2 battalion chiefs
1 rescue
1 squad
1 deputy chief




### Units & Descriptions

**Source:** https://en.wikipedia.org/wiki/New_York_City_Fire_Department

**Engine —** A typical FDNY engine company, also known as a pumper or rig. Pictured is Engine Co. 34, quartered in Manhattan. FDNY engine companies are tasked with fire suppression; securing a water supply from a fire hydrant, deploying handlines, then extinguishing a fire. These units respond to other emergencies as well. The apparatus of an engine is known as a pumper truck and carries a pump (usually 1,000–2,000 gallons per minute), a water tank (usually 500 gallons), fire hoses of varying diameters (usually 1 3/4”, 2 1/2”, 3 1/2” and 4”) in 50’ lengths, emergency medical services supplies, ground extension ladders, and an assortment of tools.

**Ladder —** A tiller or tractor-drawn aerial ladder is another type of Ladder Truck operated by the FDNY. Pictured is a tiller ladder truck operated by Ladder Co. 5, quartered in Manhattan. FDNY ladder companies (also known as truck companies) are tasked with forcible entry, search and rescue, ventilation, and ladder-pipe operations at the scene of a fire. A Ladder Company can operate three types of ladder trucks: The first ladder truck is an aerial ladder truck, equipped with a 100’ aerial ladder mounted at the rear of the apparatus; The second ladder truck is a tower ladder truck, equipped with either a 75’ or 95’ telescoping boom and bucket mounted in the center of the apparatus; The third ladder truck is a tractor drawn aerial ladder truck, or tiller/tractor trailer ladder truck known as a Hook and Ladder Truck, equipped with a 100’ aerial ladder.

**Rescue —** A typical FDNY rescue company, also known as a rescue truck. Pictured is Rescue Co. 1, which serves a large portion of Manhattan. FDNY Rescue Companies are composed of the elite, highly and specially trained, most experienced members of the New York Fire Department. A rescue company is tasked with responding to and dealing with specialized fire and rescue incidents that are beyond the scope and duties of a standard engine or ladder company. Rescue companies operate rescue trucks, colloquially known as “tool boxes on wheels”, which carry a wide variety of specialized tools and equipment to aide in operations at technical rescues, such as rope rescue, building collapse rescue, confined space rescue, trench/excavation rescue, machinery rescue, and water rescue.

**Squad —** Haz-Mat. Company 1, quartered in Queens, responds to all major hazardous materials-related calls citywide.FDNY squad companies are also composed of specially trained firefighters of the New York Fire Department. Squad companies were initially established by the FDNY to serve as “manpower companies” to supplement the manpower and operations of engine and ladder companies. Today, squad companies can function as either engine or ladder companies at the scene of a fire, but are also equipped with the same equipment and specialized tools as the rescue companies. Some of these tools include The Jaws Of Life, various specialized cutting and lifting equipment and other tools. In particular, members of a squad company are highly trained in mitigating hazardous materials (hazmat) incidents, supplementing the FDNY’s single hazmat company. Squad companies also operate a Freightliner M2-based medium rescue as a second piece of apparatus in response to Haz-Mat incidents.

**Chief Units —** FDNY EMS Supervisor 936’s Command pickup, a 2012 GMC Extended Crew Cab 4X4 Diesel Pickup XLS built by Odyssey Specialty Vehicles. A Battalion Chief’s Unit is a command vehicle tasked with the responsibility of delivering a Battalion Chief to the fire scene. Once on the fire scene, the vehicle then takes on the role of a Command Vehicle, utilizing its radios and MDT equipment. There are 53 Battalion Chief’s Units in the FDNY. A Division Chief’s Unit, like a Battalion Chief’s Unit, is a command vehicle tasked with the responsibility of delivering a Division Chief to the fire scene. Once on the fire scene, the vehicle then takes on the role of a Command Vehicle, utilizing its radios and MDT equipment. There are 9 Division Chief’s Units in the FDNY.

**Recuperation and Care Unit —** Collapse Rescue 1’s apparatus. A vehicle that is specially outfitted with equipment that will enable it to provide rehabilitation to firefighters on a fireground. Tactical Support Unit (T.S.U.) : A 4x4 vehicle equipped with generators and a variety of high intensity lights to aid firefighters during low light conditions. In addition, specialized equipment such as extrication tools and a six-person Avon boat is also carried. Thawing Unit : The Thawing Units are vehicles that carry a portable steam-generating boiler; its high-pressure steam is used to thaw frozen hydrants, [58] connections, and hoselines,[59] and to keep equipment on the fireground, such as aerial ladders, free of ice.[60] There is only one thawing unit per borough.[58]
Brush Fire Unit : A vehicle that is a four-wheel-drive, all-terrain unit used to reach hilly, remote and marshy areas to extinguish fires involving weeds, grass and other vegetation. Along with regular firefighting equipment, it carries its own water, as well as rakes, shovels, and backpack extinguishers.
Ambulance : The New York City Fire Department staffs EMT-Basic and EMT Paramedic Ambulances to provide emergency medical services to the city of New York. These are commonly referred to by the slang term bus.

