CISC5950 – Big Data Programming – Spring 2022

Professor: Ying Mao
Lab 2
Due date: 4/27/2022
Student: Yuliya Akchurina

Lab 2. Spark. Question 1

In this lab, please, based on your previous code, implement the K-Means algorithm, you can use any spark related library package.

1. Please redo Project 1 Part 2 Question 2 (Part 1).

Question1.

The goal of this task is to use Spark on Hadoop HDFS in order to calculate k-means on the provided data and after finding the four clusters and the cluster centers to compute the hit rate for each of the four players given the shot outcome for each data point. The four players are James Harden, Chris Paul, Stephen Curry, Lebron James.

The data is provided in a csv file: shot_logs.csv.

Data is located at /mapreduce-test/mapreduce-test-python/project1/shot_logs.csv

Only selected columns are used from this dataset: player_name, SHOT_DIST, CLOSE_DEF_DIST, SHOT_CLOCK, SHOT_RESULT.

The dataset can be loaded two ways:

1. **Directly, without the use of test.sh script.**
   For that in the lab2_q1.py file uncomment the line 29.

```
#df=spark.read.csv("/mapreduce-test/mapreduce-test-python/project1/shot_logs.csv", header=True, inferSchema=True).withColumn("id", monotonically_increasing_id())
```

   And comment out the lines 32, 35, 37, 38 in the lab2_q1.py file

```
31      # Working wuth data input with test.sh
32      df = spark.read.format("csv").option("header", "true").load(sys.argv[1])
33
34      # add row number column
35      df= df.withColumn("new_column",lit("ABC"))
36
37      w = Window().partitionBy('new_column').orderBy(lit('A'))
38      df = df.withColumn("id", row_number().over(w)).drop("new_column")
```

If data is loaded directly then to run the lab2_q1.py file navigate to the /spark-examples/Lab2/l2q1 folder and type python3 lab2_q1.py

The outcome of running lab2_q1.py this way is below. Max hit rate and the cluster id it belongs to.

```
Here are the most comfortable zone for each player
        based of the highest hit rate
+-------------+------------------+----------+
|  player_name|      max_hit_rate|cluster_id|
+-------------+------------------+----------+
|   chris paul|0.5408805031446541|         3|
| james harden| 0.559322033898305|         1|
| lebron james|0.6481481481481481|         1|
|stephen curry|0.6190476190476191|         1|
+-------------+------------------+----------+

Here are the most comfortable zone for each player
        based of the highest hit rate.
        With max hit rate rounded to two decimal points.
+-------------+------------+----------+
|  player_name|max_hit_rate|cluster_id|
+-------------+------------+----------+
|   chris paul|        0.54|         3|
| james harden|        0.56|         1|
| lebron james|        0.65|         1|
|stephen curry|        0.62|         1|
+-------------+------------+----------+
```

Also we can see max hit rate calculated for each cluster for each player under “avg(SHOT_RESULT)” column.

```
+-------------+----------+-------------------+
|  player_name|cluster_id|   avg(SHOT_RESULT)|
+-------------+----------+-------------------+
|   chris paul|         0|0.43388429752066116|
|   chris paul|         1|               0.48|
|   chris paul|         2|0.49538461538461537|
|   chris paul|         3| 0.5408805031446541|
| james harden|         0| 0.4249084249084249|
| james harden|         1|  0.559322033898305|
| james harden|         2| 0.3333333333333333|
| james harden|         3|0.47619047619047616|
| lebron james|         0|0.36395759717314485|
| lebron james|         1| 0.6481481481481481|
| lebron james|         2|0.40963855421686746|
| lebron james|         3| 0.5350877192982456|
|stephen curry|         0|0.39215686274509803|
|stephen curry|         1| 0.6190476190476191|
|stephen curry|         2|0.43680709534368073|
|stephen curry|         3| 0.6037735849056604|
+-------------+----------+-------------------+
```

2. Running with test.sh file.

To run the lab2_q1.py with test.sh run file, use the lab2_q1.py as is. To run the file navigate to /spark-examples/Lab2/l2q1 folder and type bash test.sh. Test.sh file will feed the input data to the lab2_q1.py .

Here are the Silhouette score and four cluster centers for the NBA dataset:

```
Silhouette with squared euclidean distance = 0.5590500085378539
Cluster Centers:
[22.34126316  4.67073684  6.48284211]
[ 5.2126898   3.34143167 18.02895879]
[22.5390264   5.32879538 16.75181518]
[7.65839637 2.97579425 8.596823  ]
```

The result of running the python file through test.sh file.

```
2022-04-27 12:54:39,771 INFO scheduler.DAGS
ssorImpl.java:0, took 0.228760 s
+-------------+------------+----------+
|  player_name|max_hit_rate|cluster_id|
+-------------+------------+----------+
|   chris paul|        0.54|         2|
| james harden|        0.56|         1|
| lebron james|        0.65|         1|
|stephen curry|        0.62|         1|
+-------------+------------+----------+
```

The cluster id is different from the previous snapshot because every tome the code runs cluster id changes but the max hit rates are the same.

There was an issue with the test.sh file. The Node was running is safe mode. It was resolved by adding this line to test.sh file

hadoop dfsadmin -safemode leave

```
10.142.0.4: starting org.apache.spark.deploy.worker.Worker, logging to /usr/lo
oot-org.apache.spark.deploy.worker.Worker-1-instance-3.out
rm: `/lab2_q1/input/': No such file or directory
mkdir: Cannot create directory /lab2_q1/input. Name node is in safe mode.
```

Data Processing and functionality of lab2_q1.py script.

The PySpark dataframe is created from columns player_name, SHOT_DIST, CLOSE_DEF_DIST, SHOT_CLOCK, SHOT_RESULT. Additional column of “id” is created which is a row number of columns. It is needed to be able to match the datapoints in the data frame after it has been modified.

The columns SHOT_DIST, CLOSE_DEF_DIST, SHOT_CLOCK are converted from string to double format. The column SHOT_RESULT is encoded to 1 if “made” and “0” if “missed” to be later used in calculation.

The feature columns SHOT_DIST, CLOSE_DEF_DIST, SHOT_CLOCK are converted into sparce vector to be used in the PySpark Kmeans model.

The following model from PySpark library is used to cluster the data:

```
from pyspark.ml.clustering import KMeans
from pyspark.ml.evaluation import ClusteringEvaluator
```

Once cluster centers are known the data points are assigned to each cluster and cluster assignment is stored in “cluster_id” column. The “cluster_id” column is added to the matching rows of the original datfarame.

To calculate the hit rate, I used the groupBy function and aggregation functions average and max on the dataframe columns. The dataframe is sorted by player name.

The two dataframes are created df_avg that has hit rate per each player for each of the 4 clusters, a total of 16 rows. And df_max that has highest hit rates among the 4 clusters for each of the 4 players, a total of four rows.

The dataframes df_max and df_avg are joined into df_result. The hit rate is rounded to two decimal places.

The printed-out result are the columns "player_name", "max_hit_rate", "cluster_id" of df_result dataframe.

```
2022-04-27 12:54:39,771 INFO scheduler.DAGS
ssorImpl.java:0, took 0.228760 s
+-------------+------------+----------+
|  player_name|max_hit_rate|cluster_id|
+-------------+------------+----------+
|   chris paul|        0.54|         2|
| james harden|        0.56|         1|
| lebron james|        0.65|         1|
|stephen curry|        0.62|         1|
+-------------+------------+----------+
```