Envisioning High Demand Areas in NYC city for Yellow Taxi's

Faculty of Graduate Studies & Research

Submitted By: Simranjeet Randhawa
Student ID: 200412297

DATE 06-04-2020

University of Regina

Table of Contents

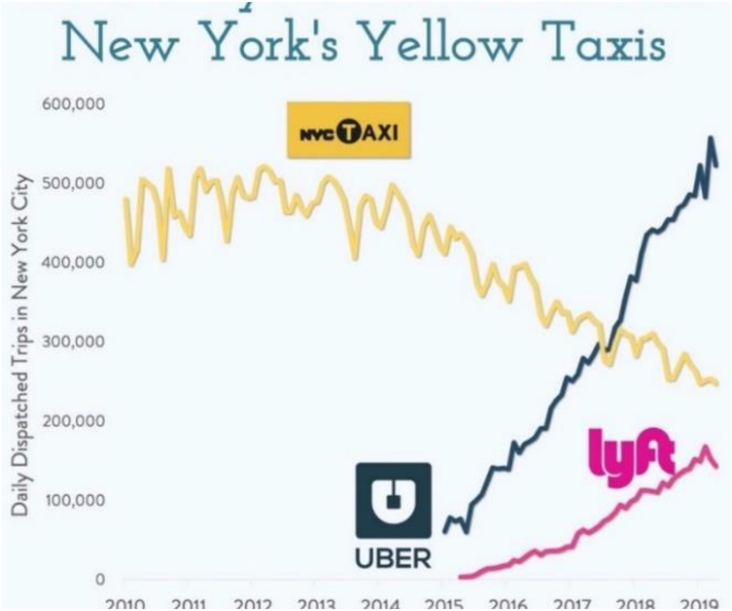
University of Regina

Introduction

- The Taxi and Limousine Commission(TLC) is an agency of the New York City government that licenses and regulates the taxi medallion.
- TLC(Taxi and Limousine Commission) mainly includes yellow taxis, FHV’s(For hire vehicles), Green cabs.
- Yellow taxi cabs is the most iconic among all as it reserve the privilege to street-hailing passengers anywhere in NYC.
- Whereas FHV provides only pre-arranged services and Green Taxi permits street hailing but is restricted to some zones only in NYC city

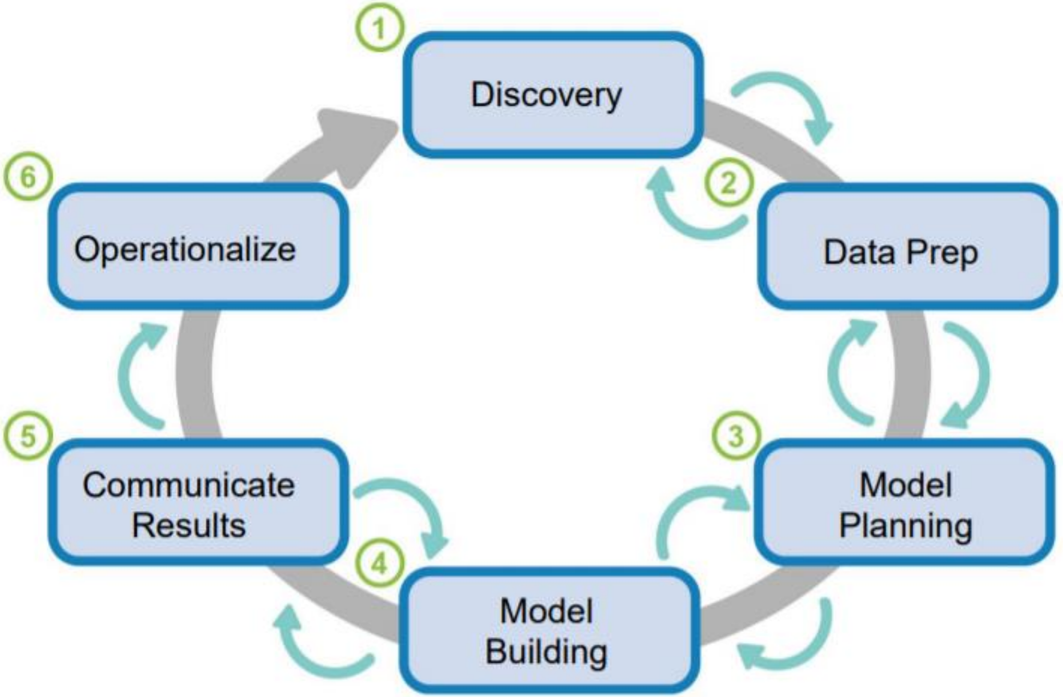
Business Problem Statement

- ❖ Main problem statement: How to redistribute these taxis in NYC so that the no taxi driver has to wait for passenger or minimise the waiting time.
- ❖ How many pickups are required in each zone of New York and can weather effect the trips in the neighbourhoods of NYC.
- ❖ This will help the yellow cab drivers prepare well ahead of time.



