

Going Back to Work

By Vishnu Thakral and Darshan Solanki

1 Problems Addressed:

- Increased social distancing in post-covid situation for people going back to work
- Provide maximum approval requests abiding to social distancing restrictions
- Prioritizing shorter period requests thus promoting employees for short duration stay in office
- Provide Fair access to maximum employees coming to office
- Real-Time, On-demand request for time extension from within office

2. Dataset:

The dataset used is of Yellow and Green Taxi for the year 2016 for New York City. Data of 3 Months i.e. March, April and May has been used and the combined data spans around 6GB.

- The analysis has been done only for Manhattan borough based on the Geo location coordinates that belonged to the Manhattan borough of NYC. The coordinates to work with can be changed thus the visualizations and the analysis in general can be easily scaled to the required level.
- The latitude and longitude have been scaled to the precision of 3 decimal digits which means that one unique entry covers 100 meters. We initially did the same with 4 digit (11 meter of precision) but as it resulted in overcrowded display of results and lesser numbers to show thresholding effect we proceeded with 4 digits of precision. If the user wants he/she can alter the precision to 6 digits thus going till 10cm.
- As the dropoff_coordinate is being considered we assume passenger_count to be the number of people that want to enter the office as a group.

3. Workflow:

1. We initialize by taking the dataset as mentioned above.
2. The dataset is cleaned by removing all unnecessary columns that include fare,tax e.t.c. and we keep dropoff_datetime, dropoff_latitude,dropoff_longitude and passenger_count.
3. Datetime is split to extract date and time along with day_of_week (It is a useful feature we generate to efficiently approve more requests while meeting the threshold for ensuring social distancing)
4. Visualizations are plotted on a cleaned data.
5. For each unique combination of latitude, longitude, calculate the average number of people visiting the location and store all these values divided by 2 in a csv (one each for the below cases) along with the latitude, longitude identifier.
 - i) By using the entire data for 90 days and averaging for a day.
 - ii) By combining based on days of the week. (A office space may hardly have employees on saturday, sunday but using them with calculation for working days will reduce the overall threshold we set)
6. For the requested permission to office space, access the relevant csv (based on choice of thresholding the company sets) and provide approval based on the present count of people allowed for that time slot.

We provide approvals in batch mode 1 night before the work day and on request extension from within the office.

The batch approval helps us optimize efficiency of approving maximum office building and resources access to max employees.

(both the techniques are discussed in the next section).

Note: We have taken randomly generated requests for testing.

7. Send a QR code that can be scanned to the user's email address.

4. Code Architecture

4.1 Data Generation:

According to Problem Statement: We aim to flatten the curve for any particular locations in Manhattan Borough. For simplicity, we demonstrate our project prototype for GS office location in Manhattan.

Assumption:

Request Approval Data is coming from internal portal wherein Employee fills in details like:

Start_time:

Time_to_spent:

End_time: (calculated from previous values)

No_of_people:

Requested_timestamp:

Mail_id:

- Data parameters should be accurate i.e mail_id so that they can receive accurate updates over mail.
- Employees Request form processes those plans to the office for immediate next day
- Assumed self-generated data as actual data from companies like Goldman Sachs and real-data can easily be configurable with our script

Restrictions and Awareness Provided to Employees:

Requests that come post 21:00pm would not be processed as a company policy; everyone should be self-disciplined in accordance with code of conduct.

Company is promoting short duration presence for all employees.

4.2 Code Explained Step-by-Step:

I] Data Generator:

In this how data is generated is explained based on parameters and assumptions stated in above section

data_generator.py

Functions:

generateTimeKeys - generate time range from 00:00 - 23:30 in intervals of 30 mins

getClosestTime - maps end_time to closest defined time interval. For e.g if start_time is 10:30 and time to be spent is 80 mins then the end_time would be 12:00 pm and not 11:50 am.

saveData - saves the generated data. Before that it cleans-up the requests by removing requests post 21:00 pm to ensure company policy.

generateData - this func, generates the data as discussed with fields discussed in section _ asit's parameters

II] Scheduler Script

approveReject.py

This is the main scheduler of our project where it fairly calculates the decision of approving and rejecting the request.

