University of Toronto
Department of Mechanical and Industrial Engineering

MIE368 Analytics in Action
Toronto Fire Trucks Optimization
Final Report

Team 08

Student Name	Student Number
Muchen Liu	1006732145
Yikai Wang	1007032550
Yuhan Jiang	1007092441
Jerry Chen	1006944899

1. Introduction
Fire incident is one of the most common and dangerous disasters that threaten people's lives.[1] There were 124 and 115 fire-related deaths of Ontarians in 2021 and 2020 respectively. The deaths of people caused by fire-incident-related issues are increasing rapidly every year. [2] To keep the people safe and reduce the casualties caused by the incidents, in every city, the government always spends huge amounts of human and financial resources on fire stations and fire apparatus.[3] To help the government reduce the expense of fire equipment and rescue more people in cases of fires, the team decided to design a model to allocate fire trucks more effectively in the City of Toronto, through comparative analysis and optimizations. To satisfy this purpose, the project focused on making a Python Based optimization model on finding the minimum number of the required fire trucks every day for each fire station, that can cover at least 90% of the fire incidents that will potentially happen in the City of Toronto. All coding parts for this project are done on Google Collab.

2. Data
There are three datasets to be used in the project, which are all imported from the Toronto Open Data Catalogue. The first dataset focuses on the fire incident data which includes the fire incidents that happened in Toronto from 2016 to 2019[4]. Since the limitations on the Copyrights of the data source, the most recent fire incident data is not available to use in the project. This data includes the coordinates of the fire incidents based on the latitude and longitude of each fire incident, and other relative information including Civilian Casualties, Estimated Dollar Loss, etc. The second dataset focuses on the fire station locations, which contains the locations of the fire station in the City of Toronto with the information on each fire station including the latitude and longitude of the stations [5]. The third dataset focuses on the fire service run areas in Toronto, which shows the geographical areas for the service and coverage areas of the Toronto Fire Service. The team uses this data to determine the responsible working area of each fire station [6]. Each run area is a geographical area that includes several blocks of the City of Toronto, which naturally allows the team to merge the fire incidents into various areas for analysis.

After importing the data into the Collaborative Projects (Google Collab) (Appendix A), the team first dropped the null values of the datasets. Then, the team converted all the geometry values in JSON formats into Longitude and Latitude as float type and stored them in the corresponding datasets as dataframes. The index of each data frame was reset for further operation. The team then removed some data that would not be used in the optimization model and only kept a few useful data in the dataframes after the EDA(see Table 1). After the operations on the data mentioned above, there are 92 fire stations and 84 fire run areas in total.

Table 1: Columns Kept in DataFrames after EDA

DataFrame	Columns Kept after EDA
Station	'Latitude', 'Longitude'
Centroid (Fire Run Areas)	'Latitude', 'Longitude'
Incident(2016-2019)	'Latitude', 'Longitude', 'Number_of_responding_apparatus', 'TFS_Alarm_Time'

3. Methods

The team used the Pearson correlation to justify the randomness of the locations of fire incidents. The fire incidents were proven to be all randomly distributed with a Pearson correlation coefficient of 0.028 in Longitudes and 0.0084 in Latitude, between the coordinates of incidents of 2016 and 2017 respectively (Figure 1). Using graphical analysis, the team plotted the fire incidents of 2016 and 2017 on two scatter plots corresponding to the longitudes and latitudes of the incidents in two years respectively (Figure 2). Considering the complexity and the problem of overfitting, the team decided to focus on the fire incidents happening in each run area and use the coordinates of the centroid of each run area, rather than focusing on each single fire incident(Figure 3). This is similar to the method of clustering in machine learning.

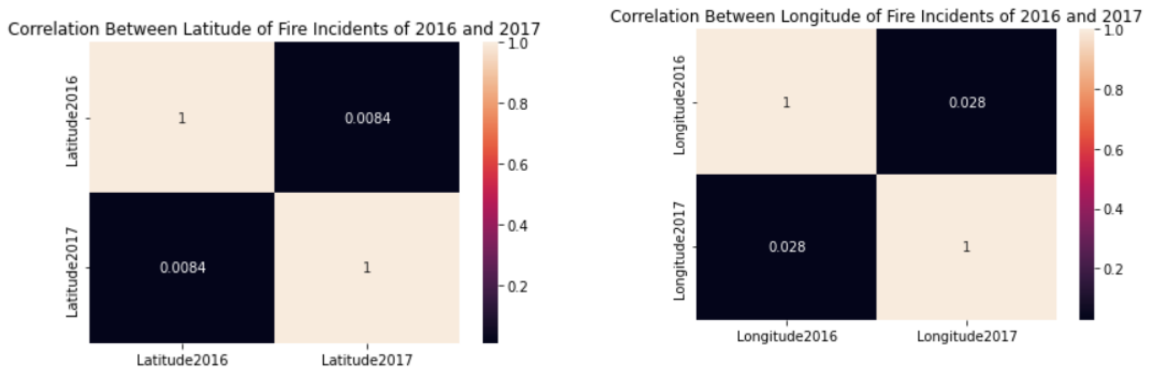


Figure 1: Pearson-correlation Shows the Randomness of the Locations of Fire Incidents

