Big Data Computing
Project 1 Report

Jeremy Martinez, Rosa Sierra, Enmanuel Aquino

Important Notes:

- For both of the files, put the data under the folder “mapreduce-test-data”.
- For the part 1 (parking violations) save the data file under the name “nyc_parking_violations_data.csv”
- For the second part (NBA Shot Logs) save the data file under the name “shot_logs.csv”.
- All the content of the zip file can also be accessed on the github link: https://github.com/rosamsierrap/Project1.git

NY PARKING VIOLATIONS

For the results of the NY Parking violations, run the bash file part1.sh

• When are tickets most likely to be issued?

A mapreduce framework is used to determine the hour of the day with more tickets issued. In the map part (see file hour_mapper.py), the <key, value> pair is constructed using <hour, 1>. But, first a validation is applied to the hour column of our file to format it as AM or PM, printed as A/P in our code. Then, 1 is added as a value for the aggregating part in the reducer.

A common reducer is used for all the questions, in which similar to a word count the key pair <hour,1> is aggregated generating the key pair <hour, value>. Then the output is sorted in descending order and only the first 10 pairs are printed on the screen.

Based on the chunk of data tested, the tickets are most likely to be issued at 04:00 PM.

```
-------------------------- Question 1: At what hour of the day are tickets most likely to be issued? -
----
04P     77
12P     76
01P     30
07A     29
08A     20
11A     18
10A     17
04A     15
10P     15
03A     13
```

• What are the most common years and types of cars to be ticketed?

The mapper for the common years (see cyear_mapper.py) and types of cars (see ctype_mapper.py) as in the Q1 is based on the format <key, 1>, being in this case <car_year, 1> and <car_type, 1>. As described in the Q1, the common reducer is used to output the top 10 count for each one of these.

Based on the dataset used, the most common car year ticketed is 2017, and the most common type of car to be ticketed is suburban.

```
-------------------------- Question 2.1: Most common year to be ticketed ----
2017    29
2015    27
2020    27
2018    24
2021    21
2019    19
2016    18
2011    17
2013    15
2012    15
```

```
-------------------------- Question 2.2: Most common type to be ticketed ----
SUBN    168
SDN     139
4DSD    41
VAN     20
CONV    6
P-U     5
PICK    5
SW      4
TAXI    3
2DSD    2
```

• Where are tickets most commonly issued?

The mapper for the common state (see state_mapper.py) used was in the format <state, 1>, using the column “Registration State” of the dataset. The common reducer was used to output the top 10 states.

Based on the output with the tested dataset, the most common state to be ticketed is NY.

```
-------------------------- Question 3: Most common state to be ticketed (where) --
NY      310
NJ      23
PA      20
CT      7
FL      6
GA      6
MA      4
TX      3
NC      3
CA      3
```

• Which color of the vehicle is most likely to get a ticket?

The mapper for the color of the vehicle (see color_mapper.py) is in the format <color, 1>. But, before making the <key, value> pair, a kind of data "cleaning" was made on the "Vehicle Color" column of the dataset, some colors with different identification keys on the column were aggregated as one color. For example, the color black appears as: ("BLK","BK","BLACK") in the dataset. As all the questions, the common reducer was used to output the top 10 colors of cars ticketed.

In our tested dataset, the color of vehicle most likely to get a ticket is gray.

```
--------------------------- Question 4: Most common color to be ticketed ----------
GRAY    98
BLACK   94
WHITE   74
RED     26
BL      24
BLUE    13
SILVER  7
GREEN   5
YELLOW  3
BR      3
```

Some considerations…

When answering questions about which category is "most likely" to receive a ticket in this dataset, it is important to note that we are basing our answers on the frequency of tickets issued. For example, if we are asked, "Which color of vehicle is most likely to receive a ticket?" and the frequency of gray vehicles being ticketed is higher than other colors, it does not necessarily mean that gray cars are inherently more likely to receive a ticket. There may simply be more gray cars in the population overall, leading to a higher frequency of tickets being issued to gray vehicles.

To measure "most likelihood," we would need additional data about the total number of vehicles of each color in New York City and how many of each color were ticketed. Without this additional information, it is difficult to draw definitive conclusions about the likelihood of a specific category receiving a ticket.