Functions:

approveReject

Basic working of our scheduler:

We calculate in each time-interval the number of people coming to office should not cross the threshold.

Threshold is calculated in two ways:

Overall Trend over 3 months:

In this, we analyze the trend of people coming to office and then we sum number of people coming to office / no.of.days

Reason for doing this, we want to closely monitor the trend over the days and the threshold should not be influenced by a single day where if some event would be occurring and all employees are invited in that case the threshold would drastically go up.

Day wise trend:

In this, the same logic is followed instead we monitor day-wise counter and divide the sum by 12, as we are dealing with 12 weeks data.

Reason for opting day wise:

As seen from exploratory analysis - Tuesday GS office is highly crowded, maybe the employee wants to plan ahead week's tasks. Maybe team discussion for overall project development. In such scenarios the employee rate should be high in order to perform efficient work.

Once we set the threshold, we now run the scheduler on 2 basis:

Batch Approval:

This works on First-Come-First-Served

For e.g employee want to check-in from 09:00 am to 10:30 am

- Then the slot in 30 min interval from start and end time inclusive will be incremented by 1
- This simply imply the employee presence in the office for 1 hr 30 min
- In that mean time, another request from 09:00 am to 09:30 am comes in, so now the current state will change as 2 people in that 09:00 - 09:30 am interval

- In that way, if the number of people surpass threshold we simply reject their requests
- This works well on group requests as well, in such cases we add the number of people in the time interval bucket and not single '1'

Drawback:

If we notice - employees demanding for higher spent in office will impact other employee requests. As in his entire trajectory, that employee is occupying 1 seat in that bucket and if that bucket is at threshold then no other employee even for a smaller duration could enter the office.

Optimized Scheduler:

In order to overcome such starvation of request and be fair to others who had small duration to spend in office.

We sort the request with burst-time in ascending order:

Burst-time is number of mins to be spent in office by an employee

- Ensures maximum small duration requests would be served despite their request timestamp being later than the employee requesting first
- Improve in overall request approval, which is proved from graphs and results.
- Promote stay home and come office for less time in truly essential need

generateQR - once the approval is met, we generate the QR code for that employee and it encodes all the information which is necessary to get the access to the building.

generatePlots - From our hypothesis, an optimized scheduler would serve more people requests and maintain social distancing, this function generates plots to showcase the results.

III] Real-Time On-Demand Request Approval

This script cater's on demand employee requests to extend their office end_time.

To handle such scenarios - we have created a prototype ChatBot which is supposed to get accessed from internally within the office once an employee arrives to it's desk.

- This real-time tracker takes in previous request_no which was assigned in QR generation.
- Fetch the information from the dataset for that employee
- Ask for a time extension and also flexible for group or single approval requests.
- Calculates real-time people present in the office and set a tolerance rate usually 0.5% of threshold
- If the employee makes it into the bucket list with tolerance imposed, a new QR is sent over mail and he just needs to swipe in back as the previous QR would not work as time has passed
- Update the data structure to accommodate new request

Strength of a Chat Bot:

- We can create multiple instances of the script deployed on servers and maintain privacy as other's employee request data would not be seen to others
- This Bot is self-terminated when the day ends i.e. 23:59 pm and re-instantiate at 00:00 am

5 Code Integration:

Create **qrCode** and **OnDemand_qrCode** folder which will store all generated QRs'

Simply run ---- main_func.py

- generate data
- cleaning of data and saving that cleaned request file
- running the scheduler to approveReject those request
- running the scheduler again on optimized data request

- Sending the QR code over mail
- Instantiate a ChatBot which terminates as the day ends

