Aims

This exercise aims to get you to:

- Submitting python script to Spark
- practice more on Spark Programming using DataFrame with Python

Spark DataFrame Programming

Question 1. Download the input text file pg100.txt from WebCMS3. Write a Spark program which outputs the number of words that start with each letter. This means that for every letter we want to count the total number of words that start with that letter.

- Ignore the letter case, i.e., consider all words as lower case.
- Ignore terms starting with non-alphabetical characters, i.e., only consider terms starting with "a" to "z".
- Use the space character to split the documents into words:

The output should be formatted as "key\tvalue" pairs.

Name your Python file as "problem1_df.py". Put the input file in the HDFS folder "/user/comp9313/input", and store your output in the HDFS folder "user/comp9313/output". The input and output paths are obtained from the arguments. The result can be found at (the same as that in Lab 6):
https://webcms3.cse.unsw.edu.au/COMP9313/23T3/resources/92011

You can run your job by the following command (HDFS paths can also be used):

```
$ spark-submit problem1_df.py file:///home/comp9313/pg100.txt
file:///home/comp9313/output
```

Question 2. Download the input text file pg100.txt from WebCMS3. Compute the average length of words starting with each letter. This means that for every letter, you need to compute: the total length of all words that start with that letter divided by the total number of words that start with that letter.

- Ignore the letter case, i.e., consider all words as lower case.
- Ignore terms starting with non-alphabetical characters, i.e., only consider terms starting with "a" to "z".
- The length of a term X can be obtained by len(X).
- Use the following split function to split the documents into terms:

```
import re
re.split("[\\s*$&#/\"\\,.:;?!\\[\\](){}<>~\\-_]+", line)
```

Your Spark program should generate a list of key-value pairs. Keys and values are separated by ",", and the values are of double precision, ranked in alphabetical order. You can see the result at:
https://webcms3.cse.unsw.edu.au/COMP9313/23T3/resources/92012

Name your python file as “Problem2_df.py”. Put the input file “pg100.txt” in HDFS folder “/user/comp9313/input”, and store your output in HDFS folder “user/comp9313/output”. The input and output paths are obtained from the arguments.

You can run your job by the following command (HDFS paths can also be used):

```
$ spark-submit problem2_df.py file:///home/comp9313/pg100.txt
file:///home/comp9313/output
```

Question 3. Download the sample input file “Votes.csv” from: https://webcms3.cse.unsw.edu.au/COMP9313/23T3/resources/92008, and put it in HDFS folder “/user/comp9313/input”. In this file, the fields are separated by ‘,’ and the lines are separated by ‘\n’. The data format of “Votes.csv” is as below:

```
- Id
- PostId
- VoteTypeId
   - ` 1`: AcceptedByOriginator
   - ` 2`: UpMod
   - ` 3`: DownMod
   - ` 4`: Offensive
   - ` 5`: Favorite - if VoteTypeId = 5 UserId will be populated
   - ` 6`: Close
   - ` 7`: Reopen
   - ` 8`: BountyStart
   - ` 9`: BountyClose
   - `10`: Deletion
   - `11`: Undeletion
   - `12`: Spam
   - `13`: InformModerator
   - `14`:
   - `15`:
   - `16`:
- UserId (only for VoteTypeId 5)
- CreationDate
```

(i). Find the top-5 VoteTypeIds that have the most distinct posts. You need to output the VoteTypeId and the number of posts. The results are ranked in descending order according to the number of posts, and each line is in format of: VoteTypeId\tNumber of posts.

(ii). Find all posts that are favoured by more than 10 users. You need to output both PostId and the list of UserIds, and each line is in format of:

PostId#UserId1,UserId2,UserId3,…,UserIdn

The lines are sorted according to the NUMERIC values of the PostIds in ascending order. Within each line, the UserIds are sorted according to their NUMERIC values in ascending order.

You can run your job by the following command (HDFS paths can also be used):

```
$ spark-submit problem3_df.py file:///home/comp9313/Votes.csv
```

You can see the result at:
https://webcms3.cse.unsw.edu.au/COMP9313/23T3/resources/92017