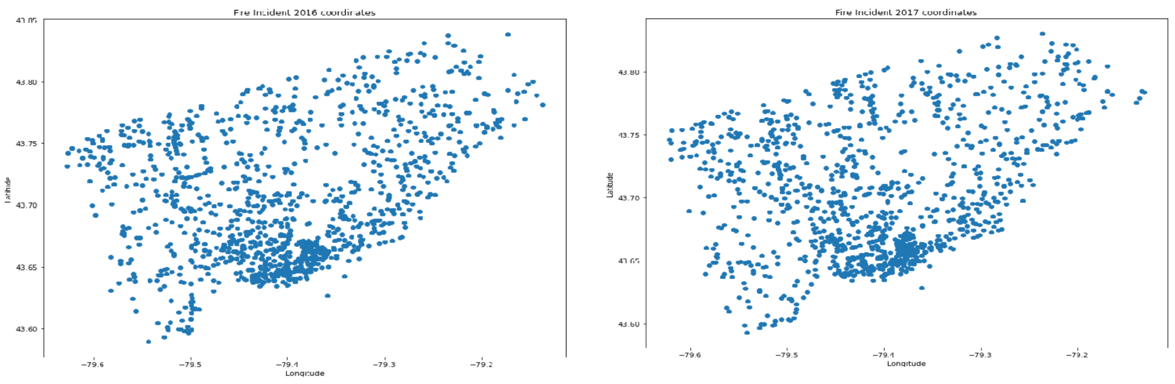
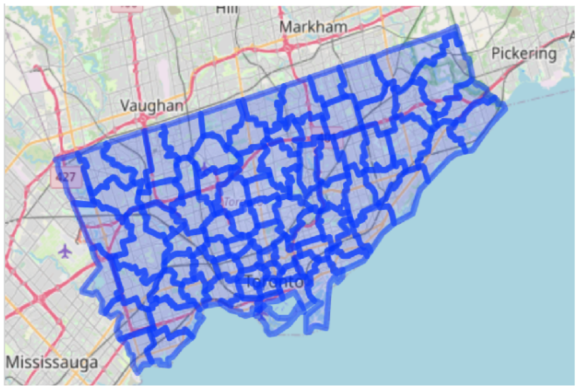


Figure 2: Scatter plots for fire incidents in 2016 and 2017



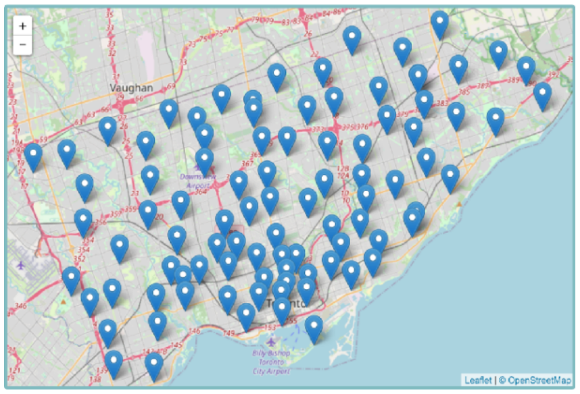
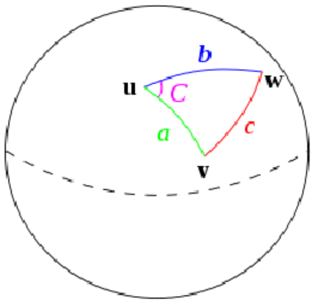


Figure 3: Run Areas with the Coordinates of their Centroids [6]

To allocate fire incidents to their corresponding fire run areas, the team calculated the distance between each fire incident and each centroid of the fire areas and related each incident to its nearest centroid of fire areas. Since the earth is approximately a sphere with a radius of 6357 km [7], the team used the Haversine Formula [8] (see Figure 3) to calculate the distance of two points on the earth with their latitudes and longitudes (Appendix B). Then, the team calculated the total number of fire apparatus needed in each area every day.



$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

Figure 4: Haversine Formula Used in Distance Calculation [8]

To identify which fire station should dispatch fire trucks to a certain fire run area, the team first related each fire station to its nearest fire run areas with no limits on the distances and then related all the fire stations within 2 km to each fire run area, since the Average Speed for the fire trucks in the City of Toronto is approximately 60 kilometers per hour, and the required response time for each dispatched fire truck to its assigned destinations is in 2 minutes.[9] This was realized by calculating the distance between the coordinate of each fire station and the coordinate of the centroid of each run area using the Haversine Formula (Figure 3).

For the optimization model (Appendix C), the purpose is to minimize the total number of fire trucks that can meet the daily demand in each run area. The team first set two sets of constant parameters. According to the Exploratory Data Analysis (EDA), the team found out that there were 92 fire stations in the City of Toronto. For denotations, each station was represented by integer ‘i’ in orders, ranging from 0 to 91. Furthermore, there were 84 fire run areas, which means that there were 84 coordinates for the centroids of the fire run areas in the City of Toronto. Each fire run area was denoted by 'j' in orders, ranging from 0 to 83.