Place all the scripts parallel to main_func.py

Data file generated over the run

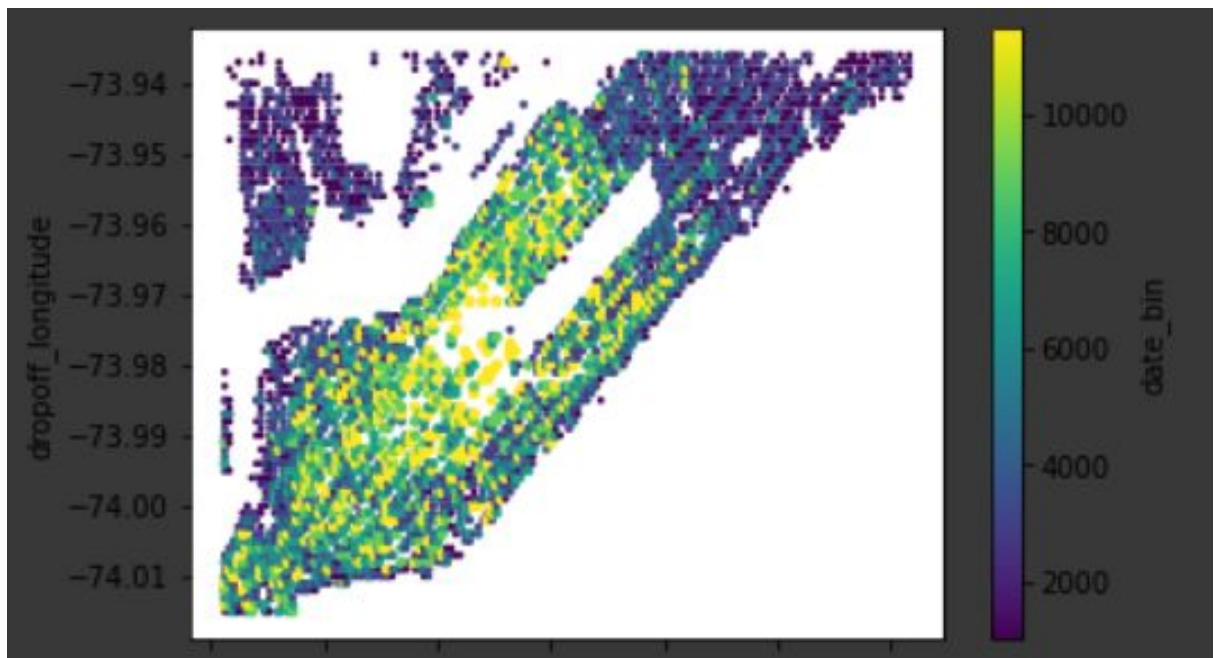
Request_sheet.p - this file can be huge hence saved as pickle to compress the data. Has complete requests from employees

Optimized_sheet.p - contains data shuffle in an optimal way to server more request