University of Regina

Data Analytics Lifecycle

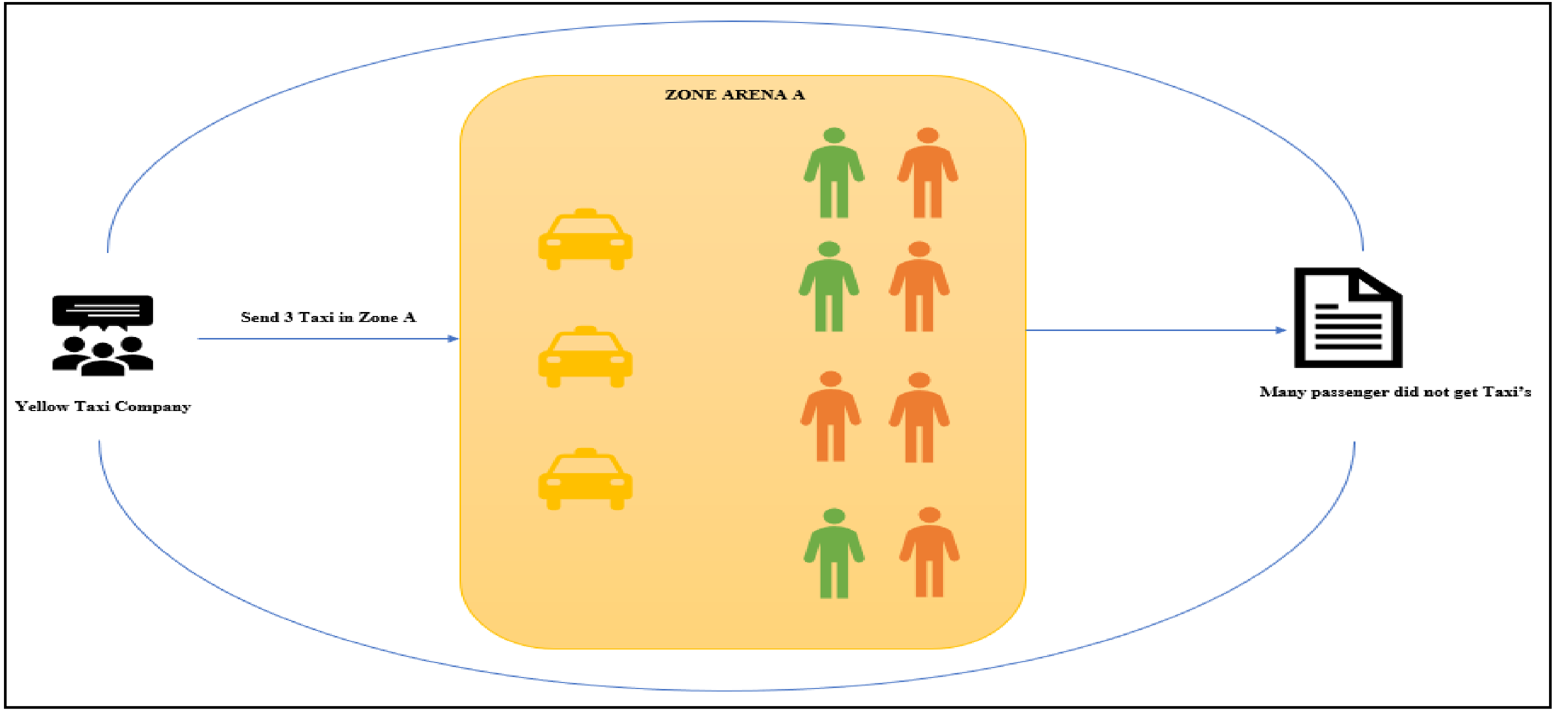


Data Discovery

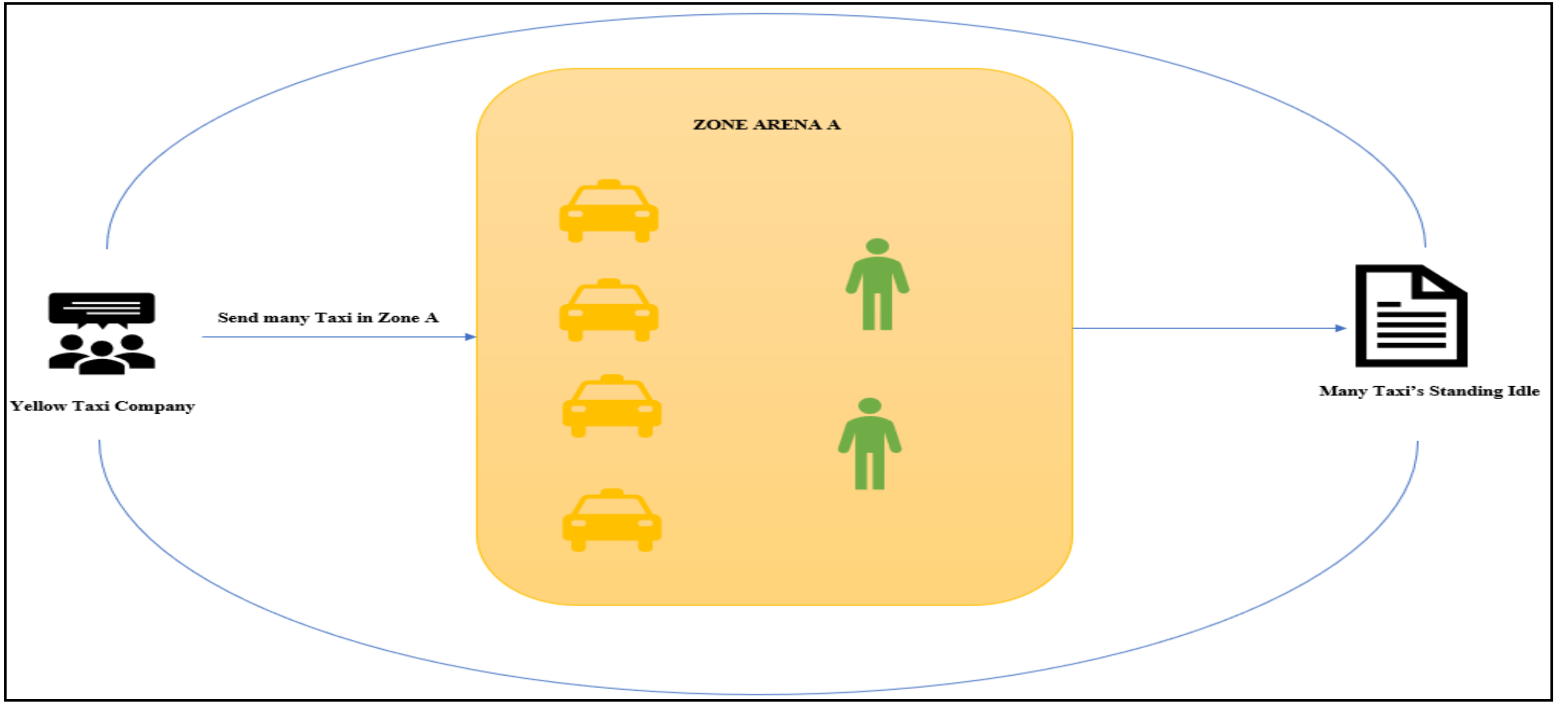
KEY ACTIVITIES:

- Business problem statement drafted.
- Problem considered as a data analytics challenge.
- Assess resource needs and availability.
- Drafted an analytic plan.

