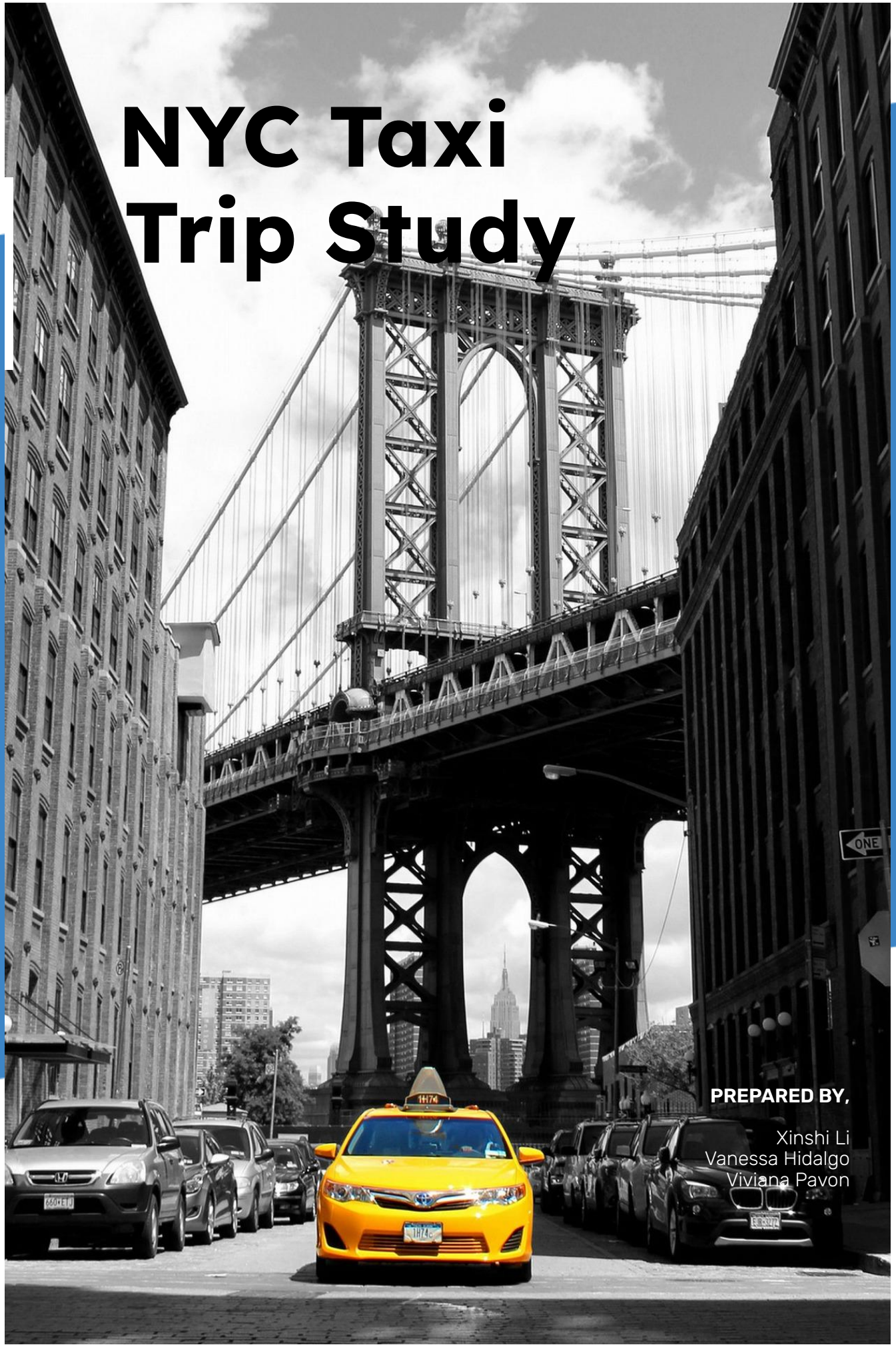
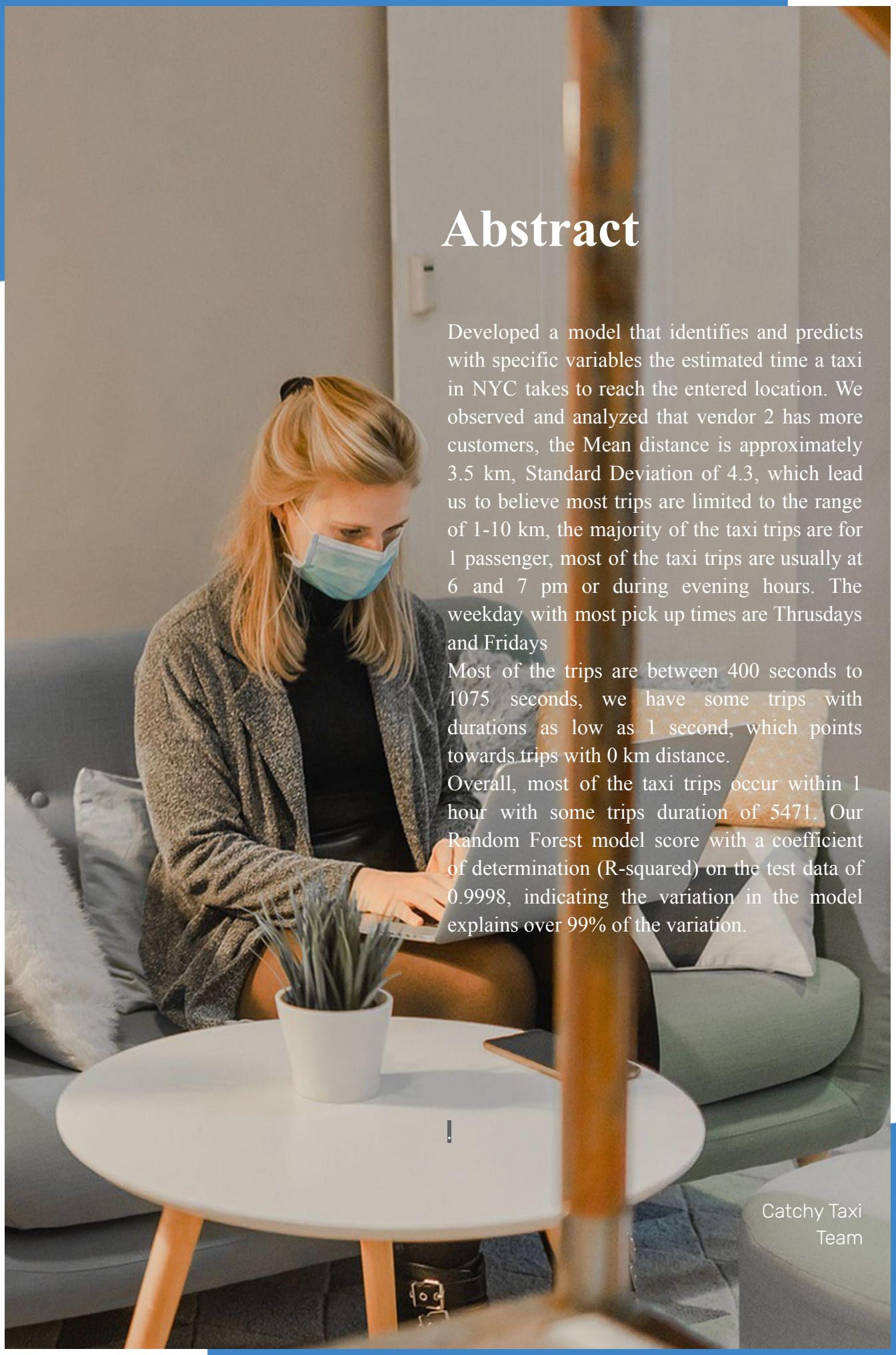
NYC Taxi Trip Study



