Лабораторная работа № 19

Тема: Основы работы с Apache Spark. Обработка текстовых данных.

Цель: Ознакомиться с основными принципами работы с Apache Spark на языке Python. Научиться создавать RDD и DataFrames. Выполнить базовую обработку данных: фильтрацию и преобразование с использованием PySpark.

Задание:

1. Возьмите текстовый файл, например отсюда: https://flibusta.su/

Конвертируйте его в UTF-8 и загрузите в HDFS.

2. В Jupyter notebook (port 8888) установите pyspark.

```
In [1]: pip install pyspark
```

```
Requirement already satisfied: pyspark in /usr/local/spark-2.4.5-bin-hadoop2.7/python (2.4.5)
Collecting py4j==0.10.7
  Downloading py4j-0.10.7-py2.py3-none-any.whl (197 kB)
     |████████████████████████████████| 197 kB 629 kB/s eta 0:00:01
Installing collected packages: py4j
Successfully installed py4j-0.10.7
Note: you may need to restart the kernel to use updated packages.
```

2. Импортируйте необходимые библиотеки, создайте сессию и проверьте:

```
In [2]: from pyspark.sql import SparkSession
```

```
In [3]: # Создание SparkSession
        spark = SparkSession.builder \
            .master("spark://spark-master:7077") \
            .appName("WordCount") \
            .getOrCreate()
```

```
In [4]: # Проверка сессии
        print(spark)
```

```
<pyspark.sql.session.SparkSession object at 0x7f6198563750>
```

3. Прочитайте загруженный файл и посчитайте общее количество слов в файле, количество уникальных слов, наиболее часто встречающиеся слова.
Пример:

```
In [40]: # Чтение файла из HDFS
         df = spark.read.text("hdfs://namenode:8020/pilevin.txt")

         # Разделение текста на слова и подсчет
         words = df.select(explode(split(regexp_replace(col("value"), "[.,!?;«»()*-]", ""), "\\s+")).alias("word"))

         # Подсчет количества вхождений каждого слова
         word_counts = words.groupBy("word").count()

         # Сортировка по количеству вхождений и выбор топ-N слов
         top_n = word_counts.orderBy(col("count").desc()).limit(20)
```

In [41]:
```
# Показ результата
top_n.show()
```

```
+----+-----+
|word|count|
+----+-----+
|    |  923|
|   и|  709|
|   в|  601|
|  не|  443|
|  на|  365|
| что|  275|
|   с|  237|
|было|  181|
|Маня|  179|
| как|  173|
| это|  148|
|  Но|  131|
|   а|  123|
|  из|  111|
| так|  109|
| все|  107|
|  ее|  100|
| она|   99|
|   у|   98|
|   к|   92|
+----+-----+
```

In [42]:
```
# Подсчет общего количества слов
total_word_count = words.count()

# Вывод общего количества слов
print(f"Общее количество слов: {total_word_count}")
```

```
Общее количество слов: 24473
```

```
# Получение и сортировка уникальных слов
unique_words = words.select("word").distinct().orderBy("word")

unique_words.show()
```

```
+------------+
|        word|
+------------+
|            |
|           1|
|         187|
|           2|
|        2021|
|         206|
|     206 год|
|           3|
|3Dтехнологию|
|           A|
|      ACSF -|
|          AI|
|       ALIVE|
|         AUX|
|          AV|
|         All|
| Allocations|
|          BE|
|         BOX|
|      Bought|
+------------+
only showing top 20 rows
```

4. Сохраните результаты в файл.

```
In [46]: # Сохранение списка уникальных слов в файл
         #unique_words.write.mode("overwrite").text("hdfs://namenode:8020/unique_words_sorted.txt")
         unique_words.coalesce(1).write.mode("overwrite").text("hdfs://namenode:8020/file1.txt")
```

Извлеките файл из контейнера:

Подключитесь к контейнеру:

```
user@hadoop:~/docker-hadoop-spark-hive2-jupyter$ sudo docker exec -it namenode /bin/bash
```

Скопируйте файл из HDFS на локальную файловую систему контейнера:

```
root@8043d6227575:/# hdfs dfs -get /file1.txt /tmp/file1.txt
```

Выйдите из контейнера и скопируйте файл с контейнера на хост:

```
root@8043d6227575:/# exit
exit
user@hadoop:~/docker-hadoop-spark-hive2-jupyter$ sudo docker cp namenode:/tmp/file1.txt .
Successfully copied 328kB to /home/user/docker-hadoop-spark-hive2-jupyter/.
```

Появилась папка file1.txt, в ней должен быть текстовый файл:

```
user@hadoop:~/docker-hadoop-spark-hive2-jupyter$ ls -l
total 24
-rw-rw-r-- 1 user user 3688 окт 29 11:23 docker-compose.yml
-rw-rw-r-- 1 user user 2678 окт 29 11:22 example.ipynb
drwxr-xr-x 3 root root 4096 окт 30 05:12 file1.txt
-rw-rw-r-- 1 user user 1663 окт 29 11:22 hadoop-hive.env
-rw-rw-r-- 1 user user 1042 окт 29 11:22 README.md
drwxrwxr-x 4 user user 4096 окт 29 11:22 spark_conf
user@hadoop:~/docker-hadoop-spark-hive2-jupyter$ cd file1.txt/
user@hadoop:~/docker-hadoop-spark-hive2-jupyter/file1.txt$ ls
file1.txt  part-00000-531ed177-86c3-440a-9788-27df86d9459e-c000.txt  _SUCCESS
```

Прочитаем первые строки:

```
user@hadoop:~/docker-hadoop-spark-hive2-jupyter/file1.txt$ head part-00000-531ed177-86c3-440a-9788-27df86d9459
e-c000.txt

1
187
2
2021
206
206 год
3
3Dтехнологию
A
user@hadoop:~/docker-hadoop-spark-hive2-jupyter/file1.txt$
```

Отчет должен содержать (см. образец):

- номер и тему лабораторной работы;
- фамилию, номер группы студента и вариант задания;
- скриншоты подтверждающие выполнение заданий.

Отчеты в формате **pdf** отправлять на email: **colledge20education23@gmail.com**