Current Scenario

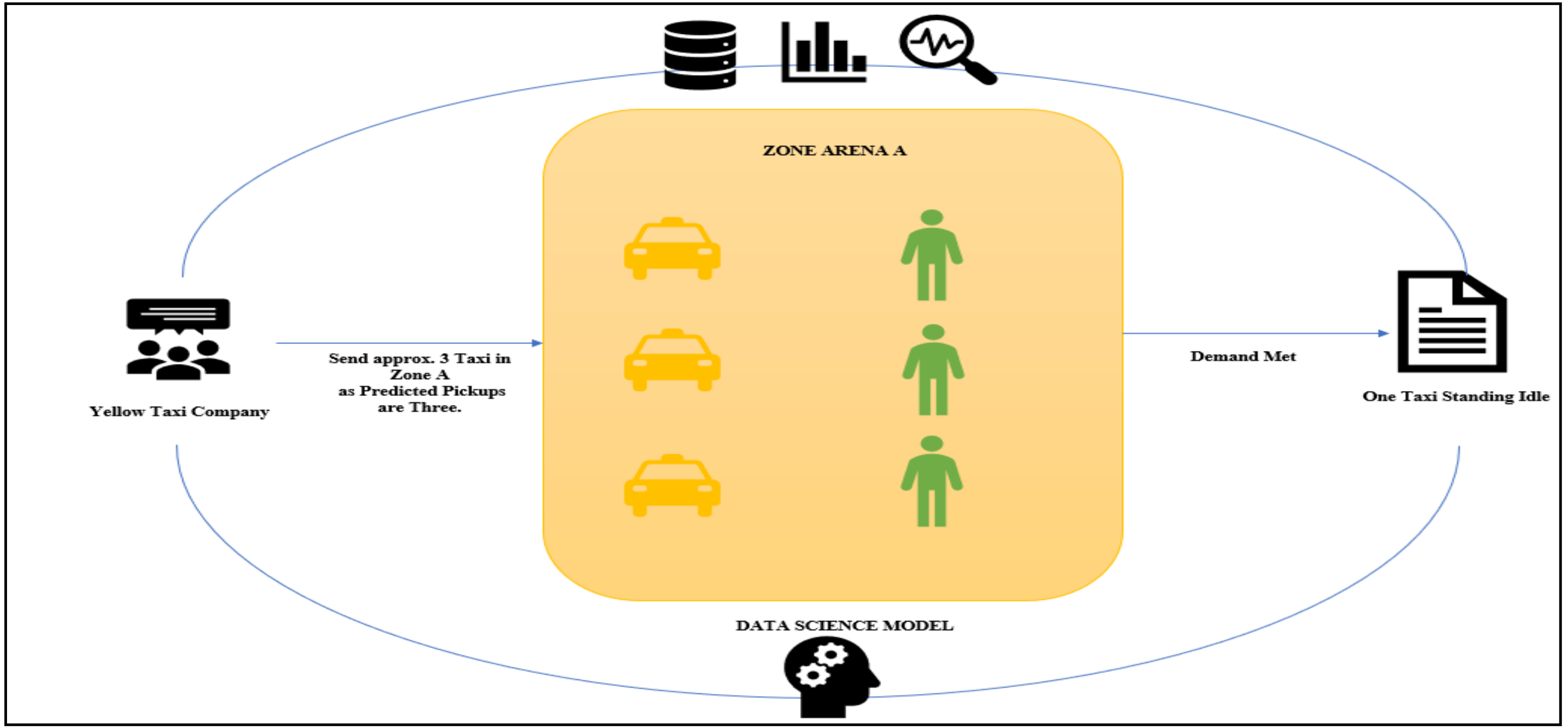


Current Scenario



University of Regina

Desired Situation



Gathering Data using Pyspark

```
from pyspark.sql import SparkSession
from pyspark import SparkFiles
spark = SparkSession.builder.master("local[*]").getOrCreate()
```

```
url1 = "https://s3.amazonaws.com/nyc-tlc/trip+data/yellow_tripdata_2018-01.csv"
spark.sparkContext.addFile(url1)
df1 = spark.read.csv("file://"+SparkFiles.get("yellow_tripdata_2018-01.csv"), header=True, inferSchema= True)
```

Spark evaluates DataFrame lazily which means computation happens only when action appears

Data Statistics

Table info

Table ID	nyc-taxi-265120:NYC.2018SecondHalf
Table size	7.15 GB
Number of rows	48,878,515
Created	Mar 5, 2020, 8:45:31 PM
Table expiration	Never
Last modified	Mar 5, 2020, 9:48:06 PM
Data location	US

Table info

Table ID	nyc-taxi-265120:NYC.2018firsthalf
Table size	6.58 GB
Number of rows	53,925,735
Created	Mar 5, 2020, 7:43:41 PM
Table expiration	Never
Last modified	Mar 5, 2020, 7:43:41 PM
Data location	US

Table info

Table ID	nyc-taxi-265120:NYC.2019firsthalf
Table size	6.8 GB
Number of rows	44,459,136
Created	Mar 5, 2020, 7:44:44 PM
Table expiration	Never
Last modified	Mar 5, 2020, 7:44:44 PM
Data location	US

Table info

Table ID	nyc-taxi-265120:NYC.2019secondhalf
Table size	6.13 GB
Number of rows	39,939,883
Created	Mar 5, 2020, 7:49:18 PM
Table expiration	Never
Last modified	Mar 5, 2020, 9:01:56 PM
Data location	US

Table info

Table ID	bigquery-public-data:noaa_gsod.gsod2019
Table size	740.54 MB
Number of rows	4,156,054
Created	Jan 11, 2019, 11:03:04 AM
Table expiration	Never
Last modified	Feb 27, 2020, 4:14:03 PM
Data location	US

Table info

Table ID	bigquery-public-data:noaa_gsod.gsod2018
Table size	723.88 MB
Long-term storage size	723.88 MB
Number of rows	4,010,814
Created	Jan 10, 2018, 1:49:27 PM
Table expiration	Never
Last modified	Jan 10, 2019, 9:15:04 PM
Data location	US

Zone Data (Shape File)