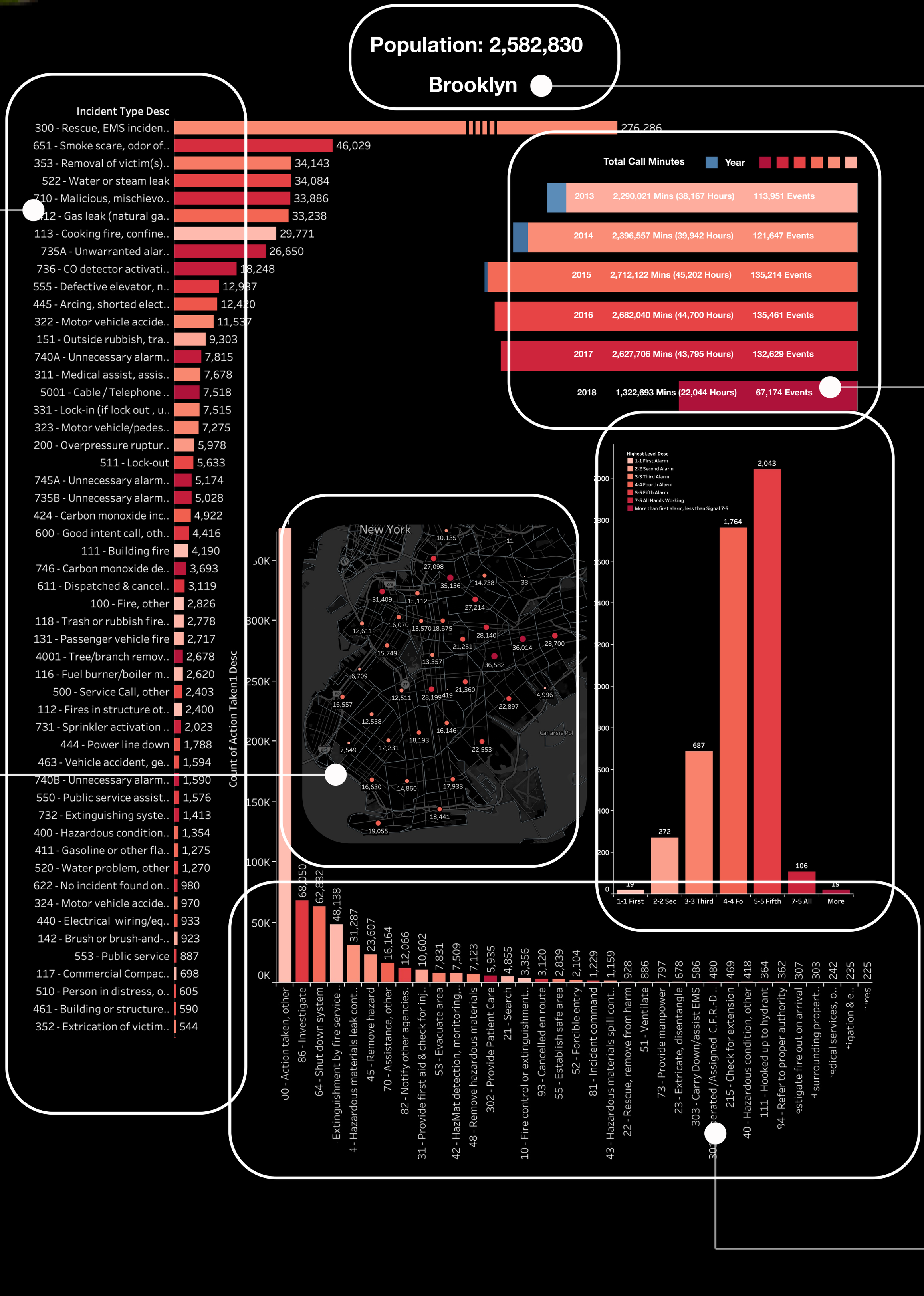
**SOC Battalion —** One of the many smaller S.O.C. Support Trucks operated by the FDNY for use at various emergencies. Haz Tac Battalion Unit is a vehicle that is assigned to an Emergency Medical Service supervisor. This Officer is the Captain of the FDNY EMS SOC unit. The Officer is trained as a Hazmat Technician and as Rescue Technician. The unit responds to speciality assignments to oversee the medical management at special assignments and the overall operation of the Haz Tac Battalion.
Haz Tac Officer’s Unit is a vehicle that is assigned to an Emergency Medical Service supervisor. This Officer is trained as a Hazmat Technician and as Rescue Technician. The unit responds to specialty assignments to oversee the medical management at special assignments. There are two units that cover the entire City of New York 24/7.

**EMS —** MERV-1. EMS MERV is a vehicle that is assigned to all major medical emergencies within its borough. The Major Emergency Response Vehicle is capable of treating multiple patients at a time.
EMS LSU is a vehicle assigned to all medical emergencies that have multiple patients. The Logistical Support Unit carries a surplus of certain medical supplies used at MCIs.
EMS Response Physician is a Fire Department EMS Medical Director who is an Emergency Physician with specialized training in Hazardous Material, Technical Rescue, and other specialized prehospital skills such as on-scene limb amputations. The Response Physician responds to major Mass Casualty Incidents or as part of the Rescue Medical Task Force for patients requiring technical rescue or prolonged extrication. There are 9 EMS response physicians throughout the city who go by the radio designation Car 5M[61] (“5 Mary Car”).

### Descriptions and Information Breakdown for Borough Charts & Visualizations

**Incident Type Description**
This block shows the type of incidents responded to by the fire department in this Borough of New York City as well as the count of how many of the calls were related to this type of incident.

**Borough Incident Map**
The map included with each set of data has the count of incidents by zip code section for each borough.