The first parameter was set for determining whether each station 'i' should dispatch a certain number of fire trucks to each fire run area 'j'. Since the parameter was used for making decisions, the parameter was set to be binary and denoted as b_i. The values for b_{ij} were dependent on the fire trucks dispatch requirements mentioned previously. Since, b_{ij} is a binary variable, the range for b_{ij} is from 0 to 1, in integer. If a station was required to dispatch the fire trucks to a fire run area, b_{ij} would be set to 1, otherwise b_{ij} stayed at 0.

The second parameter was representing the number of the demanded fire engines in fire run area 'j' on each day in a year. The team was using the data in 2016 for building the optimization model; there are 366 days in 2016 since the year 2016 was a leap year. Therefore, each day was denoted as integer 'k', where k is within the range from 0 to 365. Then, the parameter was denoted as D_{jk}. The values for D_{jk} were defined by summing up the required number of the responding apparatuses for all the fire incidents that were classified to the Fire Run Service Area 'j' on the day 'k'. Since the data of 'Number of Responding Apparatus' recorded on the Fire Incident Data shows the total number of the apparatus responds to the corresponding incident, including the Hydraulic Cutters, Extrication Stabilization Struts, Gas Powered Hydraulic Pump, Low-Pressure Airbag, Water/foam extinguisher, Stihl roof chainsaw, Low-Pressure Airbag Hoses and Manifold, Jaws of Life, K-12 Rescue Saw and the Scaling Ladder [10]. The team assumed that one fire truck could carry all ten types of responding apparatuses. Therefore, the team assumed that the final values for D_{jk} were $\frac{1}{10}$ of the total number of responding apparatuses needed in each fire run area in one day.

After setting the constant parameters for the optimization model, the team then defined the decision variable as x_{ij}, as an integer variable, showing the total number of fire trucks assigned from station 'i' to the fire run area 'j'.

Then, the team set up the Objective function of the optimization model, by minimizing the sum of the number of fire trucks assigned from each station 'i' to fire run area 'j', which is minimizing the sum of the decision variables. Correspondingly, the team set up the constraint to the objective function in two parts. The first constraint to the model was that the total number of fire trucks dispatched to each fire run area 'j' from all the fire stations should be greater or equal to the demanded number of fire trucks in fire run area j on every day k (D_{jk}).
The second constraint was on the nonnegativity of the values for the decision variables, which showed that the number of fire trucks dispatched from station 'i' to fire run area 'j' should be greater or equal to 0, for real satisfaction.

Below is a summary of the constant parameters, decision variables, and optimization model.

Constant Parameters:
b_{ij}: A binary variable determining whether station i should dispatch fire trucks to centroid j (j = {0,1}), determined by the distance between centroids and stations.
D_{jk}: An integer variable shows the number of the demanded fire engines in centroid j on day k (k = 1, 2, 3, …, 366).

Decision Variable:
x_{ij}: An integer variable shows the number of fire engines assigned from station i (i = 1, 2, 3, …, 92) to centroid j (j = 1, 2, 3, …, 84).

Model:

$$Objective: Minimize(\sum_{i}^{92}\sum_{j}^{84} x_{ij}),\ \{i = 1, 2, 3, \ldots, 92\}, \{j = 1, 2, 3, \ldots, 84\}$$

Subject to: $\sum_{i}^{92} x_{ij}b_{ij} \geq D_{jk}$ for ∀ k, {k=1, 2, 3, …, 365}

$x_{ij} \geq 0$ (nonnegativity)

After the optimization process, the team used fire incident data for different years from the dataset and calculated the percentage for coverage of the incidents of the years respectively. Since the team used the data from fire incidents in 2016 for the optimization model, the data from fire incidents from 2017 to 2019 were applied for scoring in model evaluation. The scores of the model were calculated by the percentage of demands met in each run area in 2017-2019 on every day (Appendix E). For the data used for evaluation, in each year, if the sum of the total number of fire trucks dispatched to each fire run area 'j' from all the fire stations was greater or equal to the demanded number of fire trucks in fire run area j on every day k, it was considered as the station could cover its responsible fire incidents, otherwise, it could not.

4. Results

After applying the optimization model in Google Collab with the data processed in Exploratory Data Analysis, the optimal total number of fire trucks needed for full coverage of incidents that happened in 2016 was 157 fire trucks (Appendix C). On the other hand, 24 stations can be removed since they did not need to store any fire trucks in the station (Appendix D). By removing the stations and applying the optimization fire trucks to each station for scoring the coverage for the fire incidents in 2017-2019, 92.84% of fire incidents in 2017 could be covered, 91.22% of fire incidents in 2018 could be covered, and 94.62% of fire incidents in 2019 can be covered (Appendix E). An average score of 92.89% of fire incidents in 2017-2019 could be covered, and it met the requirements of the team's purpose of more than 90% of fire incidents could be covered by applying the model of the optimization.

5. Discussion

After the optimization of the number of fire trucks in each station, the result showed that the model was efficient in use. In the optimization model, the team removed 24 stations in the City of Toronto since they did not need to store any fire trucks in the station, according to the optimization model. Keeping those stations would reduce the efficiency of the optimization.