PREPARED BY,

Xinshi Li
Vanessa Hidalgo
Viviana Pavon



Abstract

Developed a model that identifies and predicts with specific variables the estimated time a taxi in NYC takes to reach the entered location. We observed and analyzed that vendor 2 has more customers, the Mean distance is approximately 3.5 km, Standard Deviation of 4.3, which lead us to believe most trips are limited to the range of 1-10 km, the majority of the taxi trips are for 1 passenger, most of the taxi trips are usually at 6 and 7 pm or during evening hours. The weekday with most pick up times are Thrusdays and Fridays

Most of the trips are between 400 seconds to 1075 seconds, we have some trips with durations as low as 1 second, which points towards trips with 0 km distance.

Overall, most of the taxi trips occur within 1 hour with some trips duration of 5471. Our Random Forest model score with a coefficient of determination (R-squared) on the test data of 0.9998, indicating the variation in the model explains over 99% of the variation.

Catchy Taxi
Team

Our insight will make taxi transportation more efficient for taxi companies (Vendor 1 and Vendor 2) to maximize their utilization by diverting cabs to the locations during specific times. Traffic planning, to use the model predictions for traffic management on specific day/time and location

Introduction

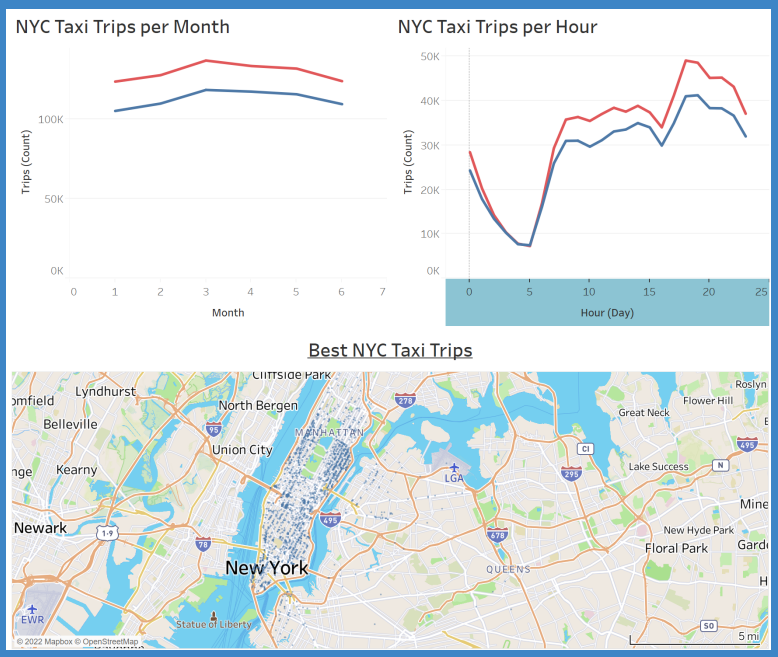
Dataset:

Rows: 729,322
Columns: 11

Target: Trip Duration

Categorical Variables :
ID
Pickup Datetime
Drop Off Datetime
Store and Forward Flag

Numerical /Continuous Variables:
Vendor ID
Passenger Count
Pickup Longitude
Pickup Latitude
Drop off Longitude



- Data Collection
- Data Cleaning
- Data Preprocessing
- Data Visualization
- Analysis & Results

Our team built a model that predicts the total ride duration of taxi trips in New York City. According to Kaggle, the dataset was released by NYC Taxi and Limousine Commission. Includes variables containing pickup/drop off location (longitude and latitude), time & duration by a NYC taxi company. The primary objective of this project is to predict the estimated time a taxi in NYC takes to reach the entered location given a specific location, date and time. A taxi company could use this prediction for developing policies and improving taxi distribution.