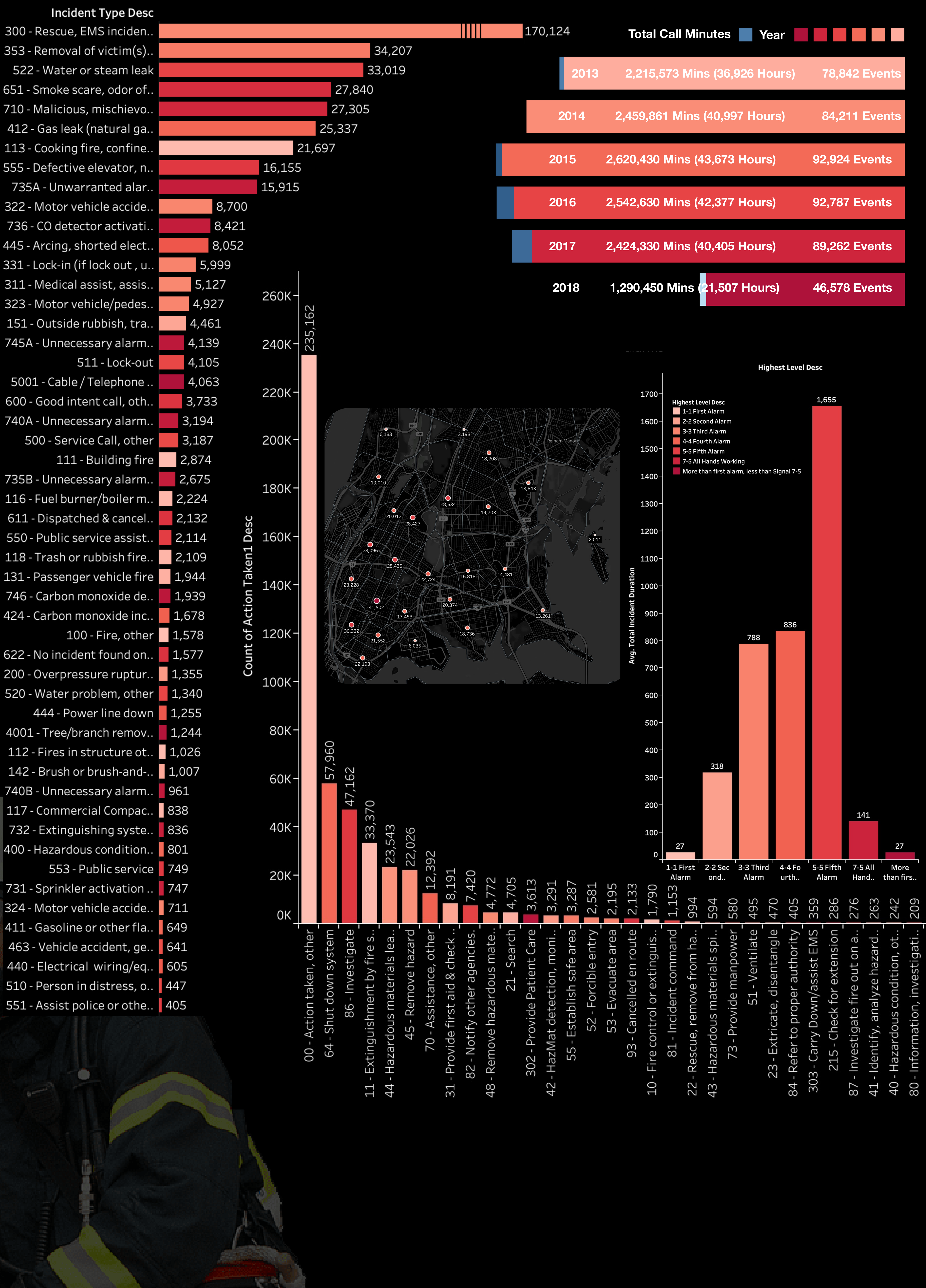

**Borough & Population**
Each of the 5 NYC boroughs are accounted for in these charts as well as a complete aggregation of all data from all boroughs. Included is also the total population for the borough.

**Incident Duration**
This block srepresents the number of incidents, events, per year and also the total sum of minutes and hours per year the fire department spent answering these events. Note for 2018 the data represents only half of the year due to reporting times.

**Action Taken Description**
This block represents the actions taken by the fire department and the amount of times these actions were taken.