To justify the efficiency, the team added 1 extra fire truck in those removed fire stations, to keep the occurrence of the fire stations in their designated areas and compare the coverage and the total number of fire trucks needed in the City of Toronto. After revising the optimization model, the new average score for coverage was 94.49%, which improved by 1.59%. However, the new total number of fire trucks needed in the City of Toronto was 181, which was 24 more than the original optimization model. Though the percentage of coverage was increased by 1.59%, it was a huge amount of expense for the government to keep 24 extra fire stations with only 1 fire truck in it.

The team also tried to concatenate the fire incident data in two years to optimize the model. By the newly concatenate data, the scores with the remaining fire incident data in the rest two years (e.g. concatenate the data in 2016 and 2018, score with the data in 2017 and 2019 for coverage, and the total number of fire trucks). The detailed coverage and the total number of fire trucks are shown in the table below (see Table 2).

Table 2: Results for optimizations after concatenation

Data for Optimization	2016 & 2017	2016 & 2018	2016 & 2019
Data for Scoring	2018 & 2019	2017 & 2019	2017 & 2018
Total Number of Fire Trucks	236	220	212
Number of Fire Stations that Can be Removed	22	22	23
Number of Fire Stations Needed	70	70	69
Percentage for Coverage	96.45%	96.90%	96.45%

However, though the coverage for the optimization increases, the total number of fire trucks needed each year was significantly more than the number of fire trucks got from the original model. It could also form a huge amount of expense for the government in purchasing, maintaining, and using the extra number of fire trucks.

Obviously, these methods were inefficient and impractical, and other new measures were needed, which showed that the method was not as efficient as the original method the team used for optimizations. On the other hand, the model significantly reduced problems of overfitting. The team classified the fire incidents to the nearest fire service areas and used the number of apparatuses needed in each fire service area for optimizations. The model could meet the general daily requirements of fire incidents to the max extent.

Although the optimization model could cover 92.89% of fire incidents from 2017 to 2019, there were still 7.11% of fire incidents that were not covered. There are various reasons why the optimization model cannot cover up to 100% of the fire incidents from 2016 to 2019. The first reason is that the fire incidents were generated randomly with uncertainties, the coverage for incidents in one year could not be a good reflection of the coverage for multiple years. Furthermore, the team was assuming that there was only one type of fire trucks provided in all stations, while the reality is that each station will contain several types of fire trucks with various levels of abilities for encountering fire incidents. For example, a fire truck with a high-pressure water pump performs stronger in solving fire incidents, compared to a fire truck with a low-pressure water pump [11]. When considering the level of performance of the fire trucks with different apparatuses, the percentage of coverage for the incidents can be improved. Also, the fire incident dataset, the dataset contains all of the fire incidents, including the miscalled fire incidents. Therefore, in reality, the total number of fire incidents should be less than the ones shown on the datasets, which also brought down the total percentage for coverage of the fire incidents by the stations.

6. Conclusions and Future Directions

In conclusion, the total number of fire trucks needed in the City of Toronto for covering all fire incidents in 2016 was 157 fire trucks, where 24 stations did not need to put any fire trucks and could be eliminated. The result of the optimization model can be applied to most situations and maximize the efficiency of fire truck distributions in the City of Toronto.

For future improvement, since the dataset obtained from Toronto Open Data Catalogue for the optimization model was only focused on the years from 2016 to 2019, the team will try to gain more recent data (eg. data in 2020 or 2021) up to present for optimizations once the data sources become available to the public for use. The team will evaluate the current optimization model with the data up to the present for scoring and defining new strategies in optimizations for maximizing the coverage of the fire incidents and minimizing the total number of fire trucks in the City of Toronto. Furthermore, the team may try to improve the accuracy of the model using different methods. Currently, the team eliminated 24 stations in the City of Toronto since they did not contribute to the fire coverage. The team will analyze the fire incidents to find the busiest areas of fire incidents and add a few new stations with some fire trucks among them for releasing the workload for the current fire stations.

Another alternative approach will be adding different weights on the fire trucks of different types in the optimization model since the functions of fire trucks of different types are not the same. Besides, the team will also consider additional factors combined with real situations and try to do the optimizations with different objectives and constraints, such as minimizing the total distances traveled by the fire trucks or minimizing the total time spent by the fire trucks in dealing with all fire incidents.

References

[1] "Fire disasters in Canada," *The Canadian Encyclopedia*. [Online]. Available: https://www.thecanadianencyclopedia.ca/en/article/fire-disasters#. [Accessed: 02-Oct-2022].

[2] CBC news, "102 Ontarians have died in fires this year and the winter months pose more risk, officials say | CBC news," *CBCnews*, 11-Oct-2022. [Online]. Available: https://www.cbc.ca/news/canada/toronto/ontario-fire-update-1.6612770. [Accessed: 04-Dec-2022].

[3] "Municipal Fire Services in Canada - fraser institute." [Online]. Available: https://www.fraserinstitute.org/sites/default/files/municipal-fire-services-in-canada.pdf. [Accessed: 05-Dec-2022].

[4] "Open data dataset," City of Toronto Open Data Portal. [Online]. Available: https://open.toronto.ca/dataset/fire-incidents/. [Accessed: 03-Nov-2022].

