What is the purpose of predicting rides?

One reason is to **bridge the gap between demand and supply**. To make sure that the customers are assured a ride when they request for the same.

If you are able to identify some key locations based on the data where the (virtual) hubs/centers could be set up, customers wouldn't face difficulty in successfully booking a ride. Captains can be mobilised to be around these locations, during the peak demand times, so that there is no supply shortage and also the arrival/wait times can be minimised, resulting in lower cancellations due to longer wait times as well as the longer traffic times for the captains (win-win for both customers as well as captains).

Second one is to get an **estimate of revenue generated**. If the business model is such that the distance covered in a ride is one of the factors affecting revenue generation, the estimate helps in making business decisions

How do you do that ?

Given the ride request data, if it is made possible to predict the no. of rides that will be booked during a particular time period, in a particular place (or area), on a particular day, then the ride-sharing drivers can be intimated about it in advance.

By analysing the distances of each of the rides, a (timeseries) forecast model can be built that predicts the distance coverage on a particular day, during a particular time period and in a particular place (or area). And based on the X's pricing strategy the expected revenue generation can be forecasted.

Data Exploration

- Owing to the fact that people travel more during the starting of work hours, it can be observed that highest no. of rides are requested during Time Block 2 (i.e. 7 AM to 11 AM)

- If you go by Location Cluster wise, the highest number of rides are requested from Cluster 0 (BTM Layout cluster)

- The no. of rides requested are particularly low on weekends compared to weekdays

- An interesting observation is that on weekdays, the highest number of rides are requested during Time Block 2 where as on weekends , it is during Time Block 3. This makes sense as people do not travel in the morning hours on weekends as much as they do on weekdays.

Data & Algorithm

- Out of nearly 8 million records, I have selected 1 million for analysis and modeling. It constitutes ride requests by customers over a period of about 4 months.
- The data is filtered to restrict the location within Bangalore city. An additional column determining the distance between pickup and drop location in km is added to the dataframe.

- First, the 'Time' column is classified into 5 blocks -
  00:00:00 - 07:00:00 - Block 1
  07:00:00 - 11:00:00 - Block 2
  11:00:00 - 17:00:00 - Block 3
  17:00:00 - 21:00:00 - Block 4
  21:00:00 - 24:00:00 - Block 5
  Each ride request is labelled accordingly

- It can be seen that there is a record whenever a customer requests a ride. So, if a customer is not able to book a ride successfully in the first attempt, multiple attempts are recorded within the same time period. These duplicate records are removed in the second step and stored in a different dataframe for analysis

- Now, all the Pickup locations (pick_lat, pick_lng) are clustered into 4 clusters using k-means clustering. The centroids of these clusters can be located on Google Maps to see that they lie in:

	Lat	Lng	
0	12.915261	77.614580	- BTM 2nd Stage, BTM Layout
1	12.982377	77.624994	- Kalhalli, Halasuru
2	12.956665	77.690554	- Jawahar Nagar, Marathahalli
3	12.966838	77.562414	- Nagamma Nagar, Cottonpete

  Each ride request is given 'location cluster number' label depending on the prediction of which cluster they fall into.

- Once the above labelling process is over, the whole data is segregated into different sections depending on time block and location cluster to find the no. of rides as well as

the total distance travelled. There are 4 subsections of locations for each time block (there are 5 time blocks) in a day.

- After this, I used '*Autoregressive Integrated Moving Average*' statistics model for Time Series forecasting. This model is applied to 20 different sections (as mentioned above) to predict the '**no. of rides requested' and 'total distance covered' on a particular day during a particular time period in a particular location cluster.**
  There are other time series forecasting methods which can be applied as well.

Results & Conclusion:

For time series forecasting, I have implemented two different models:

1. ARIMA
2. VAR - Vector Auto Regressive

The difference between the two is that in case of VAR, the dependency between the time-dependent variables is also considered while forecasting.