Data used:

/kaggle/input/nyc-taxi-trip-duration/nyc_taxi_trip_duration.csv

Variables Description:

- **ID:** It is a unique identifier for each trip
- **Vendor ID:** A code indicating the provider associated with the trip record.
- **Pick up Date time**: Date and time when taxi pick up passenger or meter was engaged.
- **Drop off Date time:** Date and time when the taxi drop off passenger or meter was disengaged.
- **Passenger count:** The number of passengers in the taxi.
- **Pick up Longitude:** An angular coordinate that defines the position of a point on a surface of earth where the meter was engaged while the passenger was picked.
- **Pick up Latitude:** The angle between the straight line in the certain point and equatorial plane where the meter was engaged while the passenger was picked.
- **Drop off Longitude:** An angular coordinate that defines the position of a point on a surface of earth where the meter was disengaged while the passenger was drop off.
- **Drop off Latitude:** The angle between the straight line in the certain point and equatorial plane where the meter was disengaged while the passenger was drop off.
- **Store and Forward Flag:** This flag indicates whether the trip log was held in vehicle memory before sending to the vendor due to the taxi not having connection to the server. Y - store and forward and N - not a store and forward trip.

Methodology: The present study will use regression and logistic model analysis in order to select the best model to predict taxi trip duration and also Random Forest and K-Nearest Neighbors Regression.

Data Collection

We collected our dataset to conduct a research about features, decision-making and strategy of NYC Taxi estimated time to reach the entered location.
The dataset was collected from Kaggle website (nyc_taxi_trip_duration.csv), and it contains two types of data: quantitative and qualitative.
With the use of Python and libraries such as pandas, and numpy, for data visualization we used Seaborn, Matplotlib, statistics, Sklearn and Tableau.

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import statistics as stats
import matplotlib.pyplot as plt
import statistics as stats
from sklearn import preprocessing
from sklearn.linear_model import LinearRegression
import statsmodels.api as sm
from sklearn.linear_model import LogisticRegression
import scipy as sp
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix
from sklearn.metrics import roc_curve,auc


df=pd.read_csv('nyc_taxi_trip_duration.csv')
```

Exploring the Dataset:

As we mentioned previously, the dataset contains 729,322 rows and 11 columns.

Number of columns and rows

```
#Number of columns and rows
print('Rows:',list(df.shape)[0])
print('Columns:',list(df.shape)[1])
```

```
Rows: 729322
Columns: 11
```

The data types we will be working on are numerical and categorical.

Datatype

```
df.dtypes
```

```
id                    object
vendor_id              int64
pickup_datetime       object
dropoff_datetime      object
passenger_count        int64
pickup_longitude     float64
pickup_latitude      float64
dropoff_longitude    float64
dropoff_latitude     float64
store_and_fwd_flag    object
trip_duration          int64
dtype: object
```

Data Preprocessing

Our data preprocessing was a step in the data mining and data analysis process that took raw data and transformed it into a format that is simple to understand and analyze.
We proceeded to clean up our data, verifying if we have null values in our dataset.

```
# % of missing values by columbs
percent_missing = df.isnull().sum() * 100 / len(df)
missing_value_df = pd.DataFrame({'column_name': df.columns,
                                 'percent_missing': percent_missing})
missing_value_df
```

	column_name	percent_missing
id	id	0.0
vendor_id	vendor_id	0.0
pickup_datetime	pickup_datetime	0.0
dropoff_datetime	dropoff_datetime	0.0
passenger_count	passenger_count	0.0
pickup_longitude	pickup_longitude	0.0
pickup_latitude	pickup_latitude	0.0
dropoff_longitude	dropoff_longitude	0.0
dropoff_latitude	dropoff_latitude	0.0
store_and_fwd_flag	store_and_fwd_flag	0.0
trip_duration	trip_duration	0.0

- There are no missing values in the dataset

	data_type	null_count	unique_count
id	object	0	729322
vendor_id	int64	0	2
pickup_datetime	object	0	709359
dropoff_datetime	object	0	709308
passenger_count	int64	0	9
pickup_longitude	float64	0	19729
pickup_latitude	float64	0	39776
dropoff_longitude	float64	0	27892
dropoff_latitude	float64	0	53579
store_and_fwd_flag	object	0	2
trip_duration	int64	0	6296

We proceed to convert into date format the following variables: pickup_datetime and drop off datetime, We checked our statistical (numerical) summary of our dataset.

```
df.head()
```

	id	vendor_id	pickup_datetime	dropoff_datetime
0	id1080784	2	2016-02-29 16:40:21	2016-02-29 16:47:01
1	id0889885	1	2016-03-11 23:35:37	2016-03-11 23:53:57
2	id0857912	2	2016-02-21 17:59:33	2016-02-21 18:26:48
3	id3744273	2	2016-01-05 09:44:31	2016-01-05 10:03:32
4	id0232939	1	2016-02-17 06:42:23	2016-02-17 06:56:31

Few insights from the summary:

Vendor ID has a minimum value of 1 and a maximum value of 2, meaning two vendor IDs, 1 and 2. Passenger count has a minimum of 0, which means is either entered by error or the driver entered deliberately to make believe the target number of rides completed.