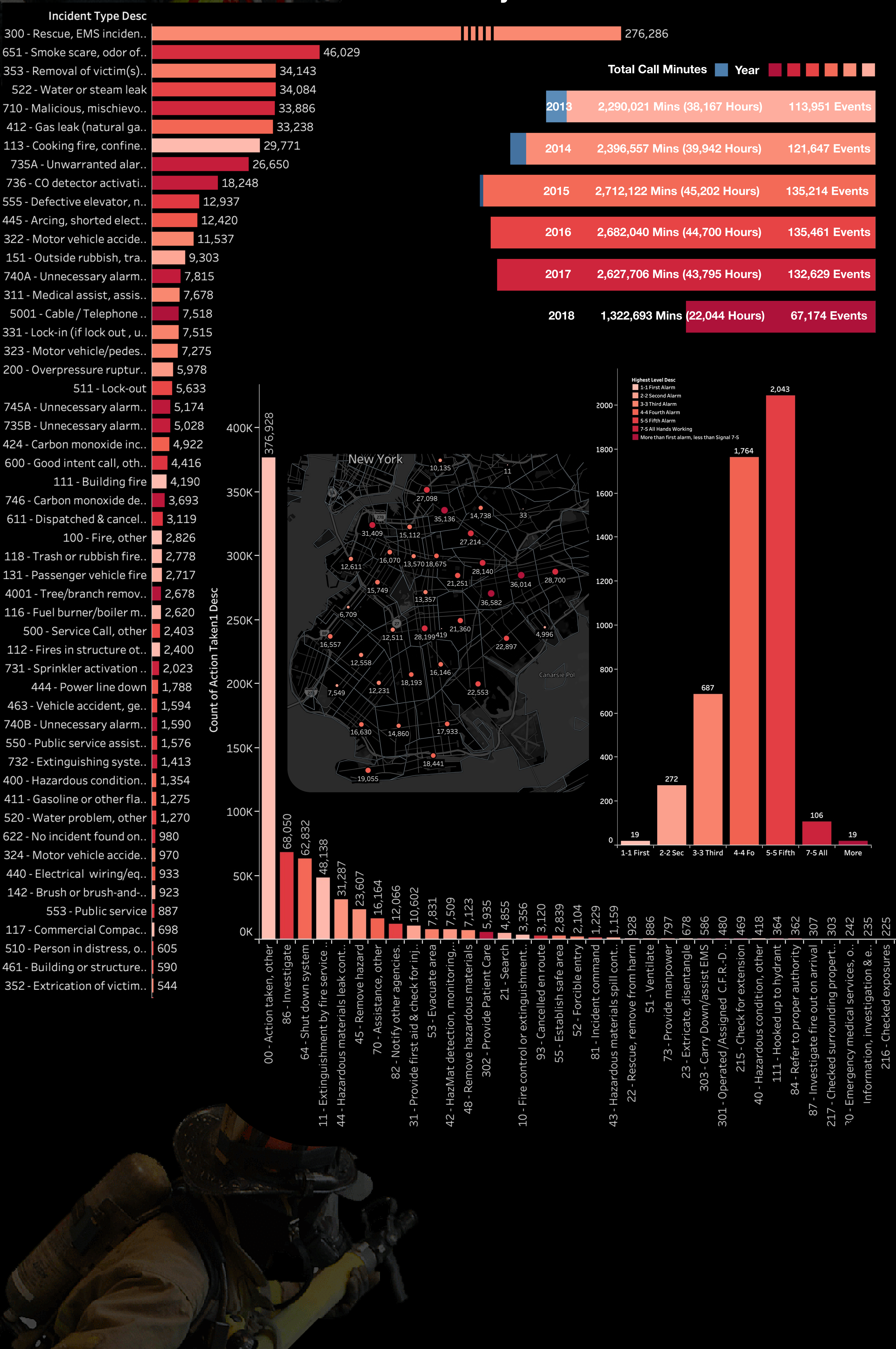
Population: 1,432,132

## The Bronx



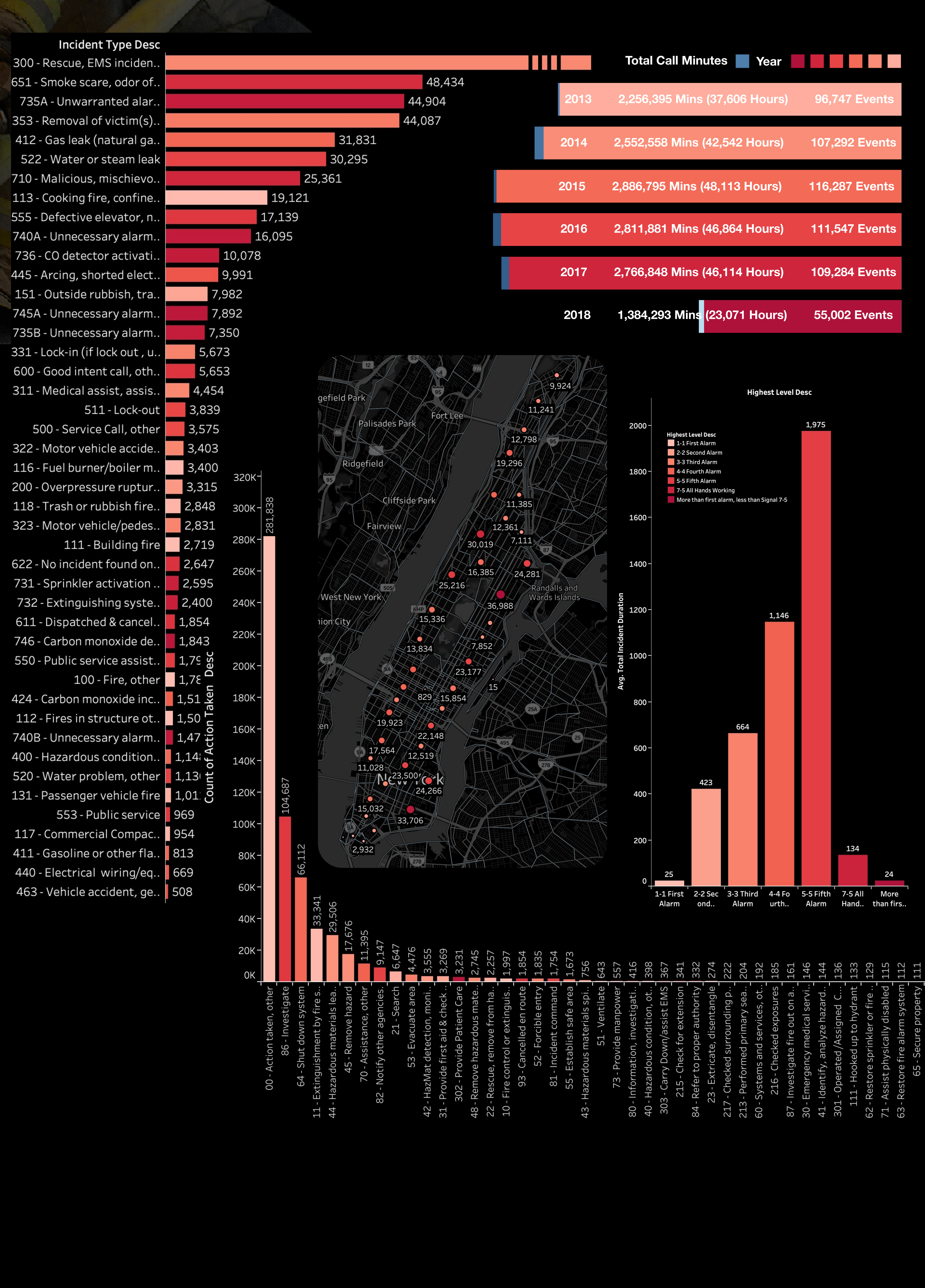

Population: 2,582,830

## Brooklyn

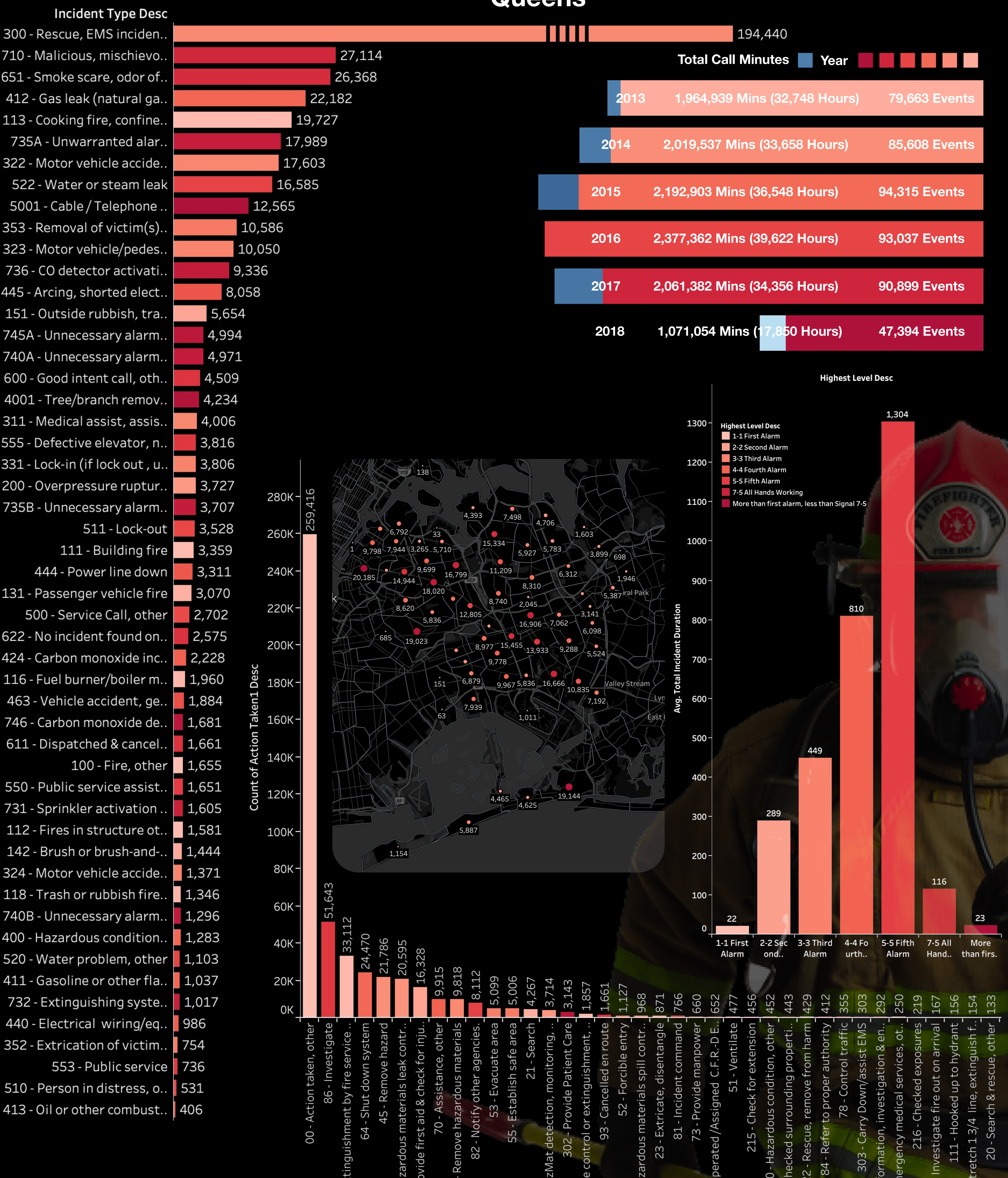



Population: 1,628,701

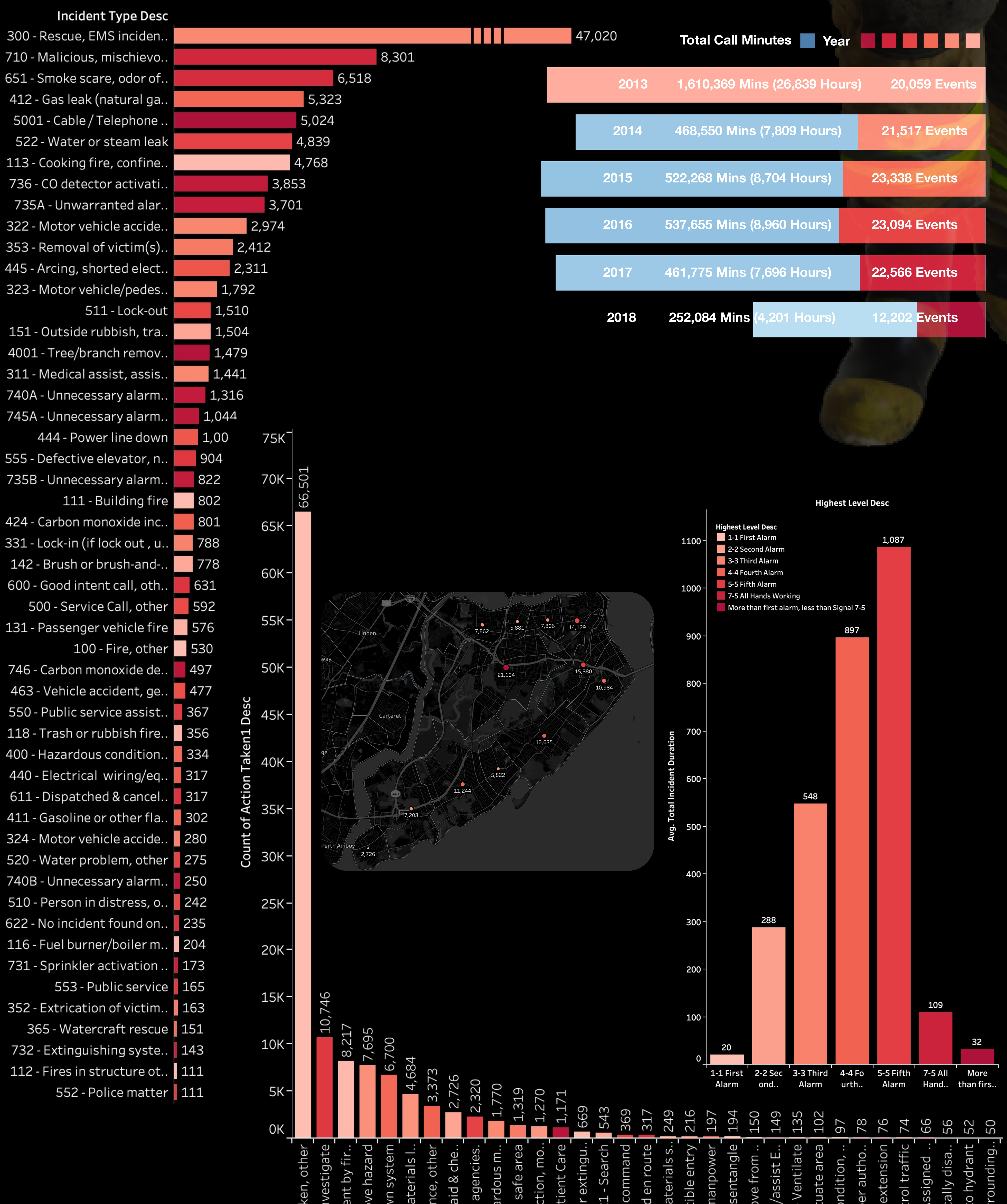
## Manhattan



## Population: 2,278,906
# Queens




## Population: 476,179
# Staten Island




## Population: 8,398,748
# All NYC Boroughs

# Machine Learning Model Training

In this project we attempted to create a prediction algorithm which would predict the type of alarm that should be called at the dispatcher's command. This would potentially increase the response speed and reduce any further damage caused during the fire or emergency the department would be responding to.

## Feature Engineering & Data Selection

**Dataset A - 'Large'** - This dataset had a large amount of rows (2.5M) but the least amount of columns (6) at the start. The reason was because of the null values being dropped from every column until only those with appropriate data remained. Also we removed any columns which would not be possible to request from a caller by the dispatcher receiving the call. This reduced the columns significantly.