[5] "Open data dataset," City of Toronto Open Data Portal. [Online]. Available: https://open.toronto.ca/dataset/fire-station-locations/. [Accessed: 03-Nov-2022].

[6] "Open data dataset," City of Toronto Open Data Portal. [Online]. Available: https://open.toronto.ca/dataset/toronto-fire-services-run-areas/. [Accessed: 28-Nov-2022].

[7] NASA, "Imagine the universe!," *NASA*. [Online]. Available: https://imagine.gsfc.nasa.gov/features/cosmic/. [Accessed: 04-Dec-2022].

[8] C. Veness, "Movable type scripts," *GIS FAQ Q5.1: Great circle distance between 2 points*. [Online]. Available: http://www.movable-type.co.uk/scripts/gis-faq-5.1.html. [Accessed: 04-Dec-2022].

[9] W. Yu, Y. Chen, Z. Chen, Z. Xia, and Q. Zhou, "Service area delimitation of fire stations with fire risk analysis: Implementation and case study," *International journal of environmental research and public health*, 19-Mar-2020. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7143634/. [Accessed: 04-Dec-2022].

[10] Webmaster, "Trucks, apparatus and equipment," *City of Guelph*, 02-Jul-2020. [Online]. Available: https://guelph.ca/living/emergency-services/fire-service/station-locations/trucks-apparatus/. [Accessed: 04-Dec-2022].

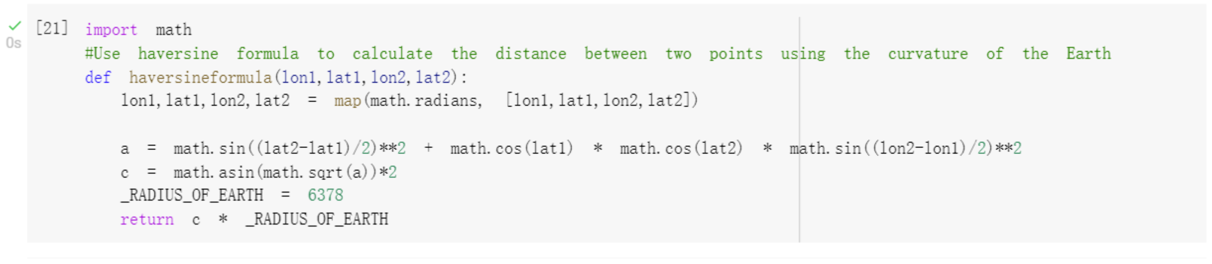
[11] K. Lepperd, "Fire fighting high pressure water pumps," Absolute Water Pumps, 06-Jan-2021. [Online]. Available: https://www.absolutewaterpumps.com/blog/fire-fighting-high-pressure-water-pumps/. [Accessed: 07-Dec-2022].

Appendix

Appendix A: Full Code Link

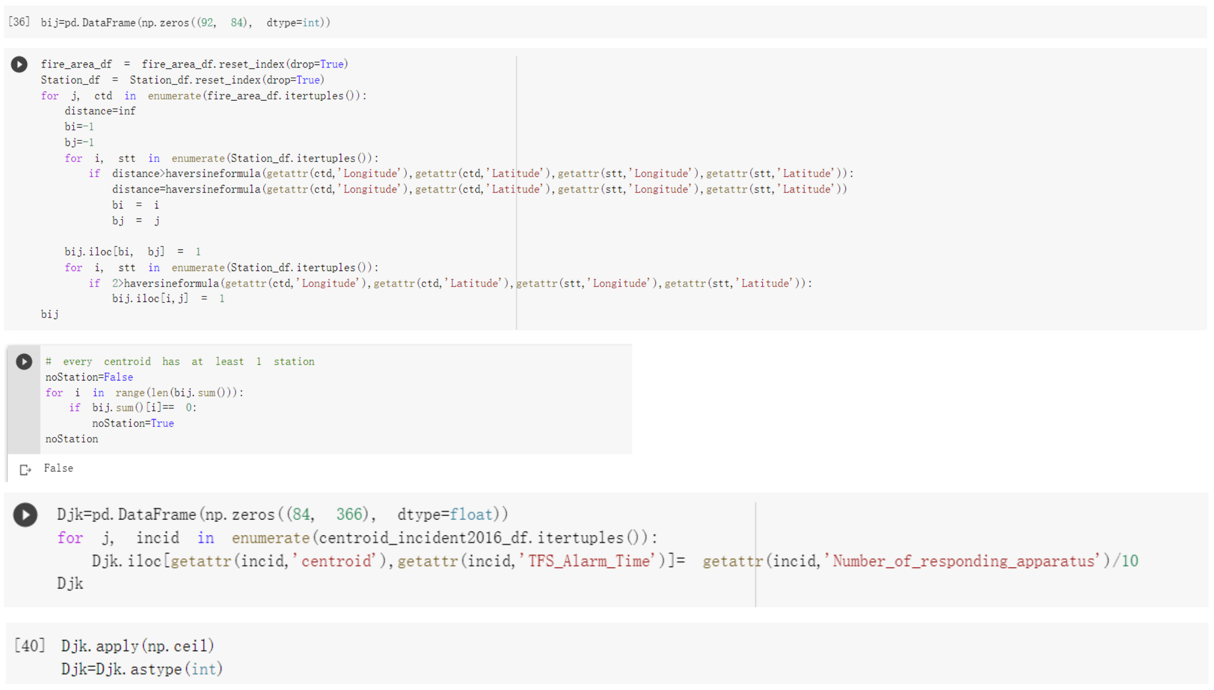
The full codes of this project can be accessed at:
https://colab.research.google.com/drive/1t6PADtYwEdxtDWC-9xWY18Bf1EQZ3lic?usp=sharing