Zone centroids using Power BI

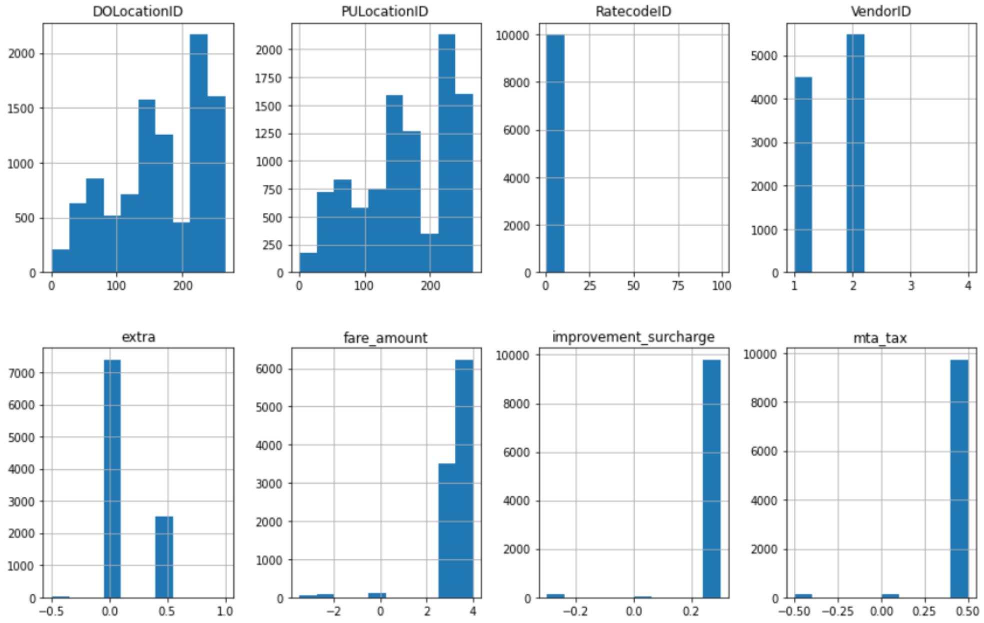
Zone Map

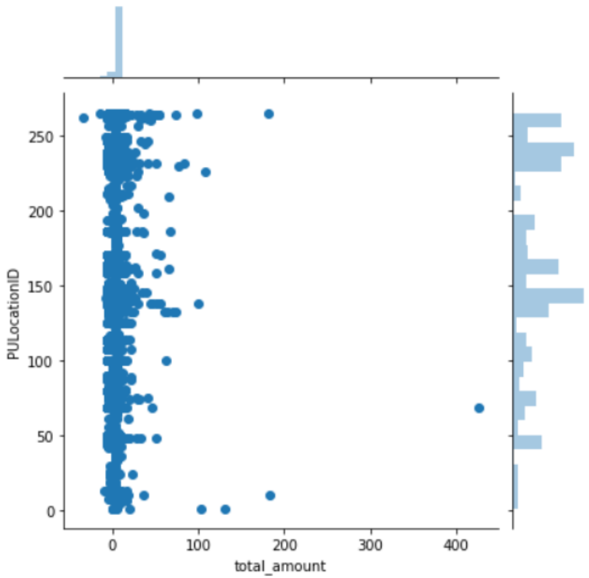
Data Preparation

KEY ACTIVITIES

- Established the analytic sandbox (Google Big Query Analytics Sandbox).
- Extract, Transform, Load, and Transform (ETLT)
- Data exploration
- Data conditioning (merging)
- Removing outliers/Missing data
- Summarize and visualize the data

Visualization Before Cleaning





University of Regina

Data Exploration using Big Query

- Analyze if there are rows with total trip amount < 0?
- Are there rows with pick up time > drop off time?
- Analyze rows that have 0 or less passenger count.
- Finding the number of trips where the trip duration is more than 12 hours.

```
query = """
SELECT * FROM `nyc-taxi-265120.NYC.2019secondhalf` where tpep_pickup_datetime > tpep_dropoff_datetime
"""
df = client.query(query).to_dataframe()
```

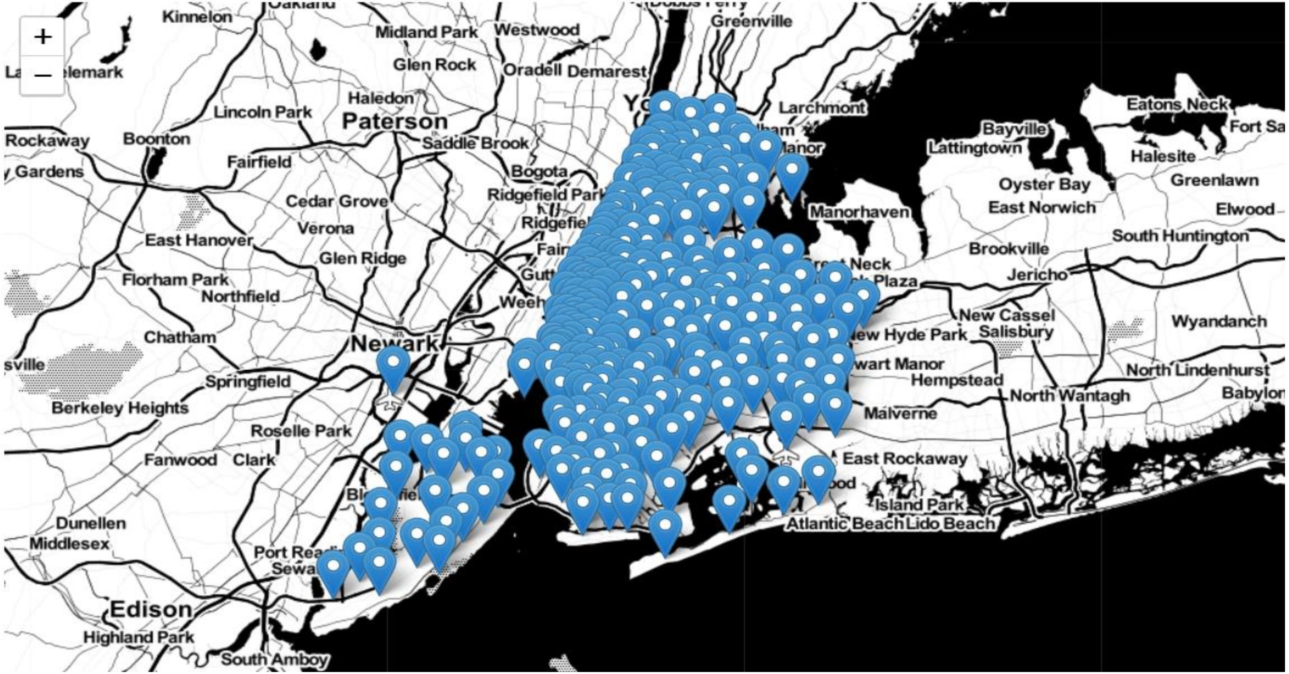
Wrangling of Data

- Also known as data munging. In this the data is transformed from a raw format into an appropriate format that can serve to be useful for the user.
- A lot of Wrangling was required to put the raw data to appropriate format. (Both using Big query and Pandas functions).

```
a=df1.pivot_table("label", "time", "zone_id")
```

```
b=df1.pivot_table("label", "time", "zone_id").unstack().reset_index()
```