```
#Summary statistics
df.describe()
```

	vendor_id	passenger_count	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	trip_duration
count	729322.000000	729322.000000	729322.000000	729322.000000	729322.000000	729322.000000	7.293220e+05
mean	1.535403	1.662055	-73.973513	40.750919	-73.973422	40.751775	9.522291e+02
std	0.498745	1.312446	0.069754	0.033594	0.069588	0.036037	3.864626e+03
min	1.000000	0.000000	-121.933342	34.712234	-121.933304	32.181141	1.000000e+00
25%	1.000000	1.000000	-73.991859	40.737335	-73.991318	40.735931	3.970000e+02
50%	2.000000	1.000000	-73.981758	40.754070	-73.979759	40.754509	6.630000e+02
75%	2.000000	2.000000	-73.967361	40.768314	-73.963036	40.769741	1.075000e+03
max	2.000000	9.000000	-65.897385	51.881084	-65.897385	43.921028	1.939736e+06

Drop Numerical Variables with Zero Variance

We proceed to analyze numerical variables with zero variance however we did not observe any numerical variable with zero variance.

```
# Checks if there is any variables with zero variance
df.std()
```

```
C:\Users\alex0\AppData\Local\Temp/ipykernel_7752/500058482.py:2: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with
'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select only valid columns before calling the reduction.
  df.std()

vendor_id                           0.498745
pickup_datetime      51 days 13:19:45.931836372
dropoff_datetime     51 days 13:20:24.060391979
passenger_count                     1.312446
pickup_longitude                    0.069754
pickup_latitude                     0.033594
dropoff_longitude                   0.069588
dropoff_latitude                    0.036037
trip_duration                    3864.626197
pickup_year                              0.0
pickup_month                        1.680661
pickup_day                          8.699772
pickup_hour                         6.402853
pickup_weekday                       1.95447
dropoff_year                             0.0
dropoff_month                       1.680815
dropoff_day                         8.699714
dropoff_hour                         6.48637
dropoff_weekday                     1.956866
dtype: object
```

- We can observe there is no numerical variable with zero variance.

Drop Categorical Variables with Zero Variance

We did not observe any categorical variables with zero variance

```
#Variance of each variable
variances = []
df_gpdescribe = pd.DataFrame(df.describe())
for i in df_gpdescribe.columns:
    variances.append([i,pow(list(df_gpdescribe[i])[2],2)])

df_variances = pd.DataFrame(variances).rename(columns={0:'Variable',1:'Variance'}).set_index(keys='Variable')
df_variances
```

Variable	Variance
vendor_id	2.487470e-01
passenger_count	1.722513e+00
pickup_longitude	4.865598e-03
pickup_latitude	1.128565e-03
dropoff_longitude	4.842512e-03
dropoff_latitude	1.298681e-03
trip_duration	1.493534e+07
pickup_year	0.000000e+00
pickup_month	2.824621e+00
pickup_day	7.568604e+01
pickup_hour	4.099653e+01
pickup_weekday	3.819952e+00
dropoff_year	0.000000e+00
dropoff_month	2.825140e+00
dropoff_day	7.568503e+01
dropoff_hour	4.207299e+01
dropoff_weekday	3.829325e+00

- We can observe there is no categorical variable with zero variance.

Remove Duplicates Using Variable Vendor ID

We did not observe any duplicate values.

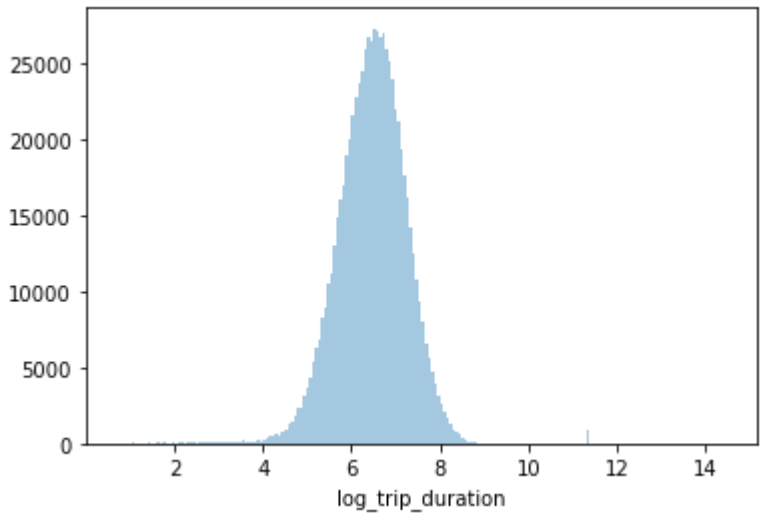
Remove duplicates Using Vendor Id