Appendix B: Haversine Formula



```
[21] import math
     #Use haversine formula to calculate the distance between two points using the curvature of the Earth
     def haversineformula(lon1, lat1, lon2, lat2):
         lon1, lat1, lon2, lat2 = map(math.radians, [lon1, lat1, lon2, lat2])

         a = math.sin((lat2-lat1)/2)**2 + math.cos(lat1) * math.cos(lat2) * math.sin((lon2-lon1)/2)**2
         c = math.asin(math.sqrt(a))*2
         _RADIUS_OF_EARTH = 6378
         return c * _RADIUS_OF_EARTH
```

Appendix C: Optimization Model



```
[36] bij=pd.DataFrame(np.zeros((92, 84), dtype=int))
```

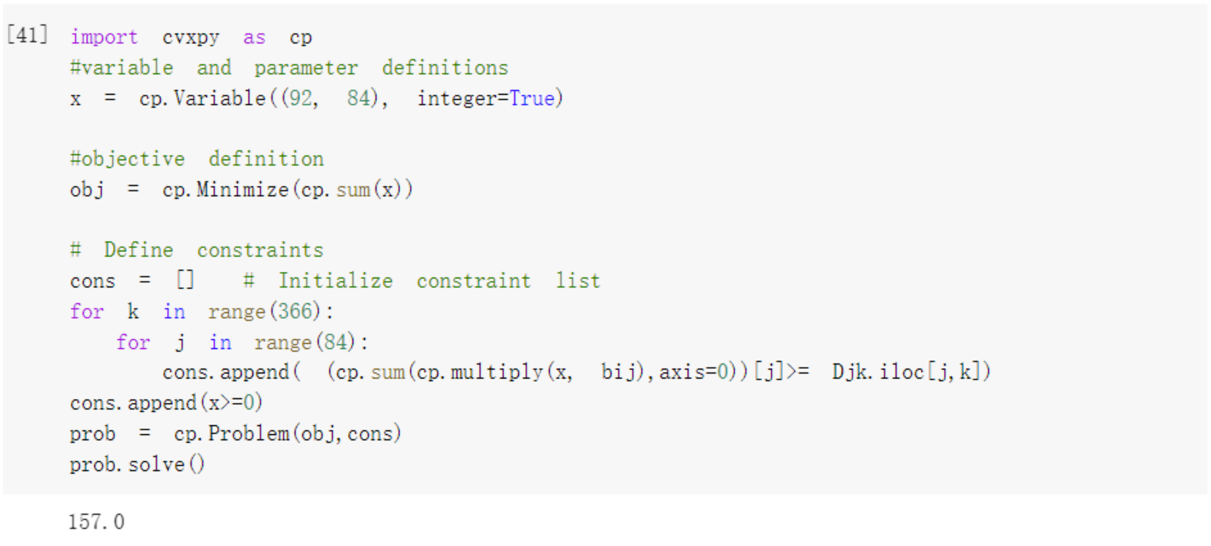
```
fire_area_df = fire_area_df.reset_index(drop=True)
Station_df = Station_df.reset_index(drop=True)
for j, ctd in enumerate(fire_area_df.itertuples()):
    distance=inf
    bi=-1
    bj=-1
    for i, stt in enumerate(Station_df.itertuples()):
        if distance>haversineformula(getattr(ctd,'Longitude'),getattr(ctd,'Latitude'),getattr(stt,'Longitude'),getattr(stt,'Latitude')):
            distance=haversineformula(getattr(ctd,'Longitude'),getattr(ctd,'Latitude'),getattr(stt,'Longitude'),getattr(stt,'Latitude'))
            bi = i
            bj = j

    bij.iloc[bi, bj] = 1
    for i, stt in enumerate(Station_df.itertuples()):
        if 2>haversineformula(getattr(ctd,'Longitude'),getattr(ctd,'Latitude'),getattr(stt,'Longitude'),getattr(stt,'Latitude')):
            bij.iloc[i,j] = 1
bij
```

```
# every centroid has at least 1 station
noStation=False
for i in range(len(bij.sum())):
    if bij.sum()[i]== 0:
        noStation=True
noStation
```

```
False
```

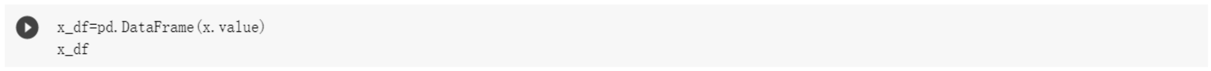
```
Djk=pd.DataFrame(np.zeros((84, 366), dtype=float))
for j, incid in enumerate(centroid_incident2016_df.itertuples()):
    Djk.iloc[getattr(incid,'centroid'),getattr(incid,'TFS_Alarm_Time')]= getattr(incid,'Number_of_responding_apparatus')/10
Djk
```

```
[40] Djk.apply(np.ceil)
     Djk=Djk.astype(int)
```

