Data Scientist at Locale

# Introduction

I have completed the Data Science task from the given tasks in Locale interview.  
There are 2 main python files which will be used as surge calculator, surge.py and predictor.py.   
Another important file is ‘cancellation\_model.sav’ which stores the trained model for predicting if the user might cancel the ride.  
Next, there is an excel file, ‘fin.xlsx’ which is a modified version of ‘data.xlsx’ and is used to calculate surge.  
A file named ‘main.ipynb’ and contains code for EDA. This file is not as extensively commented out since this was written on a prototype basis.

# Algorithm Used to calculate Surge

The algorithm is fairly simple to understand. First it loads up the data and then makes up two groupings. First is made on basis of ['week', 'from\_area\_id', 'start\_hour'] where week is week of year, from\_area\_id is the starting point of travel and start hour is the hour in which ride will start.  
Second grouping is made on the basis of ['week', 'from\_area'] and are same as mentioned above.  
  
Next, we calculate the number of rides in that particular hour, one hour before and one hour later in the past four weeks. For eg: if the hour is 15, we will not only calculate the rides in the 15th hour but also in the 14th and 16th hour.  
The number of rides in 14th and 16th hour will be multiplied by 0.5 to give a lower weightage to them and for a particular week, the sum will be divided by 2 because  
0.5(x-1) + x + 0.5(x+1).  
This is repeated for the past four weeks and is divided by four to get average number of rides from an area in a particular hour from the past four weeks. Let’s call this value ‘**X**’.

Now, we will use the second grouping and will calculate the average number of rides in the past four weeks. The algorithm is same as above but for a particular week, the total rides are divided by 24 to get each hour’s average ride and then in the end, it’s divided by 4 to get each week’s hourly average ride from an area. Let’s call this value ‘**Y**’.

Now, the result is calculated as X/Y. if the value is less than or equal to 1, no surge fee is charged. If it’s more than 1, we divide the value by 6 and the result is added to 1 and that becomes the surge charge.   
One condition is also included which includes predicting cancellation. If the cancellation model predicts the user might cancel the ride. The extra surge is divided by 2 and then added to 1 and returned as surge charge.

This helps the company reduce cancellations and hence, increasing revenue.

# Future Developments

This was the first iteration of what a surge price model might be for a company.  
 A few ideas on how it can be improved.

Instead of taking area as fixed unit, we can forget about area and if data can be provided for driver’s location, a surge fee can be calculated on that basis.

Instead of using Area-Id, a particular location can be used and rider’s location can be mapped with the driver and based on that, using the distance and already mentioned algorithm to find user’s surge.

One thing that I missed here is using spatial data. This can be used in continuation with the previous point.

Next, this model works for point to point travel since outstation and rental do not have enough data point to make a sustainable model and might even increase cancellation instead of increasing revenue.

**Note:** Please run the python code in 64-bit software. Please refer to this, <https://stackoverflow.com/questions/21033038/scikits-learn-randomforrest-trained-on-64bit-python-wont-open-on-32bit-python>