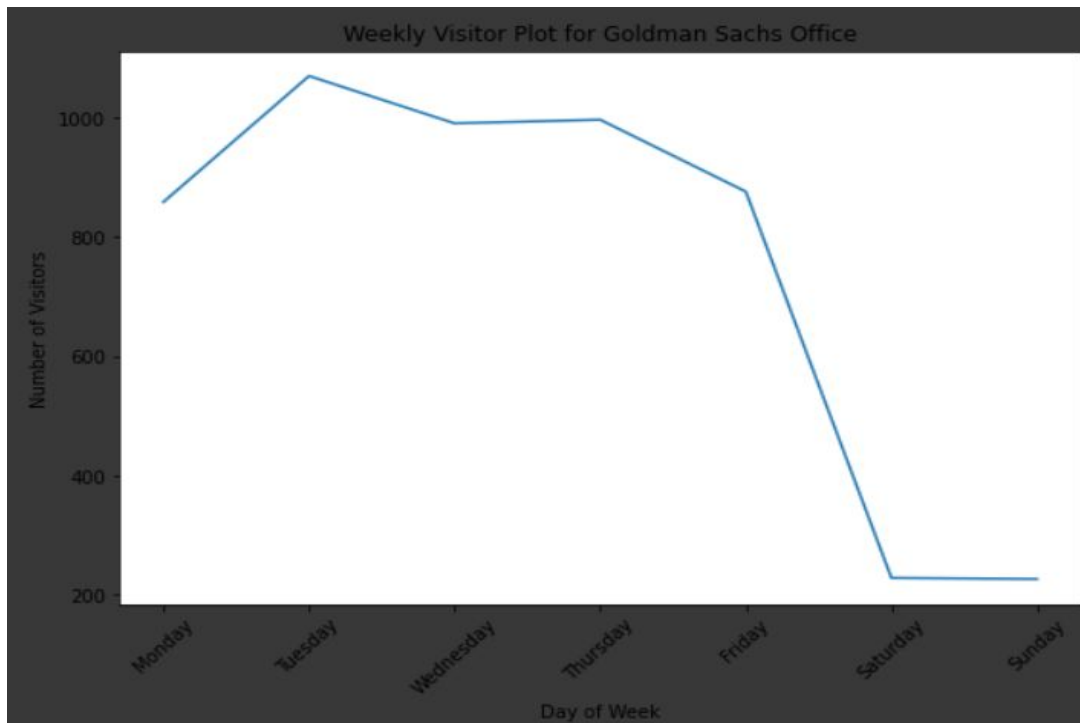
Scheduler.p - this data involves for which scheduler has approved the request and the model is saved to be retrieved when on-demand request is served.

6. Visualizations:

- Locating hotspots: The data is clubbed as per latitude and longitude to get the hotspots in the selected region.