Comparison of the results of these two algorithms implemented on the first subsection (where Time Block: 1 and Location Cluster: 0 are selected)

Statistics of 'Count'

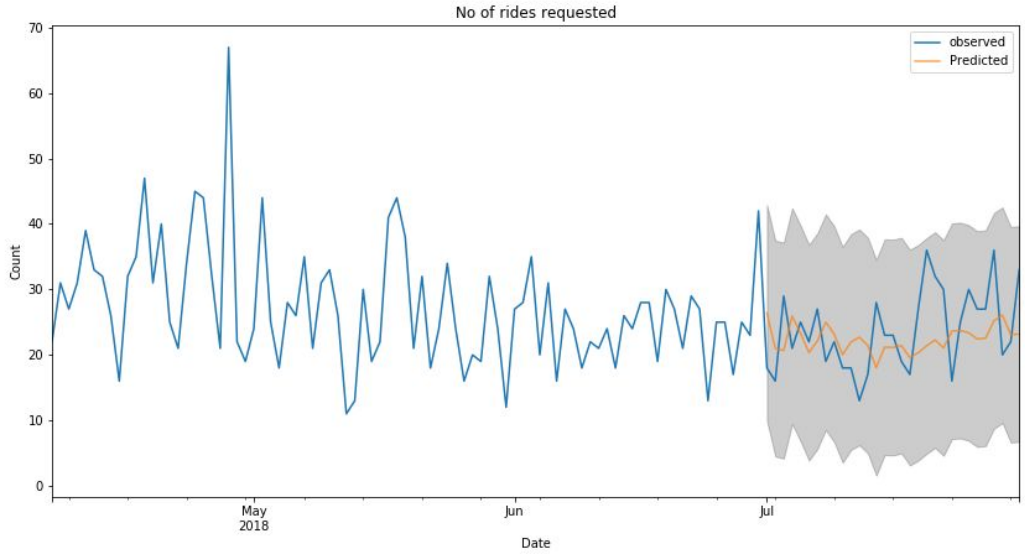
count	116.000000
mean	26.318966
std	8.583705
min	11.000000
25%	20.750000
50%	25.000000
75%	31.000000
max	67.000000

Statistics of 'Distance'

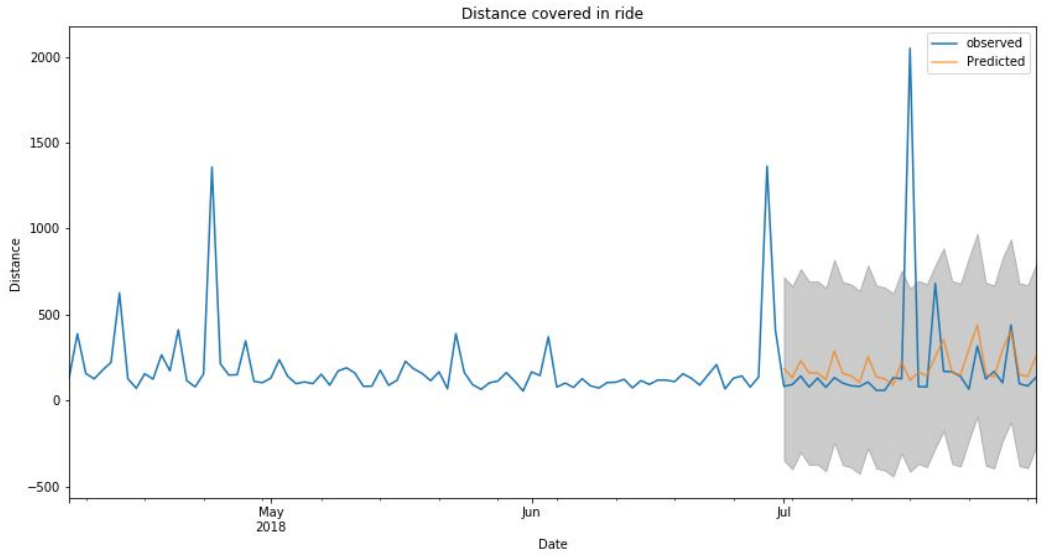
count	116.000000
mean	187.826287
std	256.829107
min	54.530295
25%	91.781435
50%	125.623275
75%	165.739892
max	2048.997777