Zone Visualization



University of Regina

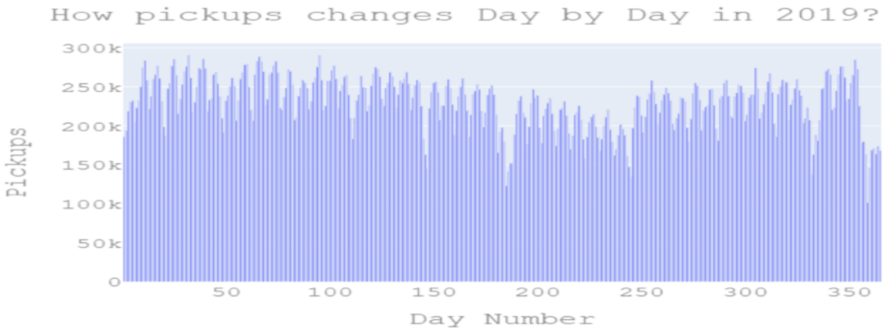
Model Planning

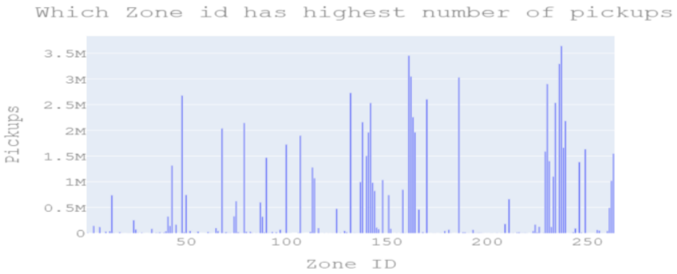
KEY ACTIVITIES

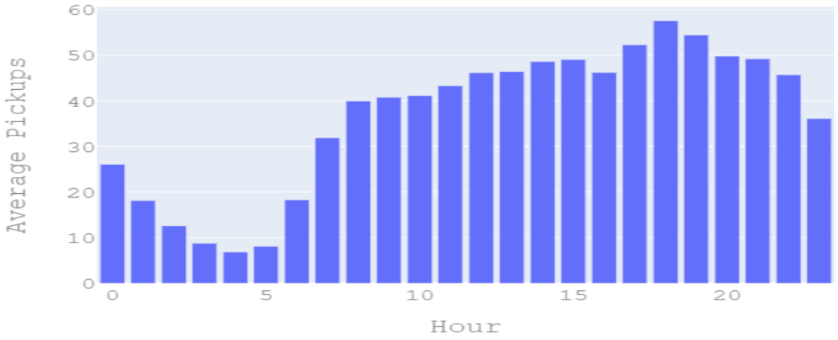
Variable Selection

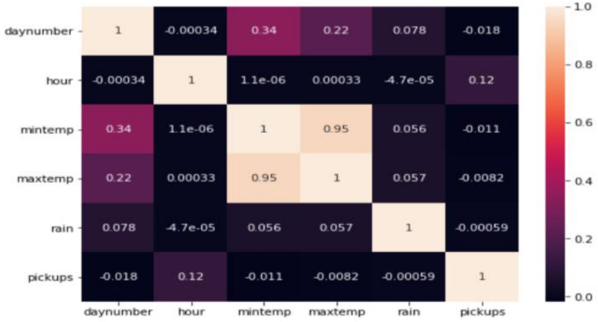
Model Selection

Data Exploration









University of Regina

Variable Selection

- **Zone id:** Zone id is a categorical variable and can be thus has been handled using the technique of One Hot Encoding.
- **Hour of the day:** It is also an important factor
- **Day of the week:** It is again a feature that can be considered to train the model.
- **Day of the year:** It is again a feature that can be considered to train the model. The pickups vary as the days progresses.
- **Min Temp**: It can be considered to build the model.
- **Max Temp:** It is the maximum temperature that can be reached in a day.
- **Rain:** Whether raining or not.

Feature Engineering

- One Hot Encoding: It is used to handle categorical variable.

```
import pandas as pd

one_hot = pd.get_dummies(data['zone_id'])
```

- Binning:

```
import pandas as pd
data.time = pd.to_datetime(data.time,format='%Y-%m-%d')
data.index = data.time
data = data.drop('time', axis=1)
data = data.resample('D').sum() # Resmapling the time series data with month starting first.
```

Model Selection

For the regression method I selected the following:

- Linear Regression
- Xgboost Regressor
- Random Forest Regressor
- Light GBM

For time series analysis I selected the following:

- ARIMA
- Prophets to get the Seasonality

Model Building

<u>KEY ACTIVITIES</u>

- I took care of following while Building model:
- Train, test spilt with cross validation.
- Grid Search CV to select the best model.
- Metrics Used to Select Best Model: MAE, MASE , RMSE.

Train, Test and Cross Validation

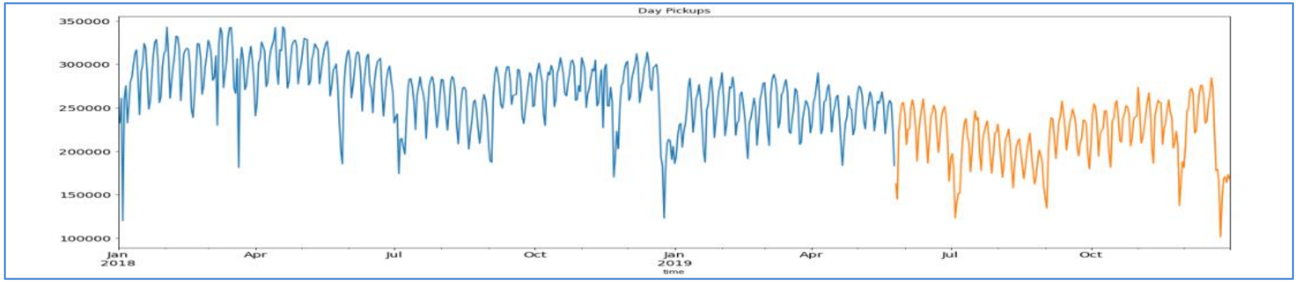
```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=20)
```

