Joanna Riascos

Dr. Sylvain Jaume

Capstone Course

March 30th, 2017

Lab 9 Apply aggregation and group operations to your dataset

1. Complete reference to the IEEE or ACM paper you are trying to outperform. Only articles published by IEEE or ACM will be accepted.

"A Cloud-Based Smart-Parking System Based On Internet-Of-Things Technologies - IEEE Xplore Document". Ieeexplore.ieee.org. N. p., 2017. Web. 4 Feb. 2017.

2. Describe the results of the paper you have chosen.

The results depict that the algorithm reduces the average waiting time of users for parking their cars.

3. Describe your results so far, and your next objective in terms of data analysis

Our results show that we have implemented the multiple regression analysis. We have also created the RDD pairs of our dataset. Next, we went ahead and plotted our model to depict the residuals. Afterwards, we created some histograms using our variables.

Our next objective is to perform cross validation so that we can visualize the accuracy of our model.

4. Describe how you will use the concepts of aggregation and group operations

The concepts of aggregation and group operations is very useful because we can find another way to merge the ozone layer data field to our parking dataset. We can also visualize and group our data fields to analyze the impact of each of the variables.

5. Create a table showing the code for every operation in the left column and the time measurements for every operation in the right column. You need at least 10 operations. Highlight in bold the lines of code where you use aggregation or group operation concepts.

parking['tip_pct'] = parking['vehiclecount'] /parking['totalspaces']	1. Division (1 min)
grouped = parking.groupby(['vehiclecount','totalspaces'])	2. Group by (1 min)
functions = ['count','mean','max', 'min', 'median']	3. Count, 4. Mean, 5. Max, (2 mins) 6. Min, 7. Median
result = grouped['tip_pct','vehiclecount'].agg(functions) result	8. Aggregate (1 min)
parkingnames = DataFrame(np.random.randn(4,4), columns=['a','b','c','d'], index=['vehiclecount', 'totalspaces', 'garagecode', 'ozone']) parkingnames.ix[2:3, ['b','c']] = np.nan	9. Grouping with dictionaries and series (2 mins)
parkingnames.groupby(len).sum() **key_list = ['vehiclecount', 'totalspaces', 'garagecode', 'ozone'] parkingnames.groupby([len, key_list]).min()**	10. Sum (1 min)