Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «Информатика и управление»

КАФЕДРА ИУК4 «Программное обеспечение ЭВМ, информационные технологии»

Домашняя работа

«Основы Spark. Установка Spark. Основные команды для работы с RDD»

ДИСЦИПЛИНА: «Технологии обработки больших данных»

Выполнил: студент гр. ИУК4-72Б _____ (Сафронов Н.С.)
(подпись) (Ф.И.О.)

Проверил: _____ (Голубева С.Е.)
(подпись) (Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Калуга, 2023

Цель работы: формирование практических навыков работы с платформой Apache Spark для обработки больших данных.

Постановка задачи

Написать скрипт для платформы Apache Spark для решения задачи, указанной в варианте.

Вариант 4

Подсчитать средний рейтинг фильма. Входный файл рейтингов имеет формат:

userId, movieId, rating, timestamp

Выполнить операцию объединения с файлом, содержащим названия фильмов. Данный файл имеет формат:

movieId, title, genres

Результат должен быть сохранен в файле в формате:

movieId, title, av_rating

Объединение выполнять согласно подходу Job-side join. В результате должны быть представлены 20 фильмов с самым высоким средним рейтингом.

Исходные файлы:

rating.csv

movies.csv

Результат выполнения работы

Листинг программы

```
from pyspark.sql import SparkSession, functions

if __name__ == '__main__':
    spark = SparkSession.builder.appName('keyone-
app').getOrCreate()

    movies_path = 'data/movies.csv'
    ratings_path = 'data/ratings.csv'
    result_path = 'data/result'
```

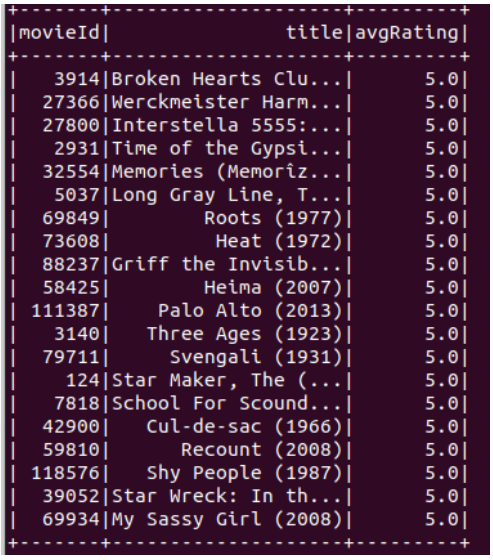
```
    movies = spark.read.csv(movies_path, header=True)
    ratings = spark.read.csv(ratings_path, header=True)

    average_ratings = ratings.groupBy('movieId').agg(
        functions.mean('rating').alias('avgRating')
    ).select('movieId', 'avgRating')

    joined_table = average_ratings.join(movies,
'movieId').select(
        'movieId', 'title', 'avgRating'
    )
    sorted_table = joined_table.orderBy('avgRating',
ascending=False)

    result_table = sorted_table.limit(20)
    result_table.show()
    result_table.write.mode('overwrite').csv(result_path)
```

Результаты выполнения скрипта



movieId	title	avgRating
3914	Broken Hearts Clu...	5.0
27366	Werckmeister Harm...	5.0
27800	Interstella 5555:...	5.0
2931	Time of the Gypsi...	5.0
32554	Memories (Memoriz...	5.0
5037	Long Gray Line, T...	5.0
69849	Roots (1977)	5.0
73608	Heat (1972)	5.0
88237	Griff the Invisib...	5.0
58425	Heima (2007)	5.0
111387	Palo Alto (2013)	5.0
3140	Three Ages (1923)	5.0
79711	Svengali (1931)	5.0
124	Star Maker, The (...	5.0
7818	School For Scound...	5.0
42900	Cul-de-sac (1966)	5.0
59810	Recount (2008)	5.0
118576	Shy People (1987)	5.0
39052	Star Wreck: In th...	5.0
69934	My Sassy Girl (2008)	5.0

Рисунок 1 – Результат выполнения скрипта

Вывод: в ходе выполнения домашней работы были сформированы практические навыки работы с платформой Apache Spark для обработки больших данных.