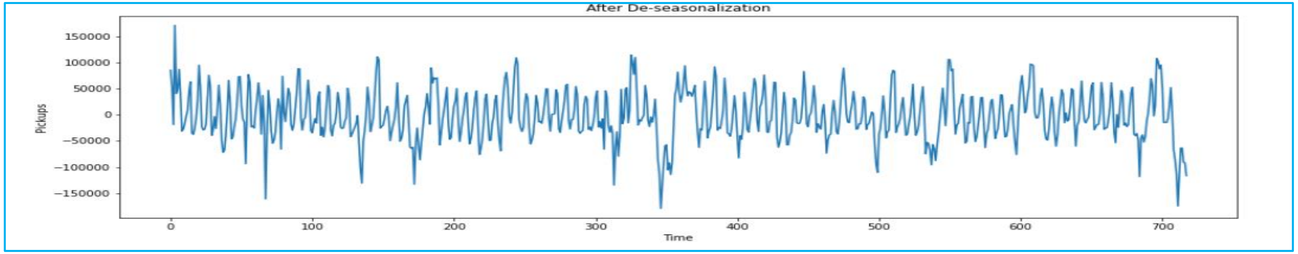
```
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import RandomForestRegressor
# Create the parameter grid based on the results of random search
param_grid = {
    'bootstrap': [True],
    'max_depth': [80, 90, 100, 110],
    'max_features': [2, 3],
    'min_samples_leaf': [3, 4, 5],
    'min_samples_split': [8, 10, 12],
    'n_estimators': [100, 200, 300, 1000]
}
# Create a based model
rf = RandomForestRegressor()
# Instantiate the grid search model
grid_search = GridSearchCV(estimator = rf, param_grid = param_grid,
                          cv = 3, n_jobs = -1, verbose = 2)
```

```
param_grid = {
    'max_depth': [80, 90, 100, 110],
    'max_features': [2, 3],
    'min_samples_leaf': [3, 4, 5],
    'min_samples_split': [8, 10, 12],
    'n_estimators': [100, 200, 300, 1000]
}
# Create a based model
rf = xgb.XGBRegressor()
# Instantiate the grid search model
grid_search = GridSearchCV(estimator = rf, param_grid = param_grid, cv = 3, n_jobs = -1, verbose = 2)
```

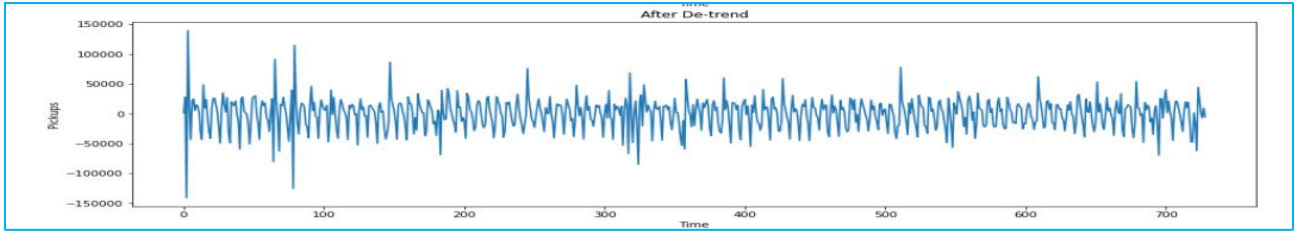
Time Series Data



Actual Dataset



Removing Seasonality



Removing Trend

University of Regina

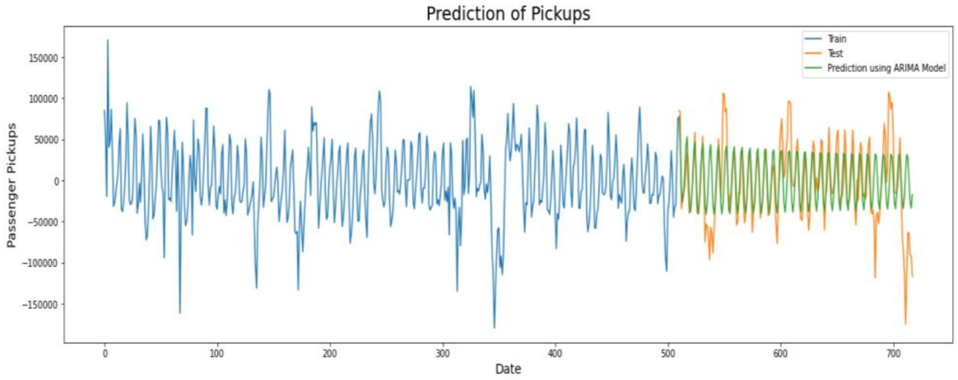
Time Series Data

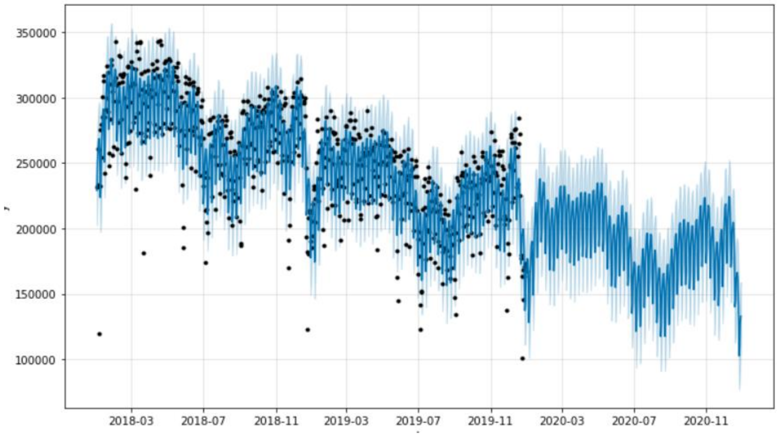
Before

```
Results of Dickey-Fuller Test:
Test Statistic                 -1.862083
p-value                         0.350103
```

After

```
Results of Dickey-Fuller Test:
Test Statistic                 -7.981551e+00
p-value                         2.615200e-12
```

