Assignment 1

MET CS 777 - Big Data Analytics
Spark Data Wrangling
(20 points)

GitHub Classroom Invitation Link
`https://classroom.github.com/a/80nVkWN-`

1 Description

The goal of this assignment is to implement a set of Spark programs in python (using Apache Spark). Specifically, your Spark jobs will analyzing a data set consisting of New York City Taxi trip reports in the Year 2013. The dataset was released under the FOIL (The Freedom of Information Law) and made public by Chris Whong (`https://chriswhong.com/open-data/foil_nyc_taxi/`).

2 Taxi Data Set

The data set itself is a simple text file. Each taxi trip report is a different line in the file. Among other things, each trip report includes the starting point, the drop-off point, corresponding timestamps, and information related to the payment. The data are reported by the time that the trip ended, i.e., upon arrive in the order of the drop-off timestamps. The attributes present on each line of the file are, in order:

	Attribute	**Description**
0	medallion	an md5sum of the identifier of the taxi - vehicle bound (Taxi ID)
1	hack_license	an md5sum of the identifier for the taxi license (Driver ID)
2	pickup_datetime	time when the passenger(s) were picked up
3	dropoff_datetime	time when the passenger(s) were dropped off
4	trip_time_in_secs	duration of the trip
5	trip_distance	trip distance in miles
6	pickup_longitude	longitude coordinate of the pickup location
7	pickup_latitude	latitude coordinate of the pickup location
8	dropoff_longitude	longitude coordinate of the drop-off location
9	dropoff_latitude	latitude coordinate of the drop-off location
10	payment_type	the payment method -credit card or cash
11	fare_amount	fare amount in dollars
12	surcharge	surcharge in dollars
13	mta_tax	tax in dollars
14	tip_amount	tip in dollars
15	tolls_amount	bridge and tunnel tolls in dollars
16	total_amount	total paid amount in dollars

Table 1: Taxi Data Set fields

The data files are in comma separated values (CSV) format. Example lines from the file are:

07290D3599E7A0D62097A346EFCC1FB5,E7750A37CAB07D0DFF0AF7E3573AC141,
2013-01-01,00:00:00,2013-01-01 00:02:00,120,0.44,-73.956528,40.716976,-73.962440,
40.715008,CSH,3.50,0.50,0.50,0.00,0.00,4.50

22D70BF00EEB0ADC83BA8177BB861991,3FF2709163DE7036FCAA4E5A3324E4BF,
2013-01-01,00:02:00,2013-01-01 00:02:00,0,0.00,0.000000,0.000000,0.000000,0.000000,
CSH,27.00,0.00,0.50,0.00,0.00,27.50

0EC22AAF491A8BD91F279350C2B010FD,778C92B26AE78A9EBDF96B49C67E4007,
2013-01-01,00:01:00,2013-01-01 00:03:00,120,0.71,-73.973145,40.752827,-73.965897
73.965897,40.760445,CSH,4.00,0.50,0.50,0.00,0.00,5.00

You can use the following PySpark Code to cleanup the data and get the required field.

```
lines = sc.textFile(sys.argv[1])
taxilines = lines.map(lambda x: x.split(','))

# Exception Handling and removing wrong data lines
def isfloat(value):
    try:
        float(value)
        return True
    except:
        return False

# For example, remove lines if they don't have 16 values and ...
def correctRows(p):
    if(len(p) == 17):
        if(isfloat(p[5]) and isfloat(p[11])):
          if(float(p[5])!=0 and float(p[11])!=0):
              return p

# cleaning up data
texilinesCorrected = taxilines.filter(correctRows)
```

You can also pre-process the data and store it in your own cluster storage.