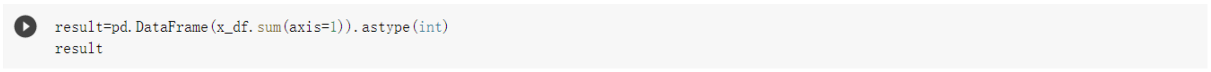


```
[41] import cvxpy as cp
     #variable and parameter definitions
     x = cp.Variable((92, 84), integer=True)

     #objective definition
     obj = cp.Minimize(cp.sum(x))

     # Define constraints
     cons = []   # Initialize constraint list
     for k in range(366):
         for j in range(84):
             cons.append( (cp.sum(cp.multiply(x, bij),axis=0))[j]>= Djk.iloc[j,k])
     cons.append(x>=0)
     prob = cp.Problem(obj,cons)
     prob.solve()

     157.0
```

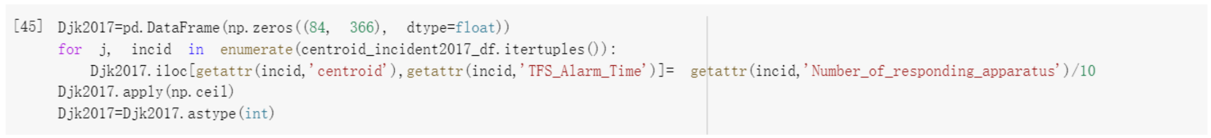
Appendix D: Score the model by the Data in 2017-2019



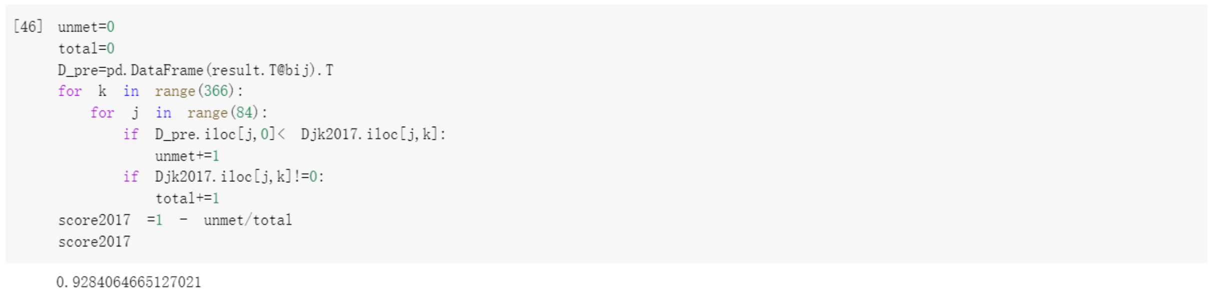
```
x_df=pd.DataFrame(x.value)
x_df
```



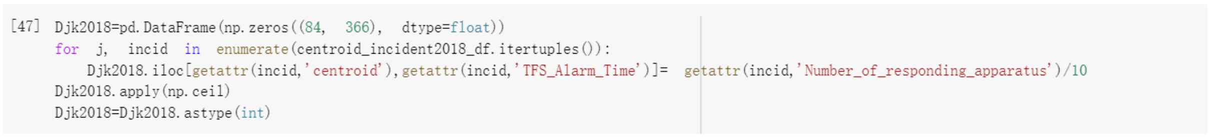
```
result=pd.DataFrame(x_df.sum(axis=1)).astype(int)
result
```



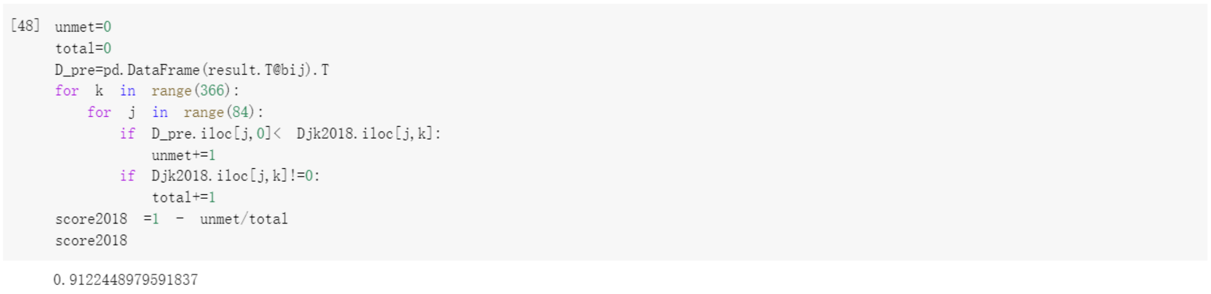
```
[45] Djk2017=pd.DataFrame(np.zeros((84, 366), dtype=float))
     for j, incid in enumerate(centroid_incident2017_df.itertuples()):
         Djk2017.iloc[getattr(incid,'centroid'),getattr(incid,'TFS_Alarm_Time')]= getattr(incid,'Number_of_responding_apparatus')/10
     Djk2017.apply(np.ceil)
     Djk2017=Djk2017.astype(int)
```