```
df.duplicated(subset=['id'])
```

```
0         False
1         False
2         False
3         False
4         False
          ...
729317    False
729318    False
729319    False
729320    False
729321    False
Length: 729322, dtype: bool
```

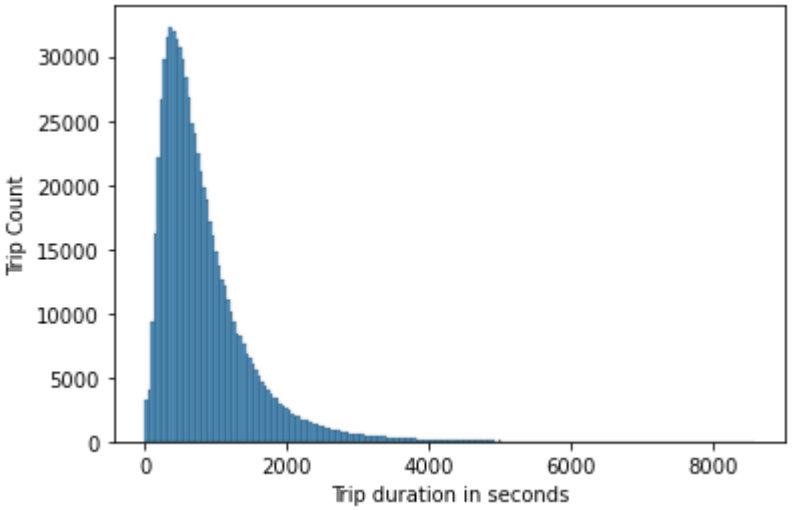
Data Visualization

By Taking Logarithm of Trip duration we can smoothen the data and figure out the distribution pattern: The distribution of the log trip duration is between 4-8, and the vertices are located around 6-7.



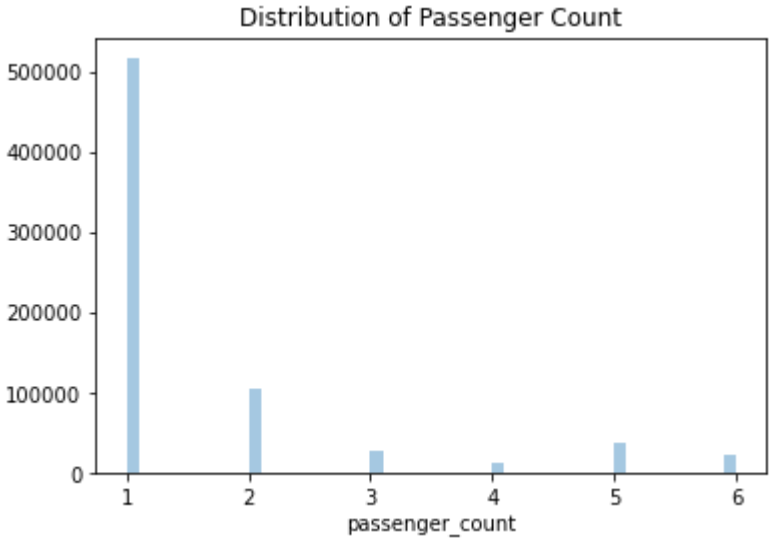
Trip Duration in Seconds Distribution:

The Distribution was shown in Right Skewed (Positive Skewness).



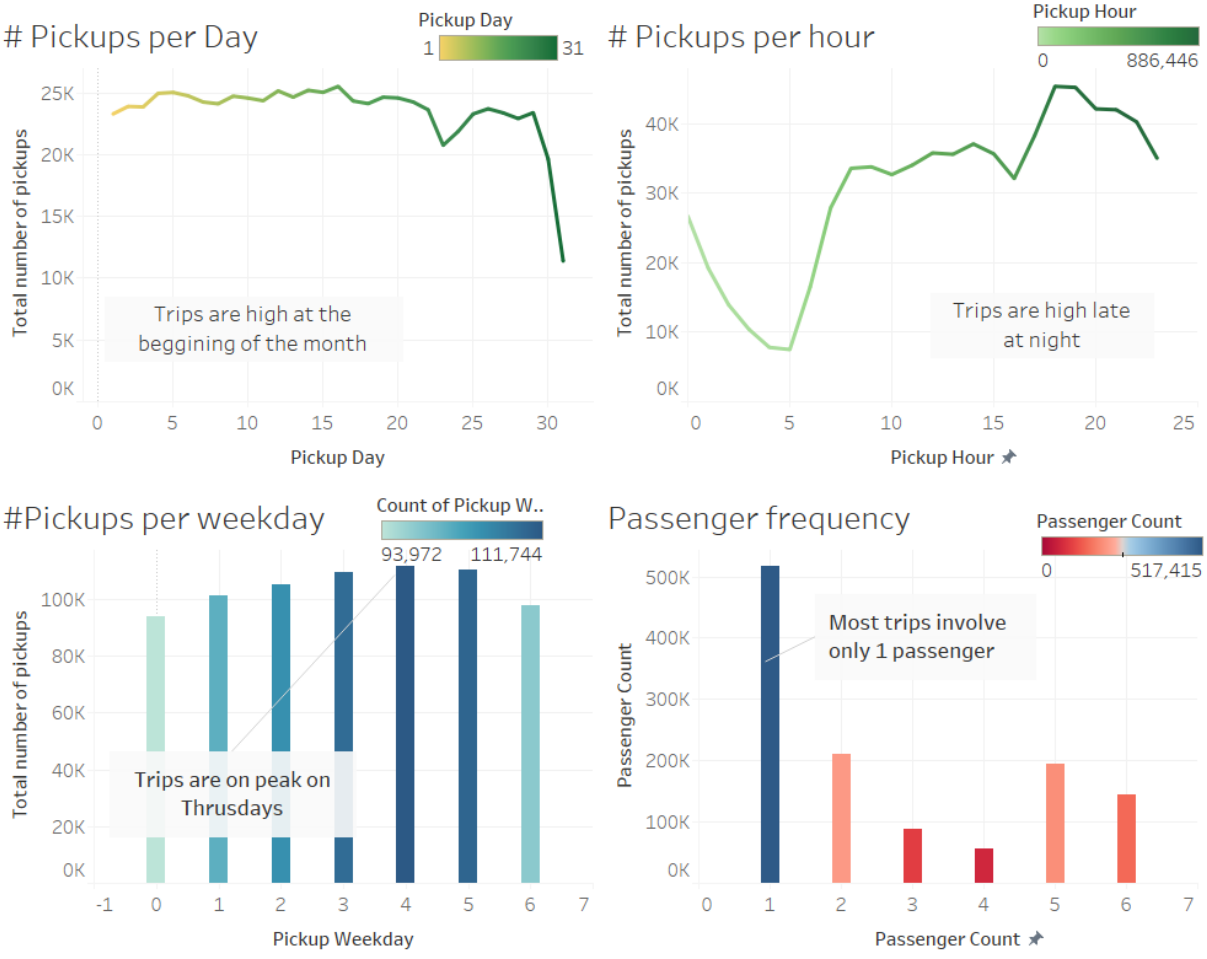
Passenger Count Distribution:

Most of the NYC Taxis have one passenger, we adjusted the passenger count scale and excluded outliers such as 7,8,9 passengers.



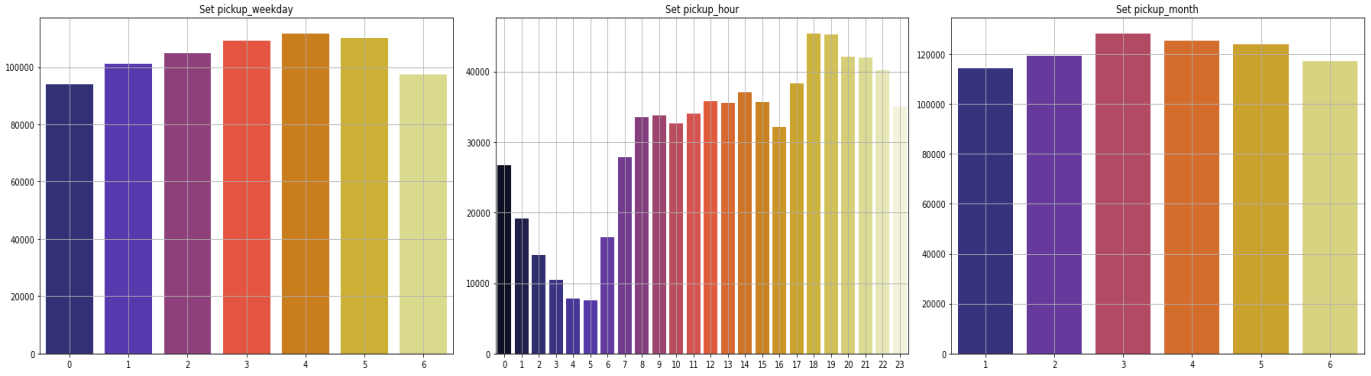
Pickups overview

For what we can observe the trips are high at the beginning of the month and are on peak on Thursdays. The trips are high late high late at night and most trips involve only 1 passenger.



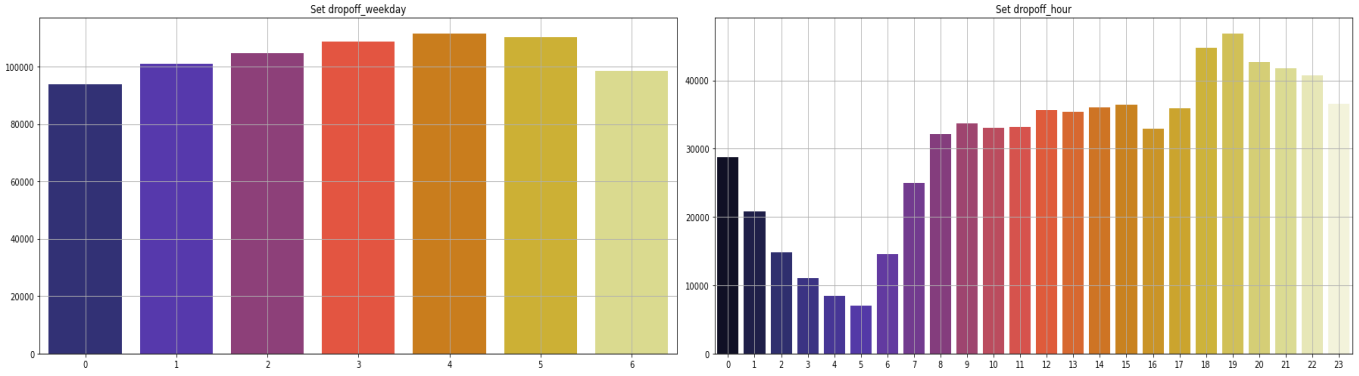
Count of occurences of each pickup day, pickup hour and pickup month:

Different Pickup Days in a Week, different pickup hours and different seasons/months all have different trip duration in NYC Taxi. Apparently, 2:00-6:00 AM is the quiet hours for NYC taxis.



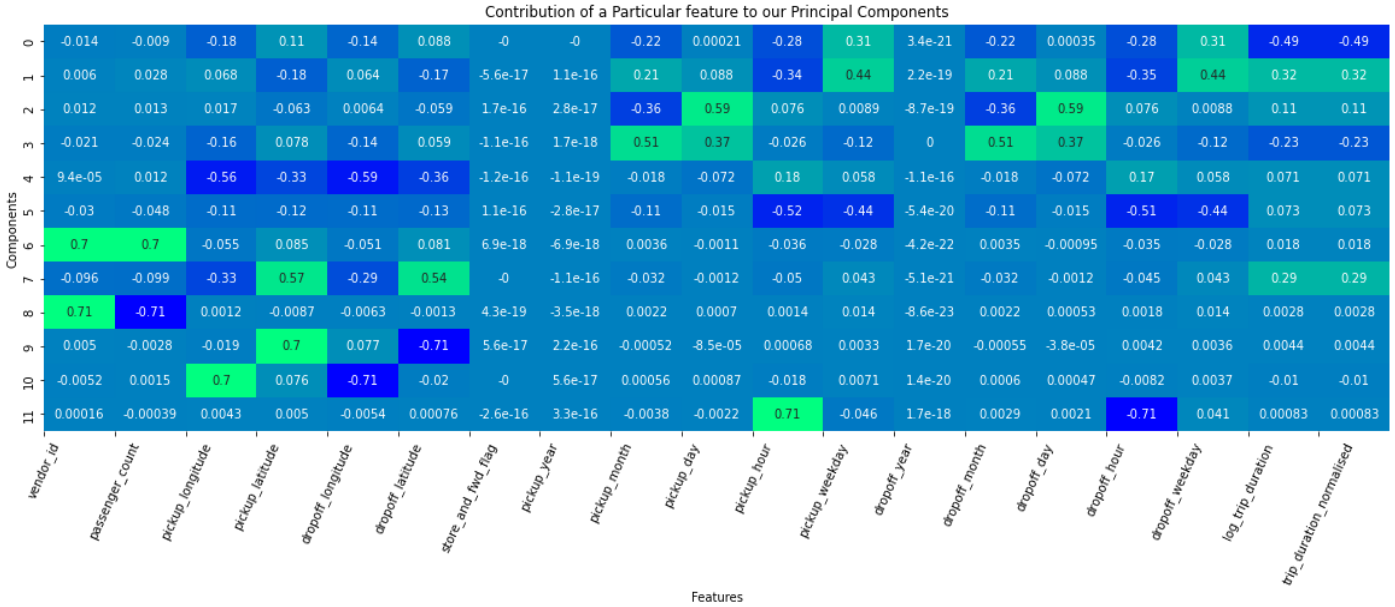
Count of occurences of each dropoff weekday and dropoff hour:

Dropoff hours distribution is very similar to the pickup hours as the rush hour and rush day are the same.

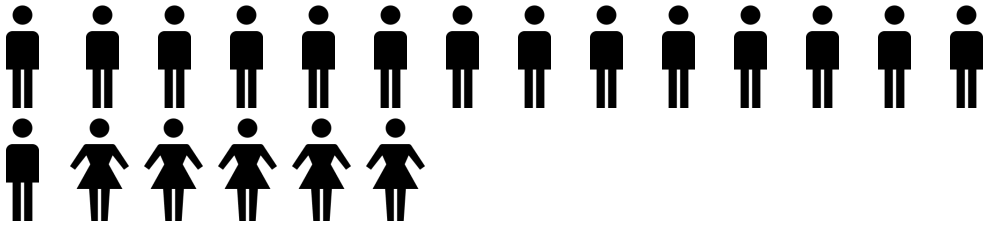