3 Obtaining the Dataset

Small data set. (93 MB compressed, uncompressed 384 MB) for implementation and testing purposes (roughly 2 million taxi trips). This is available at Amazon S3:
`https://s3.amazonaws.com/metcs777/taxi-data-sorted-small.csv.bz2`

You can download or access the data sets using the following internal URLs:

	Google Cloud
Small Data Set	gs://metcs777/taxi-data-sorted-small.csv.bz2
Large Data Set	gs://metcs777/taxi-data-sorted-large.csv.bz2

Table 2: Data set on Google Cloud Storage - URLs

	Amzon AWS
Small Data Set	s3://metcs777/taxi-data-sorted-small.csv.bz2
Large Data Set	s3://metcs777/taxi-data-sorted-large.csv.bz2

Table 3: Data set on Amazon AWS - URLs

4 Assignment Tasks

4.1 Task 1 : Top-10 Active Taxis (5 points)

Many different taxis have had multiple drivers. Write and execute a Spark Python program that computes the top ten taxis that have had the largest number of drivers. Your output should be a set of (medallion, number of drivers) pairs.

Note: You should consider that this is a real world data set that might include wrongly formatted data lines. You should clean up the data before the main processing, a line might not include all of the fields. If a data line is not correctly formatted, you should drop that line and do not consider it.

4.2 Task 2 - Top-10 Best Drivers (7 Points)

We would like to figure out who the top 10 best drivers are in terms of their average earned money per minute spent carrying a customer. The total amount field is the total money earned on a trip. In the end, we are interested in computing a set of (driver, money per minute) pairs.

4.3 Task 3 - Best time of the daty to Work on Taxi (8 Points)

We would like to know which hour of the day is the best time for drivers that has the highest profit per miles. Consider the surcharge amount in dollar for each taxi ride (without tip amount) and the distance in miles, and sum up the rides for each hour of the day (24 hours) – consider the pickup time for your calculation. The profit ratio is the ration surcharge in dollar divided by the travel distance in miles for each specific time of the day.

Profit Ratio = (Surcharge Amount in US Dollar) / (Travel Distance in miles)

We are interested to know the time of the day that has the highest profit ratio.

4.4 Task 4 - (For Advanced Students – no points)

Here are two further tasks for advanced groups.

- How many percent of taxi customers pay with cash and how many percent using electronic cards? Analyze these payment methods for different time of the day and provide a list of percents for each day time? As a result provide two numbers for total percentages and a list like (hour of day, percent paid card)
- We would like to measure the efficiency of taxis drivers by finding out their average earned money per mile. (Consider the total amount which includes tips, as their earned money) Implement a Spark job that can find out the top-10 efficient taxi divers.
- What are mean, median, first and third quantiles of tip amount? How do find the median?
- Using the IQR outlier detection method find out the top-10 outliers.

5 Important Considerations

5.1 Machines to Use

One thing to be aware of is that you can choose virtually any configuration for your EMR cluster - you can choose different numbers of machines, and different configurations of those machines. And each is going to cost you differently!

Pricing information is available at: `http://aws.amazon.com/elasticmapreduce/pricing/`

Since this is real money, it makes sense to develop your code and run your jobs locally, on your laptop, using the small data set. Once things are working, you'll then move to Amazon EMR.

We are going to ask you to run your Spark jobs over the "real" data using 3 machines with **4 cores and 8GB RAM each** as workers. This provides 4 cores per machine (16 cores total) so it is quite a bit of horsepower. On the Google cloud take 4 machines with 4 cores and 8 GB of memory.

As you can see on EC2 Price list , this costs around 50 cents per hour. That is not much, but **IT WILL ADD UP QUICKLY IF YOU FORGET TO SHUT OFF YOUR MACHINES**. Be very careful, and stop your machine as soon as you are done working. You can always come back and start your machine or create a new one easily when you begin your work again. Another thing to be aware of is that Amazon charges you when you move data around. To avoid such charges, do everything in the "N. Virginia" region. That's where data is, and that's where you should put your data and machines.

- You should document your code very well and as much as possible.
- You code should be compilable on a unix-based operating system like Linux or MacOS.

5.2 Academic Misconduct Regarding Programming

In a programming class like our class, there is sometimes a very fine line between "cheating" and acceptable and beneficial interaction between peers. Thus, it is very important that you fully understand what is and what is not allowed in terms of collaboration with your classmates. We want to be 100% precise, so that there can be no confusion.

