Основы работы с Apache Spark

Изучите теоретическую часть.

```
# Устанавливаем OpenJDK
!apt-get install openjdk-8-jdk-headless -qq > /dev/null
# Закачиваем Spark
!wget -q http://archive.apache.org/dist/spark/spark-3.2.0/spark-3.2.0-bin-hadoop2.7.tgz -O spark.tgz
# Распаковываем архив со Spark
!tar xf spark.tgz
# Устанавливаем пакет findspark для работы со Spark из Python
!pip install -q findspark
# Настраиваем переменные окружения для работы с Apache Spark
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-3.2.0-bin-hadoop2.7"
# Находим установку Spark
import findspark
findspark.init()
# Подключаем необходимые модули для работы со Spark из Python
from pyspark.sql import SparkSession
# Создаем сессию Spark на локальном компьютере
```

```
# Создаем сессию Spark на локальном компьютере
spark = SparkSession.builder.master("local[*]").getOrCreate()
!wget http://files.grouplens.org/datasets/movielens/ml-100k.zip -O /content/ml-100k.zip -q
!unzip -qq /content/ml-100k.zip -d "sample_data/"
```

```
replace sample_data/ml-100k/allbut.pl? [y]es, [n]o, [A]ll, [N]one, [r]ename: yes
replace sample_data/ml-100k/mku.sh? [y]es, [n]o, [A]ll, [N]one, [r]ename: All
```

Задача 1

Модернизируйте заготовку заменив все участки `<put your code here>` на ваш код для того, что бы:

- вычислять и выводить на экран статистику по числу оценок для каждого фильма
- вычислять и выводить на экран статистику по числу оценок для всех фильмов

Статистика для *каждого* фильма:

```
Marks for film 346: 1 -> 7, 2 -> 10, 3 -> 32, 4 -> 49, 5 -> 28
Marks for film 474: 1 -> 0, 2 -> 6, 3 -> 34, 4 -> 59, 5 -> 95
Marks for film 265: 1 -> 1, 2 -> 13, 3 -> 62, 4 -> 91, 5 -> 60
Marks for film 465: 1 -> 4, 2 -> 8, 3 -> 26, 4 -> 30, 5 -> 17
Marks for film 451: 1 -> 15, 2 -> 31, 3 -> 37, 4 -> 54, 5 -> 33
Marks for film 86: 1 -> 4, 2 -> 10, 3 -> 23, 4 -> 67, 5 -> 46
Marks for film 257: 1 -> 2, 2 -> 28, 3 -> 81, 4 -> 126, 5 -> 66
Marks for film 222: 1 -> 7, 2 -> 30, 3 -> 108, 4 -> 155, 5 -> 65
Marks for film 40: 1 -> 9, 2 -> 9, 3 -> 20, 4 -> 17, 5 -> 2
Marks for film 29: 1 -> 15, 2 -> 34, 3 -> 45, 4 -> 14, 5 -> 6
```

Для *всех* фильмов:

```
Marks for films ALL: 1 -> 6110, 2 -> 34174, 3 -> 27145, 4 -> 11370, 5 -> 21201
```

```
import collections
rdd = spark.sparkContext.textFile("/content/sample_data/ml-100k/u.data")
#<put your code here>

def printStat(inp):
  #<put your code here>
  print(f'Marks for film {ind}: 1 -> {marks[0]}, 2 -> {marks[1]}, 3 -> {marks[2]}, 4 -> {marks[3]}, 5 -> {marks[4]}')

for i in aggPairRDD.mapValues(lambda x: dict(collections.Counter(x))).collect():
  printStat(i)

#<put your code here>
```

Задача 2

Произведите подсчёт частоты встречаемости слов с использованием ApacheSpark RDD. Ячейка ниже скачивает текст. Вам требуется:

- Очистить текст от знаков препинания и пустых строк
- Перевести в нижний регистр и разделить по пробелам
- Подсчитать наиболее часто встречающиеся символы
- Использовать RDD

Пример вывода:

```
[('и', 2204),
 ('в', 1977),
 ('я', 1252),
 ('не', 1247),
 ('на', 1094),
 ('он', 755),
 ('как', 717),
 ('с', 693),
```

```
 ('что', 653),
 ('его', 502)]
```

```
!wget http://www.lib.ru/INOOLD/BALZAK/shagren.txt_Ascii.txt | iconv -f cp1251
i = 0
with open('/content/shagren.txt_Ascii.txt', encoding="cp1251") as inF, open('/content/shagren.txt_utf8.txt', "w") as outF:
  for line in inF:
    outF.write(line)
```

```
--2021-11-22 10:25:48--  http://www.lib.ru/INOOLD/BALZAK/shagren.txt_Ascii.txt
Resolving www.lib.ru (www.lib.ru)... 81.176.66.163
Connecting to www.lib.ru (www.lib.ru)|81.176.66.163|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/plain]
Saving to: ‘shagren.txt_Ascii.txt.2’

shagren.txt_Ascii.t     [    <=>             ] 510.68K   570KB/s    in 0.9s

2021-11-22 10:25:51 (570 KB/s) - ‘shagren.txt_Ascii.txt.2’ saved [522937]
```

Colab paid products - Cancel contracts here