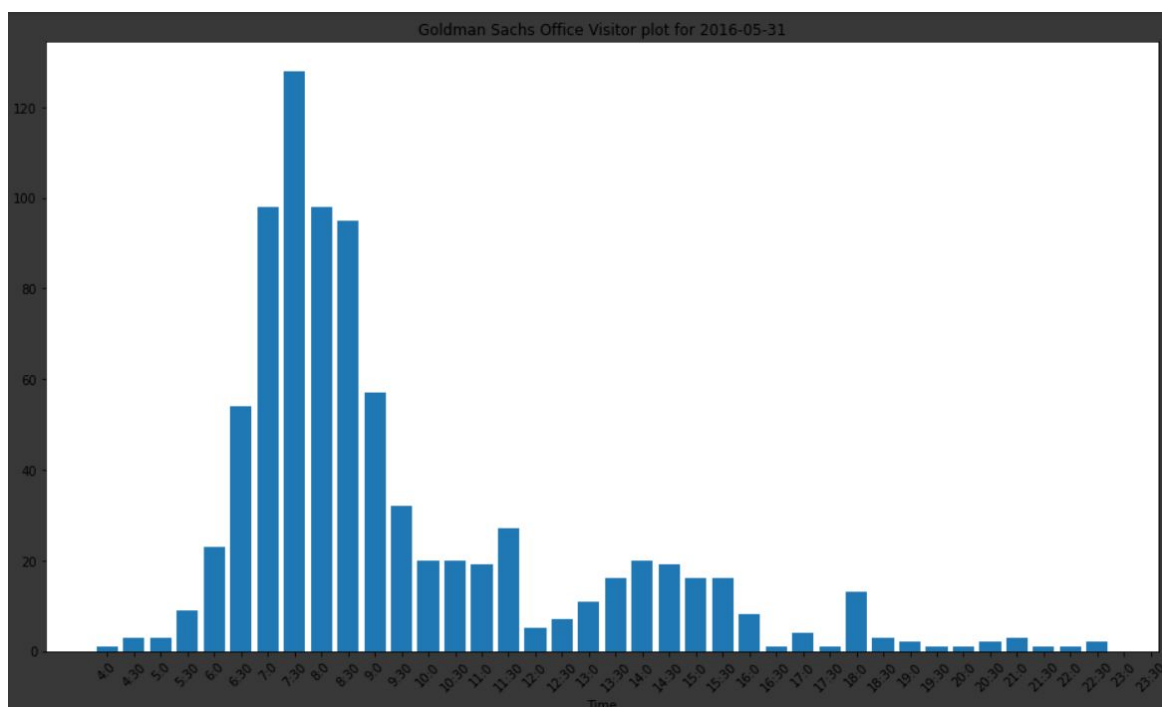


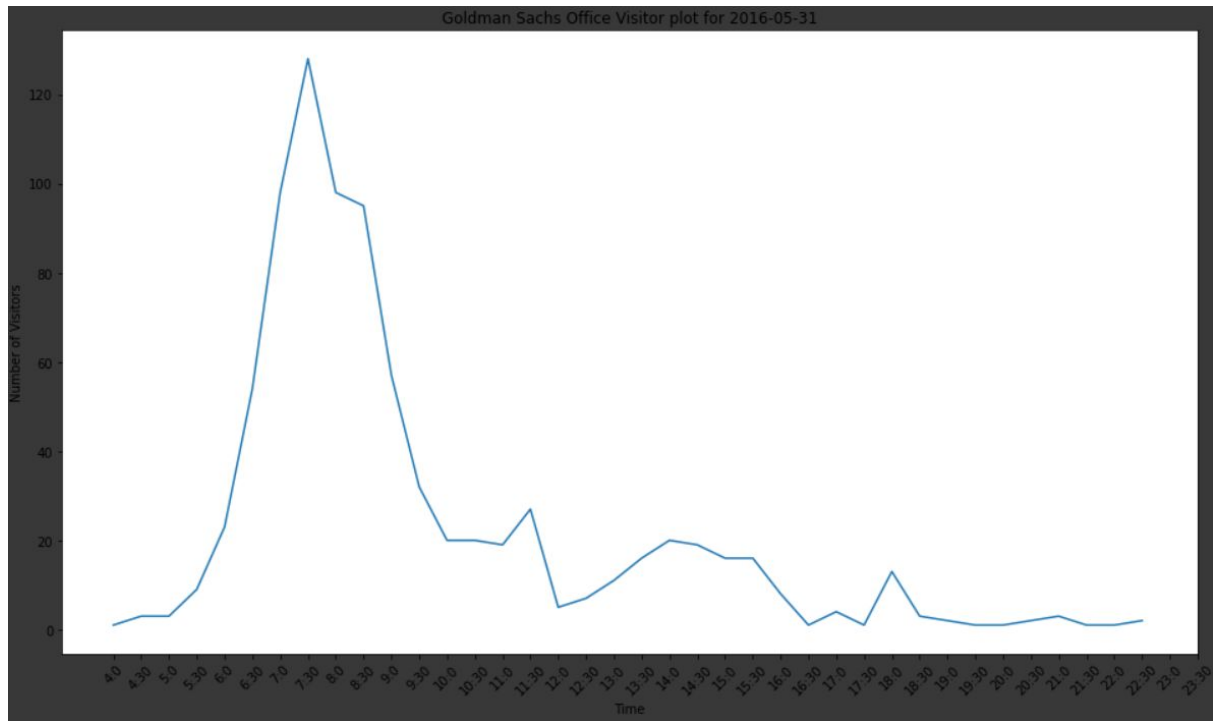
ii) Day wise plot of the employees visiting office of Goldman Sachs



It is evident that saturday and sunday are reducing the average and thus the threshold of employees whom we can permit to come to office.

iii) Hourly plot of Goldman Sachs employees in office (Bar and Line).





We can see that the maximum employees arrive at the office around 8am.