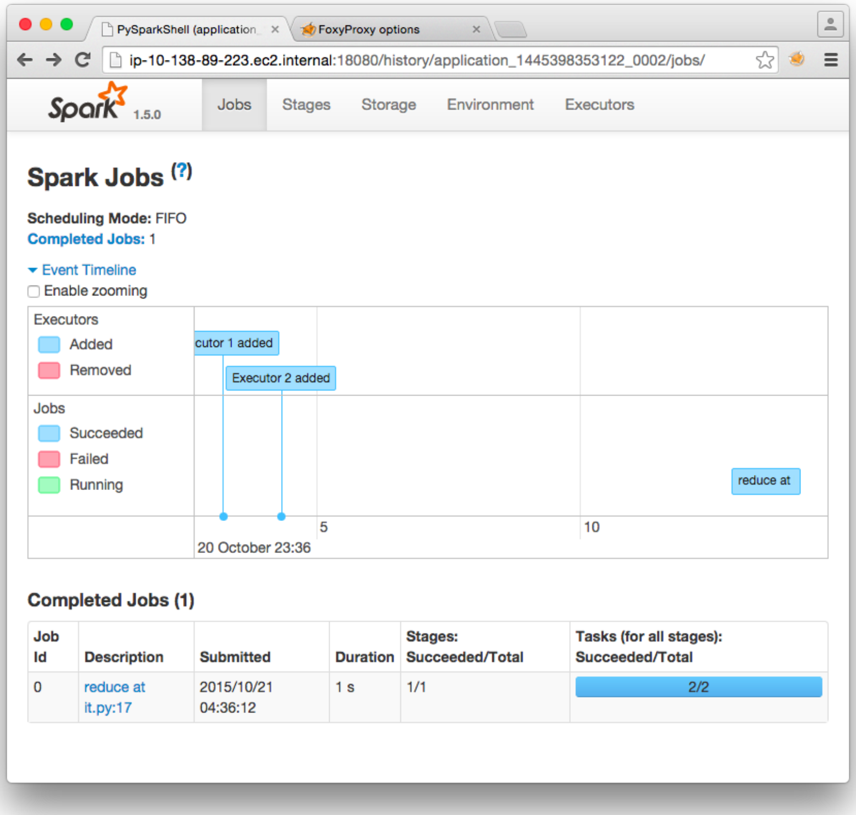
The rule on collaboration and communication with your classmates is very simple: you cannot transmit or receive code from or to anyone in the class in any way—visually (by showing someone your code), electronically (by emailing, posting, or otherwise sending someone your code), verbally (by reading code to someone) or in any other way we have not yet imagined. Any other collaboration is acceptable.

The rule on collaboration and communication with people who are not your classmates (or your TAs or instructor) is also very simple: it is not allowed in any way, period. This disallows (for example) posting any questions of any nature to programming forums such as **StackOverflow**. As far as going to the web and using Google, we will apply the **"two line rule"**. Go to any web page you like and do any search that you like. But you cannot take more than two lines of code from an external resource and actually include it in your assignment in any form. Note that changing variable names or otherwise transforming or obfuscating code you found on the web does not render the "two line rule" inapplicable. It is still a violation to obtain more than two lines of code from an external resource and turn it in, whatever you do to those two lines after you first obtain them.

Furthermore, you should cite your sources. Add a comment to your code that includes the URL(s) that you consulted when constructing your solution. This turns out to be very helpful when you're looking at something you wrote a while ago and you need to remind yourself what you were thinking.

5.3 Turnin

Create a single document that has results for all three tasks. For each task, copy and paste the result that your last Spark job wrote to Amazon S3. Also for each task, for each Spark job you ran, include a screen shot of the Spark History.



Spark Jobs (?)

Scheduling Mode: FIFO
Completed Jobs: 1

☐ Enable zooming

Completed Jobs (1)

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
0	reduce at it.py:17	2015/10/21 04:36:12	1 s	1/1	2/2

Figure 1: Screenshot of Spark History

Please zip up all of your code and your document (use .zip only, please!), or else attach each piece of code as well as your document to your submission individually.

Please have the latest version of your code on the GitHub. Zip the files from GitHub and submit as your latest version of assignment work to the Blackboard. We will consider the latest version on the Blackboard but it should exactly match your code on the GitHub