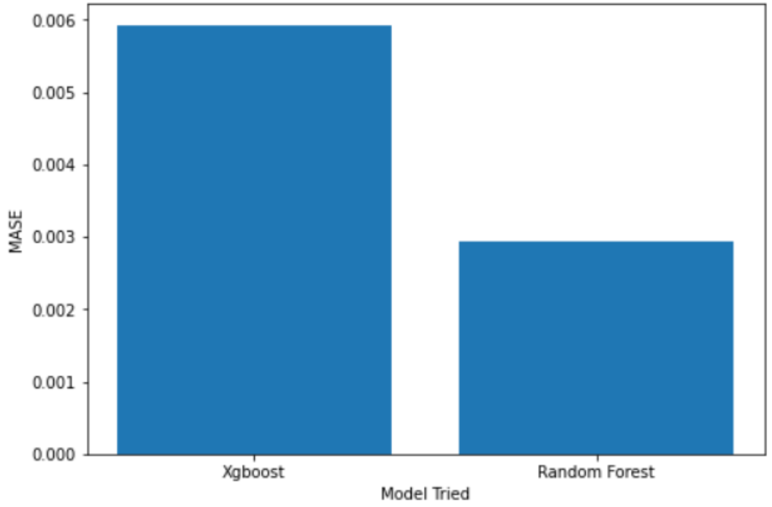




University of Regina

Metrics Calculated

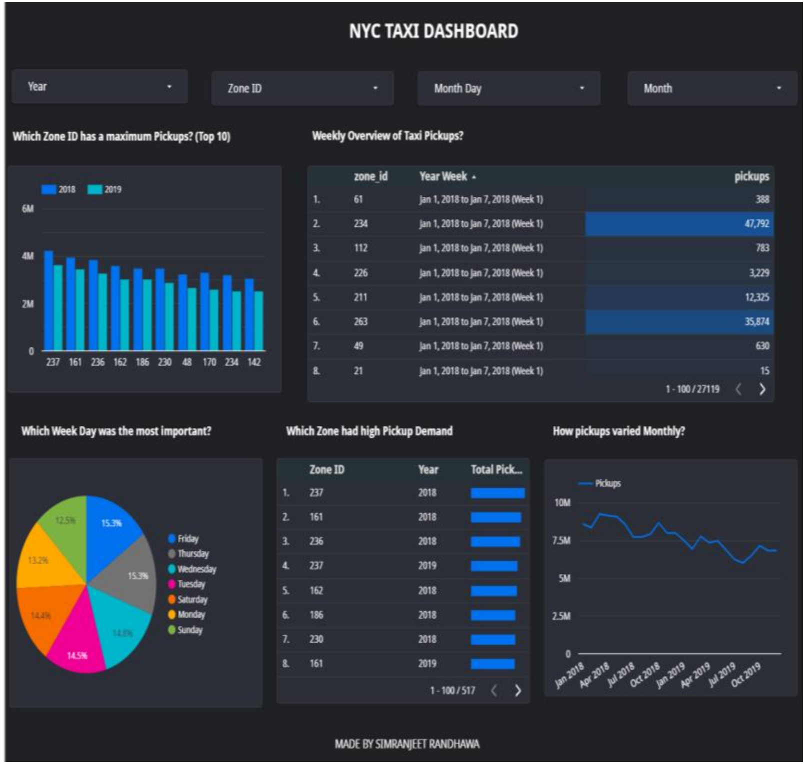
Model	MAE	R2	RMSE
Random Forest Regressor	11.57897810134032	0.45528366614765486	24.93736930641297
Xgboost	13.57897810134032	0.7528366614765484	32.93736930641297
Light GBM	18.57897810134032	0.9552836661476548	40.93736930641297
ARIMA	14.95411850134032	0.81357330134032	35.56811210134032
Linear Regression	21.57677810134032	0.9972836661476548	43.93737930641297



Communication

<u>Key Activities:</u>

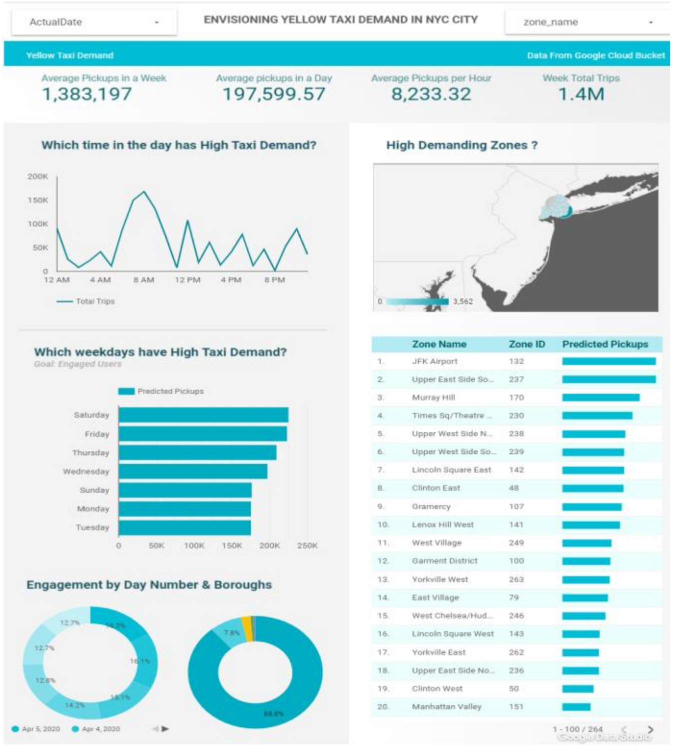
- Prepared Dashboards for the executive.
- Prepare Jupyter Notebooks for each phase.
- Model Used - Random Forest Regressor.
- Due to least MAE and MASE.



University of Regina

Communication

- Predicting upcoming weeks pickups and representing the pickups in the form of a Dashboard.
- Dashboard gives the ability to analyse the data quickly and effectively.
- It will also help the business takes effective decision in advance.



Communication

Operationalize

- A fully functional Flask app is deployed so that model can be reused.
- Google Cloud App Engine was used to deploy my model.
- I used the local server for development and us ed the App Engine Google server for production. Also App Engine provides the DevOps support.