**Dataset B - 'Medium'** - The feature engineering was similar but for this set we prioritize a hybrid approach keeping certain columns and as much data as possible resulting in: 9 columns and 542K rows.

**Dataset C - 'Small'** - For this last dataset we prioritized columns above all. The result was 13 columns and 10.8K rows.

## Data Exploration & Synthesis

The data in each dataset was very comparable, only diverging slightly in how the data was to be cleaned and prepared for a mathematical model.

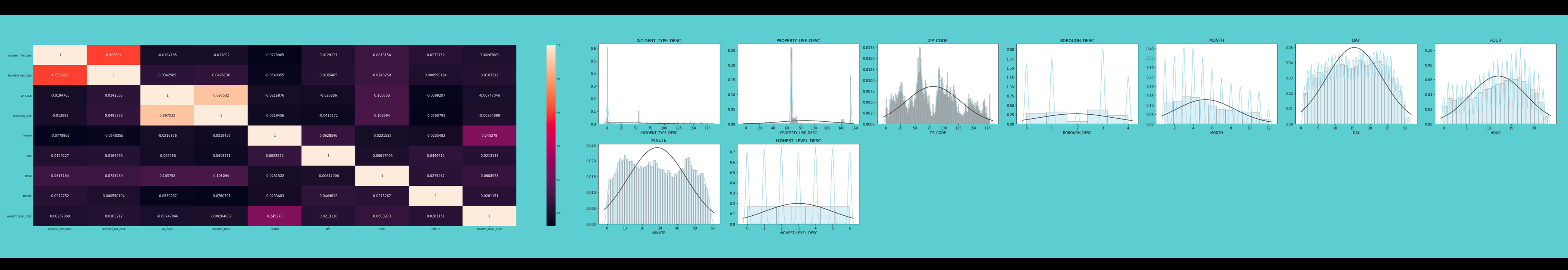
```
encoded_for_largedf = ['PROPERTY_USE_DESC',
'ZIP_CODE','INCIDENT_TYPE_DESC', 'BOROUGH_DESC']

def encode_column(df, column_to_encode):
    """
    will create a target column from categorical
/ text data labels
    """
    tc = preprocessing.LabelEncoder()
    for ce in column_to_encode:
        tc.fit(df[ce])
        df[ce] = tc.transform(df[ce])

encode_column(large_df, encoded_for_largedf )
```

Columns were categorical in nature although, many are numerical. All items were treated for labeling by using the ***sklearn preprocessing.LabelEncoder()*** function.