Importance of features in Particular Principal Component:

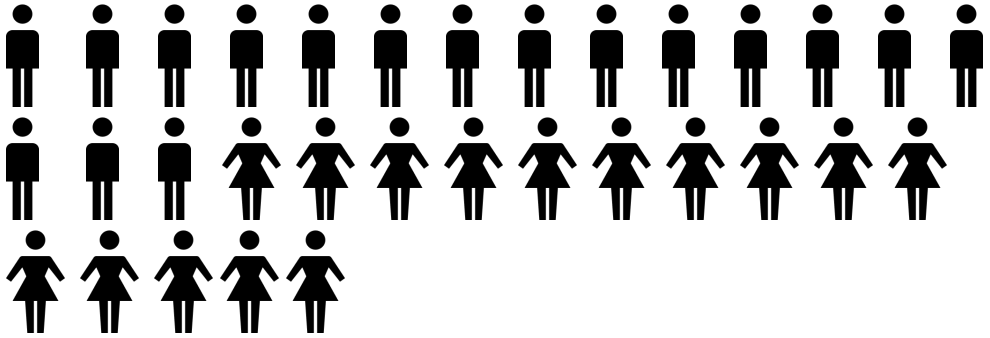
It's not hard to find the important features of each variables using the heatmap visualization.



Analysis & Results



29.05%
More than one passenger



70.95%
One passenger

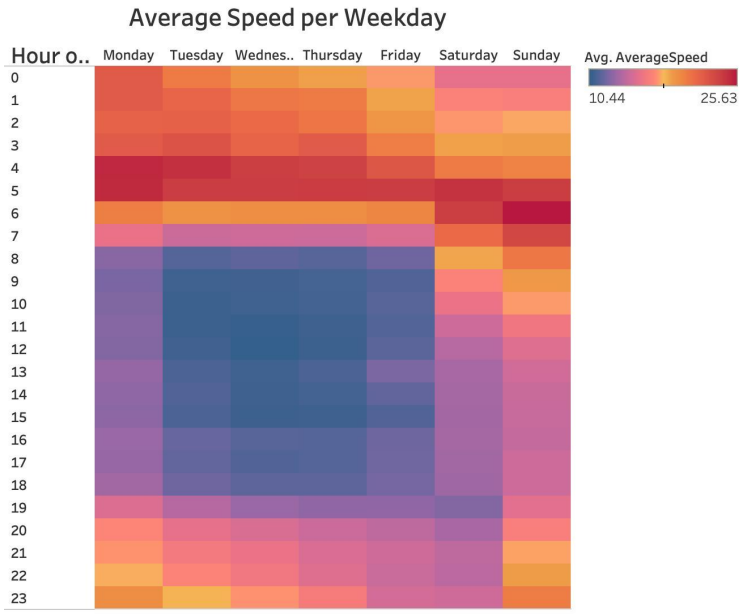
> If in New York, you arrive late for an appointment, say"I took a Taxi".

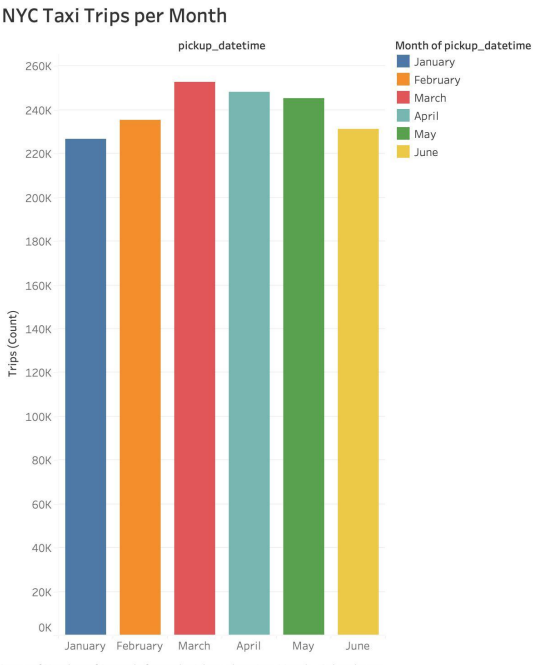
From our team's SMART goal, we know that our Specific is to develop a model to estimate a taxi ride in New York City, where we can investigate and analyze traffic patterns and later on with the information we would help taxi drivers to improve their length of drive in order to compete with the increasing Uber/Lyft demand. Our final goal of the project is to analyze taxi duration data, within 1 month, determine the Taxi trip duration will increase by 10% on a day-by-day basis compared to the same trip distance last week.

So in order to better analyze and achieve the specific goal and final goal, we create and build models to predict the total trip duration of taxi trips in New York City. We also take plenty of variables into our consideration as the independent variables in order to find the highest correlated variables of our independent variable–Taxi trip duration in New York City.

From this course we learnt, in the first step we identified the SMART Goal of this NYC Taxi data. Then we used Python and Excel to clean and preprocess the taxi data, which helped us to better analyze and predict the trip duration result and build a more accurate model. After we collected our final data set, we used the Tableau to create more visualizations to analyze how to achieve the final SMART goal and also used the visualizations to show the various factors that affect the dependent variable.