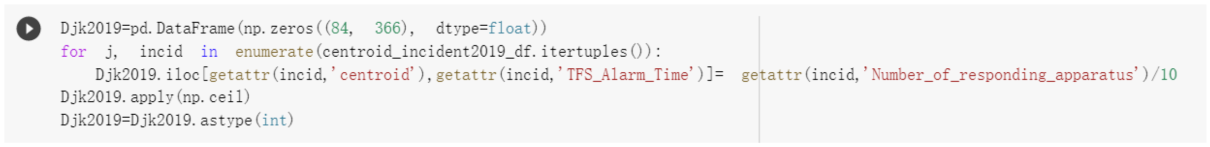
```
[46] unmet=0
     total=0
     D_pre=pd.DataFrame(result.T@bij).T
     for k in range(366):
         for j in range(84):
             if D_pre.iloc[j,0]< Djk2017.iloc[j,k]:
                 unmet+=1
             if Djk2017.iloc[j,k]!=0:
                 total+=1
     score2017 =1 - unmet/total
     score2017

     0.9284064665127021
```



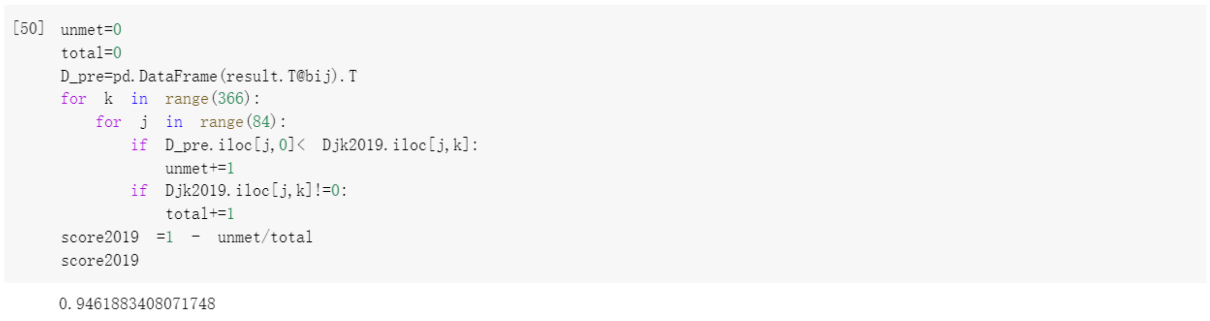
```
[47] Djk2018=pd.DataFrame(np.zeros((84, 366), dtype=float))
     for j, incid in enumerate(centroid_incident2018_df.itertuples()):
         Djk2018.iloc[getattr(incid,'centroid'),getattr(incid,'TFS_Alarm_Time')]= getattr(incid,'Number_of_responding_apparatus')/10
     Djk2018.apply(np.ceil)
     Djk2018=Djk2018.astype(int)
```



```
[48] unmet=0
     total=0
     D_pre=pd.DataFrame(result.T@bij).T
     for k in range(366):
         for j in range(84):
             if D_pre.iloc[j,0]< Djk2018.iloc[j,k]:
                 unmet+=1
             if Djk2018.iloc[j,k]!=0:
                 total+=1
     score2018 =1 - unmet/total
     score2018

     0.9122448979591837
```

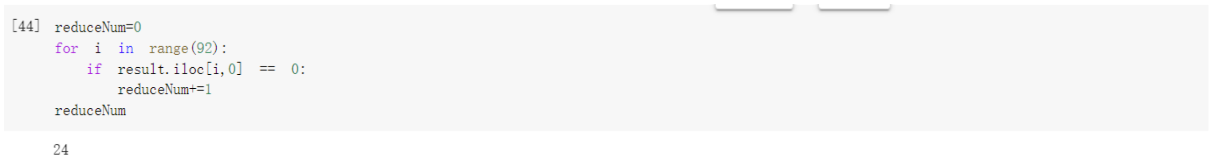


```
Djk2019=pd.DataFrame(np.zeros((84,  366),  dtype=float))
for  j,  incid  in  enumerate(centroid_incident2019_df.itertuples()):
    Djk2019.iloc[getattr(incid,'centroid'),getattr(incid,'TFS_Alarm_Time')]=  getattr(incid,'Number_of_responding_apparatus')/10
Djk2019.apply(np.ceil)
Djk2019=Djk2019.astype(int)
```



```
[50] unmet=0
     total=0
     D_pre=pd.DataFrame(result.T@bij).T
     for  k  in  range(366):
         for  j  in  range(84):
             if  D_pre.iloc[j,0]<  Djk2019.iloc[j,k]:
                 unmet+=1
             if  Djk2019.iloc[j,k]!=0:
                 total+=1
     score2019  =1  -  unmet/total
     score2019

     0.9461883408071748
```

Appendix E: 24 Fire Stations can be Removed



```
[44] reduceNum=0
     for  i  in  range(92):
         if  result.iloc[i,0]  ==  0:
             reduceNum+=1
     reduceNum

     24
```