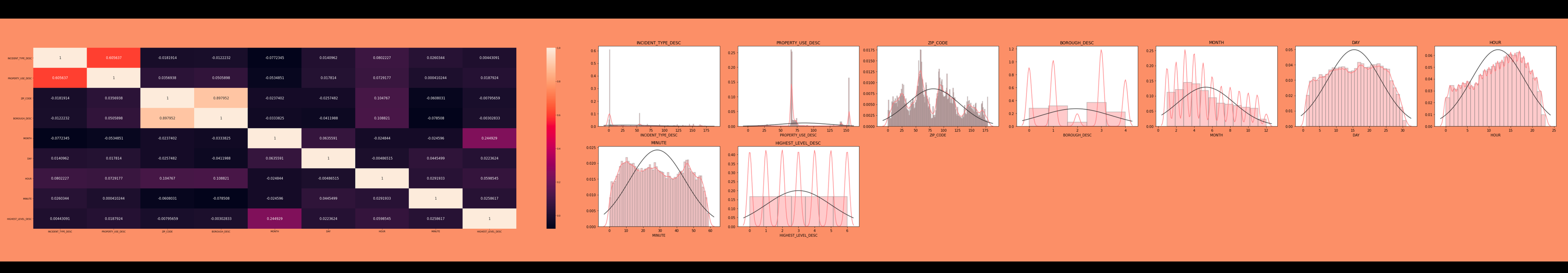
```
# creates a new csv file from the cleaned up dataset to later test the classification model
large_df.to_csv('large_df.csv')
# passes the data through the custom SMOTE function written to balance our dataset around the labels in
HIGHEST_LEVEL_DESC
large_df = balance_data_SMOTE(large_df, 'HIGHEST_LEVEL_DESC')
# samples the new dataset for 1M samples with replacement
large_df_s = large_df.sample(1000000, random_state=3, replace=True)
# creates a heatmap of all of the columns to show correlation between features on the balanced dataframe
draw_heatmap(large_df)
# draws heatmap of all columns in the synthetic sampled dataframe
draw_heatmap(large_df_s)
# shows histograms for all columns in the balanced dataframe
draw_histograms(large_df, color_c='lightblue')
# shows histograms for all columns in the synthetic dataframe for comparison
draw_histograms(large_df_s, color_c='lightcoral')
# writes the new synthetic file into a csv for use in the classification model notebook
large_df_s.to_csv('large_df_s.csv')
```

The classification algorithm's target feature would be the 'HIGHEST_LEVEL_DESC' which stores the alarm type. Once we encoded this column it was possible to see the imbalance between the classes. Because of this we used the ***imblearn*** library and passed the SMOTE function onto the datasets to balance the datasets. After this we were able to re-sample with replacement and create new datasets with balanced datasets. We took 1,000,000 synthetic samples from each dataset created by the ***SMOTE*** algorithm.

**Balanced Dataset**



**Sampled Dataset**



## Model Training & Selection

For this project we used supervised learning classification algorithms to create the prediction of the alarm to be called by a dispatcher.

The algorithms used were: ***Decision Tree Classifier***, ***Random Forest Classifier***, ***Gradient Boosting Classifier***, ***XG-Boost Classifier*** and a ***K-Nearest Neighbors Classifier***.

Below are the model training and test set results. We also passed through a test set based on the original cleaned data, before sampling, to test how well the model would generalize to 'unseen' data. The results were most favorable with our 'Small' dataset engineered to have the maximum number of features and then sampled from the SMOTE balanced set.

**To use this model in production a Random Forest would be ideal using the features and data engineering strategy implemented in the Small dataset's processing.** *(Below are explanations for the scoring and assessment of the models and datasets.)*

### Accuracy Score

Proportion of total number of predictions that were actually correct.

Categorical accuracy = 1, means the model's predictions are perfect.

**Source:**
https://peltarion.com/knowledge-center/documentation/evaluation-view/classification-loss-metrics/categorical-accuracy

### Macro F1 Score

Macro F1-score (short for macro-averaged F1 score) is used to assess the quality of problems with multiple binary labels or multiple classes.

Macro F1-score = 1 is the best value, and the worst value is 0.

**Source:**
https://peltarion.com/knowledge-center/documentation/evaluation-view/classification-loss-metrics/macro-f1-score

### Macro Precision Score

Macro-precision measures the average precision per class. It's short for macro-averaged precision.

Precision = 1 means the model's predictions are perfect, all samples classified as the positive class are truly positive.

**Source:**
https://peltarion.com/knowledge-center/documentation/evaluation-view/classification-loss-metrics/macro-precision

### Macro Recall Score

Macro-recall measures the average recall per class. It's short for macro-averaged recall.

Macro-recall = 1 means the model's predictions are perfect, all truly positive samples was predicted as the positive class.

**Source:**
https://peltarion.com/knowledge-center/documentation/evaluation-view/classification-loss-metrics/macro-recall

### Micro F1 Score

Micro F1-score (short for micro-averaged F1 score) is used to assess the quality of multi-label binary problems.

It measures the F1-score of the aggregated contributions of all classes.

If you are looking to select a model based on a balance between precision and recall, don't miss out on assessing your F1-scores!

**Source:**
https://peltarion.com/knowledge-center/documentation/evaluation-view/classification-loss-metrics/micro-f1-score

### Micro Precision Score

Micro-precision measures the precision of the aggregated contributions of all classes.

It's short for micro-averaged precision.

Precision = 1 means the model's predictions are perfect, all samples classified as the positive class are truly positive.

**Source:**
https://peltarion.com/knowledge-center/documentation/evaluation-view/classification-loss-metrics/micro-precision

### Micro Recall Score

Micro-recall measures the recall of the aggregated contributions of all classes. It's short for micro-averaged recall.

Micro-recall = 1 means the model's predictions are perfect, all truly positive samples was predicted as the positive class.

**Source:**
https://peltarion.com/knowledge-center/documentation/evaluation-view/classification-loss-metrics/micro-recall

### Run Time (secs)

Time it took our system to execute the training and test procedures.

Calculated using python's time library.

### Weighted F1 Score

Calculate metrics for each label, and find their average weighted by support (the number of true instances for each label). This alters 'macro' to account for label imbalance; it can result in an F-score that is not between precision and recall.

**Source:**
https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html

### Weighted Precision Score

Calculate metrics for each label, and find their average weighted by support (the number of true instances for each label). This alters 'macro' to account for label imbalance; it can result in an F-score that is not between precision and recall.