The results and prediction model could be clearly visualized by looking at the Tableau Storyboard and Dashboard. We found the Taxi trips duration in New York City was related to pick-up time, Month, Week Day and locations. Rush hours, Holiday season and Weekend/Weekday could all affect the trip duration in New York City.





Through all the analysis graphs, we can clearly see the result of our final smart goal. Trip duration is determined by various factors such as season, time, month and geographical location. We can find that Friday and Saturday are those days in a week when NewYorkers prefer Rome in the city, so trip duration will be affected on those days.

In March, more visitors will be in NYC during the spring holiday season and January being the lowest due to the cold weather and snowfalls. Trip duration will change according to different seasons and numbers of trip demands.

From analyzing the Hourly Pickup and Dropoff visualizations, we can see that Rush hours are 17:00 - 22:00 PM. During 2:00 -6:00 AM, people are sleeping at home. If one passenger takes a taxi during the rush hours, the trip duration will increase according to the traffic. If the taxi company wants to achieve the final goal, Taxi trip duration will increase by 10% on a day-by-day basis compared to the same trip distance last week within one month, the company needs to set target months as March and April for further discussion and search. NYC Taxi Company can also build a regression model to predict the trip duration according to rush hours, holiday seasons and weekdays. Such a prediction can help the company to better arrange and dispatch NYC taxis. Through our analysis, we learned a lot and discussed many interesting questions which helped us to get a better understanding about the SMART goal and data visualization, such as

- What types of Variables do we have ?
- Are there any False trips or Invalid data points which exceeds Trip Duration well above impossibility ?
- What's the most frequent travel destination ?

We also figured out that there were some limitations of our case study and analysis. We are working with very large data, even though we've done data cleaning, some of the visualization still look cluttered and complex. Since we are analyzing data from New York, we need more data from cities to verify our conclusions and analysis results in further research. In the future, we will collect more accurate, comprehensive and multi-dimensional data for analysis and discussion, and design more beautiful and easy-to-understand visualizations.

Conclusion

From our group's study and analysis of New York City taxi trip duration data, we found that there is a strong correlation between trip duration and taxi demand. Then we visualize the data for presentation and storytelling. Through data analysis and visual analysis, we finally find the prerequisites for achieving the ultimate goal and propose countermeasures for the profit growth of New York taxi companies.

Public Tableau Link:

https://public.tableau.com/app/profile/xinshi.li/viz/NYC_16611327270280/TripDuration#3