ARIMA

Count:
RMSE: 6.57

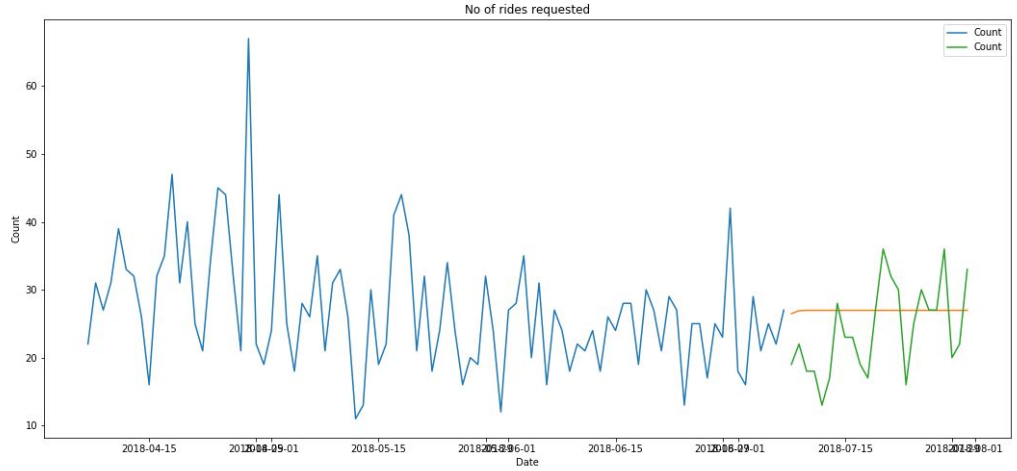


Distance:
RMSE: 367.71

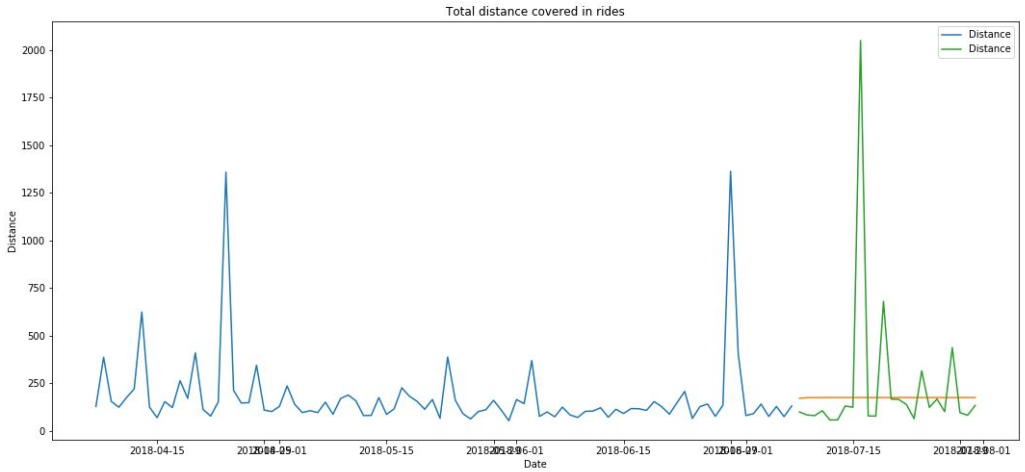


VAR

Count:
RMSE : 6.98



Distance:
RMSE: 406.56



Similar comparison can be done for other subsections as well

Observations:

- Performance using ARIMA is better than VAR
- The three peaks in the Distance graph correspond to:

Date	Count	Distance
2018-07-16	23.0	2048.997777
2018-06-29	23.0	1362.995626
2018-04-24	45.0	1358.407085

This could be because of some economic factors.

- Predicted values for ‘no. of rides’ are not in whole numbers. These can be rounded off to the nearest integer value.

- The process of applying time forecasting model for each subsection (also for two time-dependent variables) is time-consuming and laborious

What could be done given more time?

- **Time block classification:** Assuming an average ride time of 30-45 min, it'll be ideal to take time blocks of 2 hr each from morning to evening, so as to improve the accuracy of predictions & on ground optimisations (achieving demand-supply equilibrium), like ride availability & wait times.

- **Location_Cluster classification:** Could process more than 4 location clusters, for the virtual hubs

- Process more than 1 million rows of raw data

- Could have analysed duplicate records