- By default we are analyzing the results for the location of Goldman Sachs which is at the coordinate 40.
- The user has 2 options for plotting.
 - i) For location of Goldman Sachs enter 0,0 as lat,lon
 - ii) For any location enter the actual coordinate values as lat lon

8. Results Explained from Code :

Below are 4 graphs which showcase overall trend of three months and limiting the curve for given threshold for that particular location

Figure 1

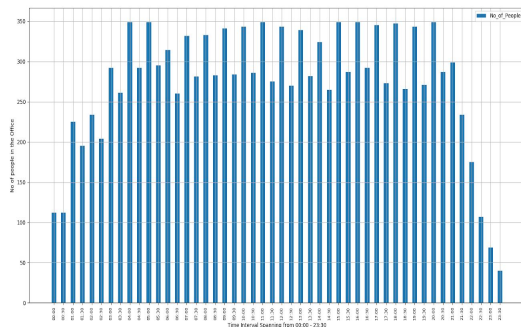


Figure 2

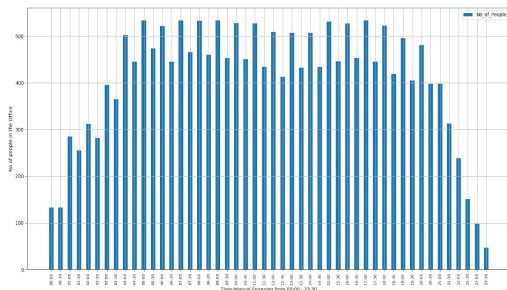
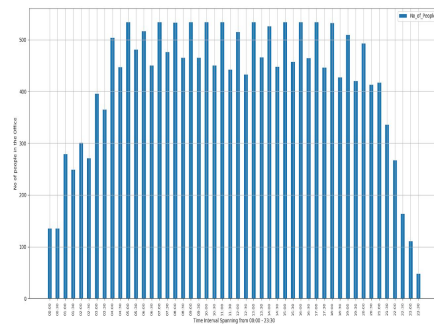


Figure 3

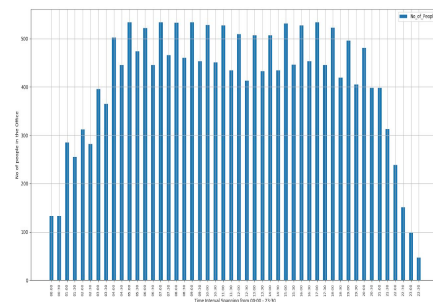


Figure 4

As we can see Figure 1 and Figure 2 where overall threshold was set by analyzing the trend for 3 months, we could observe the request optimized from 63% - 76%

Figure 3 and Figure 4, here the threshold set is day-wise and not overall hence we could achieve more approval requests and optimization has gone up to 93% which is a good acceptance rate and fairness maintained for all the employees.

```
Percentage of approved requests 63.71769383697813 %
Percentage of approved requests 76.49105367793241 %
Percentage of approved requests 90.4075546719682 %
Percentage of approved requests 93.09145129224652 %
|
```

This is ChatBot where employees can ask for a time extension and get the response at that moment itself.

The working of this ChatBot is explained clearly in section

```
Hi
[30, 87, 103, 114, 188, 334, 352, 373, 380, 397]
Alexa - Welcome to helpdesk for extension time
Enter Previous Request Number Allocated to Fetch All Information
103
-----User Information-----

Start_Time : 04:00
End_Time : 04:30
Time spent : 0.5
People Approved : 1
Enter the time to be extended(mins): 80
Do you want to request extension for 1 people Y/N : N
Enter no of people3
Wait for Processing your On Demand Request.....
  You will be notified on mail Shortly
Please Check your Inbox
|
```

7. Future Work:

As the taxi drop off can not happen in central park we can see that the area is clean but in reality it has people so more data like cell phone signals can be used to allow the number of people to the park and maintain social distancing to public places as well.

Can account for more than one location and analyzing the trends area-wise rather particularly building, this can restrict the number of people in locality wise.

Here is the link for demonstration of our project working:

https://drive.google.com/open?id=11YZCzLehT5_aS1MQEpZYFeuXVMxO2VX-