**Source:**
https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_score.html

### Weighted Recall Score

Calculate metrics for each label, and find their average weighted by support (the number of true instances for each label). This alters 'macro' to account for label imbalance; it can result in an F-score that is not between precision and recall.

**Source:**
https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall_score.html

**Small Dataset Model Training**

| Score | XG-Boost Classifier | Gradient Boosting Classifier | K-Neighbors Classifier | Decision Tree Classifier | Random Forest Classifier |
|---|---|---|---|---|---|
| Accuracy Score | 0.9558 | 0.4296 | 0.9999 | 1.0000 | 1.0000 |
| Macro F1 Score | 0.9552 | 0.4239 | 0.9999 | 1.0000 | 1.0000 |
| Macro Precision Score | 0.9561 | 0.4249 | 0.9999 | 1.0000 | 1.0000 |
| Macro Recall Score | 0.9561 | 0.43 | 0.9999 | 1.0000 | 1.0000 |
| Micro F1 Score | 0.9558 | 0.4296 | 0.9999 | 1.0000 | 1.0000 |
| Micro Precision Score | 0.9558 | 0.4296 | 0.9999 | 1.0000 | 1.0000 |
| Micro Recall Score | 0.9558 | 0.4296 | 0.9999 | 1.0000 | 1.0000 |
| Run Time(secs) | 189.251 | 100.715 | 12.039 | 6.1780 | 37.1620 |
| Weighted F1 Score | 0.9551 | 0.4237 | 0.9999 | 1.0000 | 1.0000 |
| Weighted Precision Score | 0.9562 | 0.425 | 0.9999 | 1.0000 | 1.0000 |
| Weighted Recall Score | 0.9558 | 0.4296 | 0.9999 | 1.0000 | 1.0000 |

**Small Dataset Model 'Unseen' Test Data**

| Score | XG-Boost Classifier | Gradient Boosting Classifier | K-Neighbors Classifier | Decision Tree Classifier | Random Forest Classifier |
|---|---|---|---|---|---|
| Accuracy Score | 0.8864 | 0.3655 | 0.9998 | 0.9999 | 0.9999 |
| Macro F1 Score | 0.8719 | 0.1794 | 0.9998 | 0.9998 | 0.9998 |
| Macro Precision Score | 0.8307 | 0.1921 | 0.9997 | 0.9997 | 0.9997 |
| Macro Recall Score | 0.9257 | 0.349 | 0.9999 | 0.9999 | 0.9999 |
| Micro F1 Score | 0.8864 | 0.3655 | 0.9998 | 0.9999 | 0.9999 |
| Micro Precision Score | 0.8864 | 0.3655 | 0.9998 | 0.9999 | 0.9999 |
| Micro Recall Score | 0.8864 | 0.3655 | 0.9998 | 0.9999 | 0.9999 |
| Weighted F1 Score | 0.8853 | 0.4343 | 0.9998 | 0.9999 | 0.9999 |
| Weighted Precision Score | 0.8882 | 0.5687 | 0.9998 | 0.9999 | 0.9999 |
| Weighted Recall Score | 0.8864 | 0.3655 | 0.9998 | 0.9999 | 0.9999 |

**Medium Dataset Model Training**

| Score | XG-Boost Classifier | Gradient Boosting Classifier | K-Neighbors Classifier | Decision Tree Classifier | Random Forest Classifier |
|---|---|---|---|---|---|
| Accuracy Score | 0.9121 | 0.5138 | 0.7674 | 0.8965 | 0.9395 |
| Macro F1 Score | 0.9116 | 0.4968 | 0.7547 | 0.8961 | 0.9396 |
| Macro Precision Score | 0.9117 | 0.5108 | 0.7525 | 0.8956 | 0.9411 |
| Macro Recall Score | 0.9123 | 0.514 | 0.7681 | 0.8968 | 0.9397 |
| Micro F1 Score | 0.9121 | 0.5138 | 0.7674 | 0.8965 | 0.9395 |
| Micro Precision Score | 0.9121 | 0.5138 | 0.7674 | 0.8965 | 0.9395 |
| Micro Recall Score | 0.9121 | 0.5138 | 0.7674 | 0.8965 | 0.9395 |
| Run Time(secs) | 33.689 | 23.375 | 2.704 | 2.4950 | 8.9120 |
| Weighted F1 Score | 0.9114 | 0.4968 | 0.7541 | 0.8958 | 0.9394 |
| Weighted Precision Score | 0.9116 | 0.5112 | 0.7522 | 0.8955 | 0.9410 |
| Weighted Recall Score | 0.9121 | 0.5138 | 0.7674 | 0.8965 | 0.9395 |

**Medium Dataset Model 'Unseen' Test Data**

| Score | XG-Boost Classifier | Gradient Boosting Classifier | K-Neighbors Classifier | Decision Tree Classifier | Random Forest Classifier |
|---|---|---|---|---|---|
| Accuracy Score | 0.7832 | 0.5963 | 0.3432 | 0.7447 | 0.7569 |
| Macro F1 Score | 0.4982 | 0.2166 | 0.1249 | 0.3708 | 0.7117 |
| Macro Precision Score | 0.3914 | 0.2331 | 0.1657 | 0.2928 | 0.6220 |
| Macro Recall Score | 0.8276 | 0.476 | 0.7904 | 0.8005 | 0.8871 |
| Micro F1 Score | 0.7832 | 0.5963 | 0.3432 | 0.7447 | 0.7569 |
| Micro Precision Score | 0.7832 | 0.5963 | 0.3432 | 0.7447 | 0.7569 |
| Micro Recall Score | 0.7832 | 0.5963 | 0.3432 | 0.7447 | 0.7569 |
| Weighted F1 Score | 0.811 | 0.6762 | 0.4385 | 0.7861 | 0.7969 |
| Weighted Precision Score | 0.8529 | 0.8815 | 0.8351 | 0.8571 | 0.8701 |
| Weighted Recall Score | 0.7832 | 0.5963 | 0.3432 | 0.7447 | 0.7569 |