Code

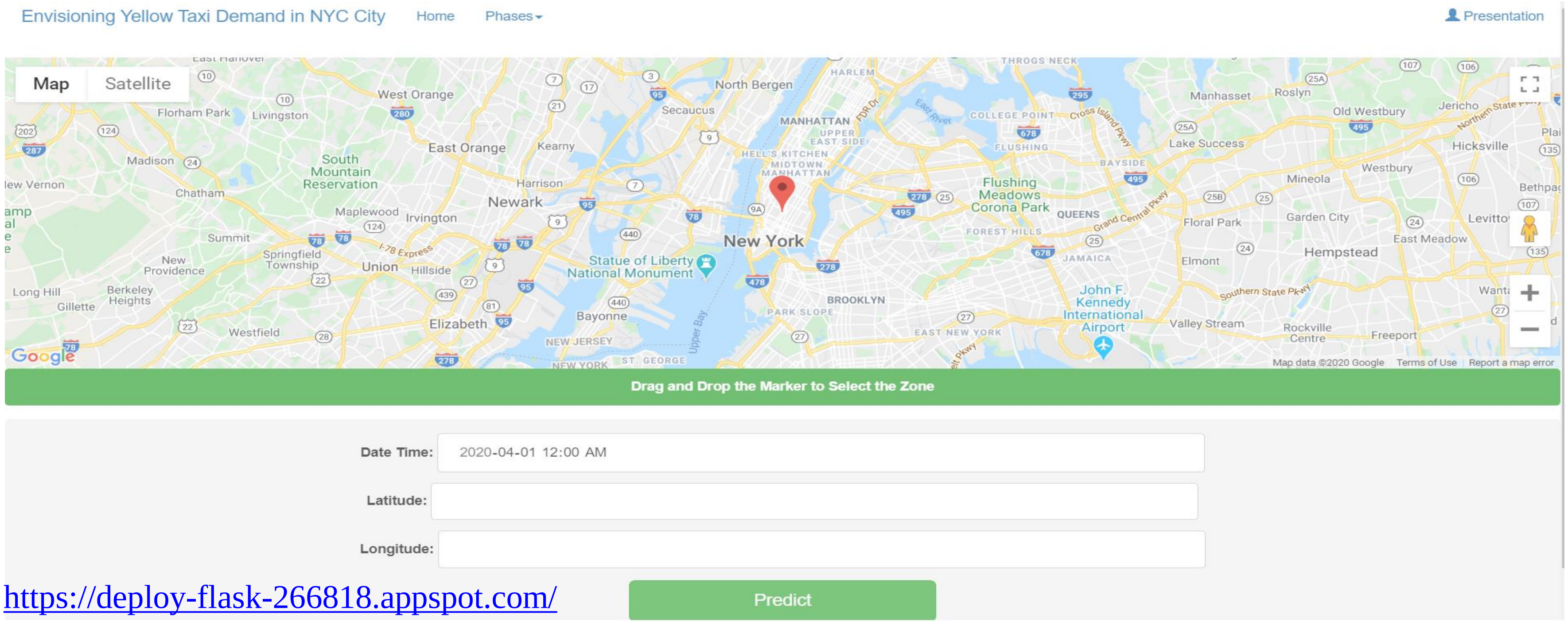
https://github.com/ssrbazpur/Envisioning-Yellow-Taxi-High-Demand-Areas-in-NYC-city/tree/master/Operationalize/Flask%20app

Website

https://deploy-flask-266818.appspot.com/

University of Regina

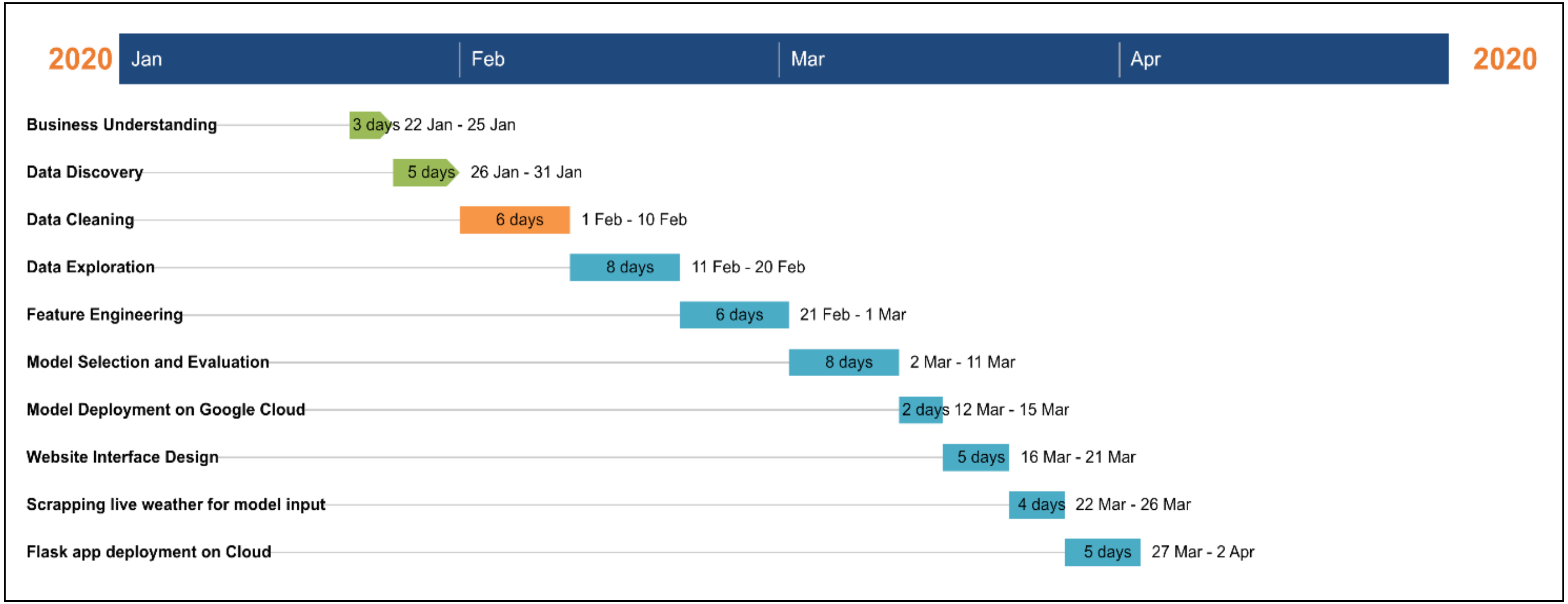
Operationalize Web Flask App



https://deploy-flask-266818.appspot.com/

University of Regina

Timeline Managed



University of Regina

Outcome Obtained

- ❑Taxi driver will be entering the date time and location and the application will return the number of pickups the taxi driver can expect.
- ❑ Live Dashboard showcasing weekly forecasted pickups in each zone of NYC city. Live Scrapped temperature is also considered.

Project Category

- Fully developed and operational cloud-based published website.
- Also, this will help NYC yellow cabs agencies to send the right number of taxis to correct location to meet the demand of the pickups at that location.
- Scrapping the live weather from google for the demand prediction.
- Not enough memory error when training the data in my laptop makes this problem a Big Data problem.

Business Value

- This website will make it possible to redistribute the yellow cabs in the right areas at the right time.
- Each and every taxi driver can use this website to know the high demanding area well before of the time.
- The executives can use my dashboard to take effective decision.
- By law, there are **13,587 taxis** in New York City. Hence the potential number of users are certainly **greater than 10,000**.

Environment and Tools

- Google Cloud Platform
- Big Query
- PySpark
- Deploy Flask app on Google Cloud / AWS.
- Power BI and Data Studio for Data Exploration.
- Ploty in Python for Data Visulizations.
- Git and Github


THANK YOU
University of Regina