NBA SHOT LOGS

Most Unwanted Defender

For each pair of the players (A, B), we define the fear score of A when facing B is the hit rate, such that B is the closest defender when A is shooting. Based on the fear score, for each player, please find out who is his ”most unwanted defender”.

To implement the most unwanted defender map-reduce framework, we run a single round of map and reduce. In the mapping phase, we construct pairs of (shooter_id, defender_id) and a value "made," where "made" is 1 if the shot was successful and 0 if it missed. The reducer then creates a dictionary that counts the number of shot attempts and successful shots for each defender. We then calculate the percentage of successful shots for each defender when they were the closest defender. The top 10 shooter/defender pairs with the lowest percentage are then identified as the most unwanted defenders.

Based on our sample data, the player pairs with the highest fear score are Dennis Schroder vs Jerryd Bayless (203471-201573) and Norris Cole vs Goran Dragic (202708-201609), with a score of 0.00% in both cases. This means that out of 10 shots made 0 were successful for these two pairs, making them the most unwanted defenders for their respective shooters, being Schroder and Dragic the defenders.

```
--------------------------- Question 1: Most unwanted defender ranking ---
----
203471-201573   0.00%   10
202708-201609   0.00%   10
2547-202336     7.14%   14
201943-202339   8.33%   12
201941-1495     8.33%   12
201976-101145   8.33%   12
2736-2546       8.33%   12
201566-203458   9.09%   11
203484-201167   9.09%   11
201609-203463   10.00%  10
```

Comfortable Zones of Shooting

For each player, we define the comfortable zone of shooting as a matrix of, {SHOT DIST, CLOSE DEF DIST, SHOT CLOCK}. Please develop a MapReduce-based algorithm to classify each player's records into 4 comfortable zones. Considering the hit rate, which zone is the best for James Harden, Chris Paul, Stephen Curry and Lebron James.

The general idea we put into practice to answer this question was to implement a K-Means algorithm with an iterative approach.

Our first step was to compute each average per player based on their shot distance, closest defender and remaining shot clock time. After that MapReduce job we were able to isolate by lines a set of stats per player. Four random centroids were generated initially and after a 10 step iteration process of updating those values based on euclidean distance from each cluster we were able to assign a space for each player.

As part of the iterative process we also did convergence checks to ensure whether it was needed to continue iterating or not before outputting the resulting clusters.

Curry, Jarden and Paul seem to land on the same cluster and Lebron is by himself on cluster 1, this makes sense because the initial three play the point guard (and shooting guard occasionally) position and it is somewhat safe to say that their play styles are similar, for instance, both Curry and Harden are shooters, because of that we were not surprised to find them in the same cluster. When it comes to Lebron, his playstyle and position are different, as a PF that plays closer to the board once again it makes sense that he was placed in a bucket with a shorter shot distance.

```
------------------------ resulting clusters ------------
1       12.5004,4.1883,12.3434
3       15.9101,4.5112,12.3667
2       19.0449,5.0171,12.2235
4       6.9236,3.1153,12.809
```

```
Curry-Stephen    3-49.31%
Harden-James     3-44.83%
James-LeBron     1-49.42%
Paul-Chris       3-48.41%
```

BONUS QUESTION

Question 1: Given a Black vehicle parking illegally at 34510, 10030, 34050 (street codes). What is the probability that it will get a ticket? (very rough prediction)

The results come from getting the car color and street codes variables. On the mapper stage, we filter the black cars observated and the violations in the street code. Afterwards, we distinguish by counting the cars of color black and the ones that weren't within the street codes. Key equals to Yes if it's black and No if is non-black, value equals to count. Finally, the reducer gets the probability of a Black color vehicle parking by having the total number of black cars parked divided by the total number of cars within the street codes.

Note: The street codes [34510, 10030, 34050] don't appear on our testing dataset. So, the run was made by adding the street codes['25390','22040'].

```
--------------------------- Question: What is the probability that it will get an ticket?? ---
--------------------
('Probability of a Black color vehicle parking is:', 0.375)
```