**Large Dataset Model Training**

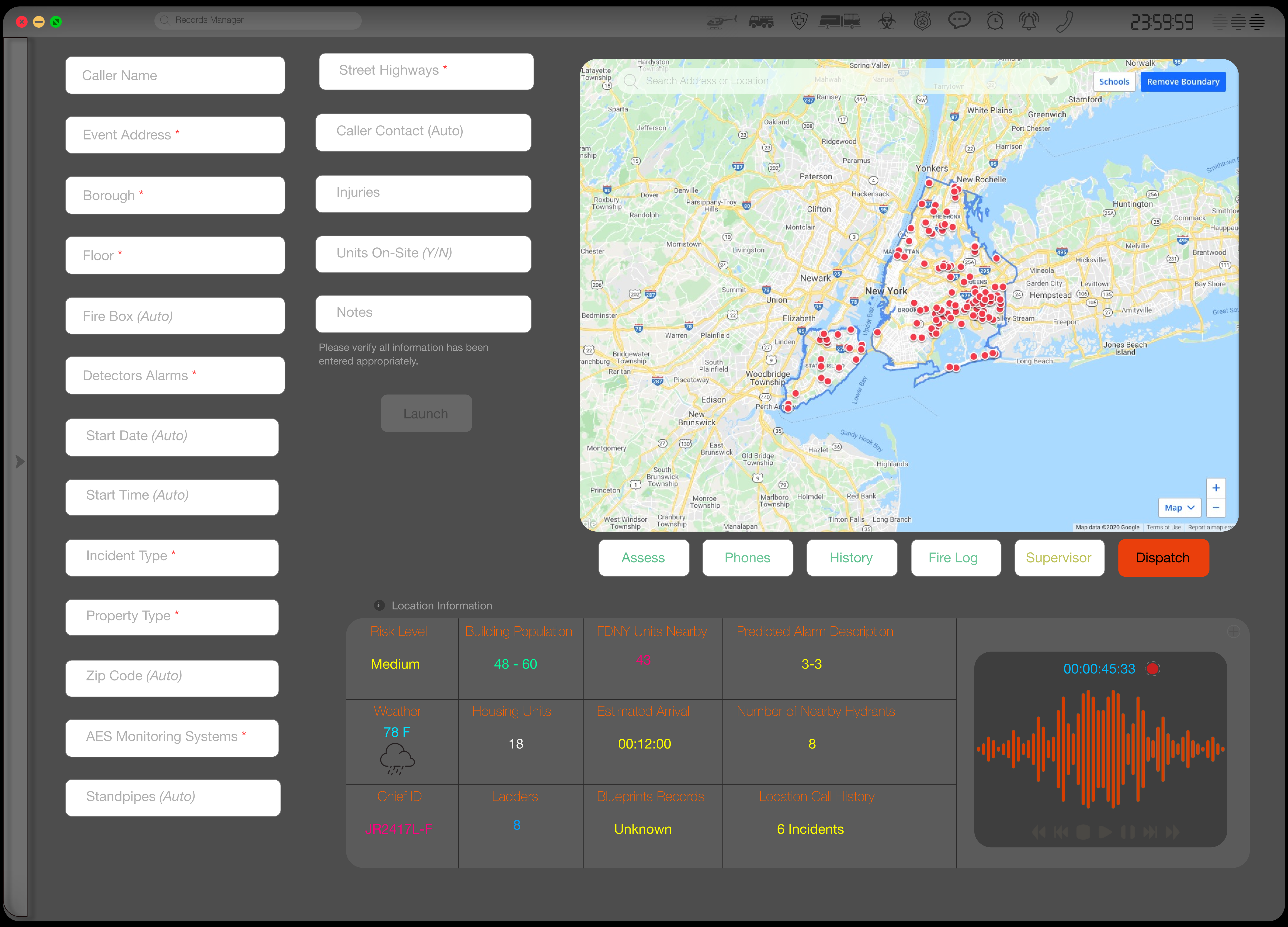
| Score | XG-Boost Classifier | Gradient Boosting Classifier | K-Neighbors Classifier | Decision Tree Classifier | Random Forest Classifier |
|---|---|---|---|---|---|
| Accuracy Score | 0.8496 | 0.5016 | 0.8509 | 0.8731 | 0.9151 |
| Macro F1 Score | 0.8454 | 0.4841 | 0.8461 | 0.8725 | 0.9138 |
| Macro Precision Score | 0.8537 | 0.4974 | 0.8471 | 0.8723 | 0.9205 |
| Macro Recall Score | 0.8489 | 0.5006 | 0.8508 | 0.8732 | 0.9149 |
| Micro F1 Score | 0.8496 | 0.5016 | 0.8509 | 0.8731 | 0.9151 |
| Micro Precision Score | 0.8496 | 0.5016 | 0.8509 | 0.8731 | 0.9151 |
| Micro Recall Score | 0.8496 | 0.5016 | 0.8509 | 0.8731 | 0.9151 |
| Run Time(secs) | 27.331 | 15.228 | 26.422 | 1.3850 | 6.3900 |
| Weighted F1 Score | 0.8456 | 0.4848 | 0.8459 | 0.8723 | 0.9136 |
| Weighted Precision Score | 0.8534 | 0.4976 | 0.8467 | 0.8720 | 0.9200 |
| Weighted Recall Score | 0.8496 | 0.5016 | 0.8509 | 0.8731 | 0.9151 |

**Large Dataset Model 'Unseen' Test Data**

| Score | XG-Boost Classifier | Gradient Boosting Classifier | K-Neighbors Classifier | Decision Tree Classifier | Random Forest Classifier |
|---|---|---|---|---|---|
| Accuracy Score | 0.6428 | 0.6047 | 0.5153 | 0.6806 | 0.6417 |
| Macro F1 Score | 0.2535 | 0.1746 | 0.1554 | 0.2178 | 0.3280 |
| Macro Precision Score | 0.2231 | 0.1808 | 0.1713 | 0.1966 | 0.2683 |
| Macro Recall Score | 0.8102 | 0.4595 | 0.8499 | 0.7679 | 0.8799 |
| Micro F1 Score | 0.6428 | 0.6047 | 0.5153 | 0.6806 | 0.6417 |
| Micro Precision Score | 0.6428 | 0.6047 | 0.5153 | 0.6806 | 0.6417 |
| Micro Recall Score | 0.6428 | 0.6047 | 0.5153 | 0.6806 | 0.6417 |
| Weighted F1 Score | 0.7174 | 0.6877 | 0.6125 | 0.7472 | 0.7159 |
| Weighted Precision Score | 0.8896 | 0.8879 | 0.8731 | 0.8763 | 0.8924 |
| Weighted Recall Score | 0.6428 | 0.6047 | 0.5153 | 0.6806 | 0.6417 |

# Conclusion

This chart represents how each model performed with each of the datasets fed through the individual algorithms. The ‘Unseen’ data is from the original data before it underwent the SMOTE and re-sampling procedure to create synthetic variables. The training dataset was created using synthetic values. **The conclusion to our model creation, is that the preservation of as many columns as possible creates a much more robust algorithm in predicting original data versus removal of any of these features.**




Above is a fictitious system based around our chosen features and providing what was thought to be relevant during a call and inclusive of our 'Predicted Alarm Description' addition.

Fore more information please visit my website:

www.eugeniosprojects.com