**SI 618 Fall 2017 Homework 3 (100 points)**

Data to be used in this homework: On the Fladoop cluster, I have put the following two files in HDFS:

hdfs:///var/si618f17/yelp\_academic\_dataset\_business.json  
hdfs:///var/si618f17/yelp\_academic\_dataset\_review.json

These files were downloaded from <http://www.yelp.com/dataset_challenge> (you cannot share the content with others without going through the approval procedure yourself). The format of the data is explained in the “Notes on the Dataset” section at <http://www.yelp.com/dataset_challenge>

Note that you do not need to download the Yelp dataset yourself as it is already put into HDFS on the Fladoop cluster.

Here you will only use: hdfs:///var/si618f17/yelp\_academic\_dataset\_business.json

The goal is to compute the number of businesses, total review count, and number of 4-star or higher reviews for each neighborhood in each city. If a business has multiple neighborhoods, its review count and stars should be attributed to all of the neighborhoods. If the neighborhoods list is empty, then we will use 'Unknown' as the name of the neighborhood.

Your final result should be a TSV file that is the same as the provided hw3\_desired\_output.tsv file.

In this desired output file, each row contains 5 columns, which are separated by a tab. For example, consider this following row:

Charlotte University City 312 7164 114

This means the neighborhood of “University City” in the city of “Charlotte” has 312 businesses, and their total review count is 7164, and 114 of the reviews are 4 stars or more.

The rows in the output file should be sorted in alphabetical order of the city names, and the neighborhoods in each city are sorted by the number of businesses in decreasing order, then by the number of review count in decreasing order, and then by the reviews with 4 or more stars in decreasing order. Lastly, in case all the numbers are equal, we want the neighborhoods sorted in alphabetical order. Note that there are some cities with names that may be data entry errors (such as “110 Las Vegas”); to match the desired output you should leave these cities in your output and not perform any filtering to remove cities with strange or incorrect names.

You MUST use Spark to do this homework. A non-Spark solution will not get any credit. HINT: You can modify from the provided example code spark\_avg\_stars\_per\_category.py. Save it as “si618\_f17\_hw3\_youruniquename.py”

**What to submit:**

Submit a zip file named si618\_f17\_hw3\_youruniqname.zip containing

* si618\_f17\_hw3\_uniquename.py
* si618\_f17\_hw3\_output\_uniquename.tsv