Universität Hamburg
Department Informatik
Wissenschaftliches Rechnen
Dr. J. Kunkel, H. Hatzel

Exercise Sheet 7
Big Data Analytics
WiSe 2016/2017
Due Date: 2016-12-09, 420 P

Please document your code, without sufficent documentation you won't recive any points.

1 Exploration of IMDb Quotes (R) (60 P)

In this task, we explore our previously cleaned IMDb quotes available in `/home/bigdata/7/imdb-quotes.csv`.
Use the methods of descriptive statistics to analyze the overall data.
Create a lab notebook PDF containing your results and their analysis.

1.1 Guiding questions

- Are there movies which quotes are extreme? (Length of text, number of quotes, number of lines, involved actors)
- Is there a correlation between "interesting words" and movie, i.e., can we deduct the movie using some characteristics (features) extracted from all quotes?

1.2 Hints

Code for reading the CSV file in R is:

```
d = read.csv("imdb-quotes.csv", header=F, stringsAsFactors=F, sep="|", quote="\"")
colnames(d) = c("movie", "year", "episode", "actors", "quote")

# data cleaning
d$year = as.numeric(d$year)
d = d[d$year > 1800 & d$year < 2020 & ! is.na(d$year),]
```

Submission:

`1-analysis.pdf` A lab notebook with your code, analysis and results.

2 Classification of Titanic Survials (Python or R) (120 P)

The dataset `/home/bigdata/7/titanic.csv` contains information on the passengers of the titanic. The following facts are recorded for each passenger: Class, Gender, Age. Class indication which category of ticket they had bought. The last fact the dataset contains is whether or not the passenger survived.
Approach the task as follows:

- Create a decision tree for all passengers and try to deduct useful rules for determining the survival or demise of a passenger.
- Execute k-fold-cross-validation to evaluate the accuracy of your decision tree[1]
- Compute the error rate of the predictions for $k = (2, .., 10)$ and visualize false negative and false positive rates in a diagram.
- Evaluate the quality of your predictions. How much training data is needed to yield some acceptable results?

You may choose to use either R or Python for this task.

[1] Implement the splitting of datasets into training set and validation set yourself.

Submission:

`2-titanic-tree.(R|py)`	Your script for analyzing the dataset.
`2-titanic-tree.pdf`	Your evaluation of the decision tree and it's performance when using k-cross-validation.

3 Computing of Word Frequencies using Hive (180 P)

For this task we use HiveQL to get word frequencies in all Wikipedia articles. Our goal is to create a map of word frequencies per article, resulting in the following output:

```
12 {"by":60,"for":48,...} # for article 12
13 ... # for article 13
```

Document your programs runtime. Export the results as a CSV to HDFS or the local file system.

3.1 Hints

If you create additional tables, please prefix your name to the table to avoid name conflicts with other groups.
The file `/home/bigdata/7/wiki-clean.csv` is already imported in HDFS[2]. A compressed SequenceFile (GZIP compression) is also available with the suffix "seq.gz" When creating the schema take special care to NOT move the file (take a look a the SQL-keywords EXTERNAL and LOCATION).
When creating the schema it's advisable to split the rows using a regular expression.

```
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.RegexSerDe' with SERDEPROPERTIES ("input.regex" =
    ↪ "^YOUR REGEX HERE$")
```

Use `SELECT * FROM TABLE wiki LIMIT 5;` to check whether or not your schema was created correctly.
If you do any preprocessing on the CSV-File please delete your custom file after finishing the task.
Some of the functions document at `https://cwiki.apache.org/confluence/display/Hive/LanguageManual+UDF#LanguageManualUDF-StringFunctions` are useful for this exercise.
Functions from `https://github.com/klout/brickhouse` might also prove useful. To enable them please enter the following commands in your Hive Shell:

```
ADD JAR /home/bigdata/brickhouse-0.7.1-SNAPSHOT.jar;
SOURCE /home/bigdata/brickhouse.hql;
```

Submission:

`3-wiki-word-count.hql`	Your Hive commands creatin of a table as well as calcuation of word frequencies and the export to CSV.

4 Wort Count in Hive using External Scripts (Python) (60 P)

This task has the same goal as *Task 2*. Now however you will approach the problem using a TRANSFORM clause instead. You can make use of your previous Python scripts. Document your programs runtime and compare it with the runtime from *Task 2*. Export the result to either HDFS or local filesystem.

4.1 Hinweise

Your SQL queries should be simpler then those in *Task 2*

[2] This file includes the header.

Submission:

`4-wiki-word-count.py`	Your Python program.
`4-wiki-word-count.hql`	Your Hive commands creation of a table as well as calculation of word frequencies and the export to CSV. Runtime comparison with the pure HIVE solution.