|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Калужский филиал федерального государственного автономного 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»**

**по дисциплине: «Технологии обработки больших данных»**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил: студент группы ИУК4-72Б | |  |  | Губин Е.В. | |
|  | | (Подпись) |  | (И.О. Фамилия) | |
| Проверил: | |  |  | Голубева С.Е. | |
|  | | (Подпись) |  | (И.О. Фамилия) | |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | | | | |
|  | - Балльная оценка:  - Оценка: | | | |

Калуга, 2025

**Цель:** выполнения домашней работы является формирование практических навыков работы с платформой Apache Spark для обработки больших данных.

**Задачи:**

1. Изучить основы Apache Spark.
2. Научиться устанавливать и конфигурировать Spark.
3. Уметь работать с RDD.
4. Получить навыки написания программ для обработки больших данных.

**Вариант №8**

**Формулировка задания:**

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

Если для фильма указано несколько жанров, оставить только первый. Сгруппировать записи по жанру и подсчитать средний рейтинг жанра. Результат сохранить в файл:

genre, av\_rating Исходные файлы:

rating.csv

[movies.csv](https://github.com/clemsonciti/Introduction-to-Hadoop-data/blob/master/ml-latest-small/movies.csv)

**Ход выполнения работы:**

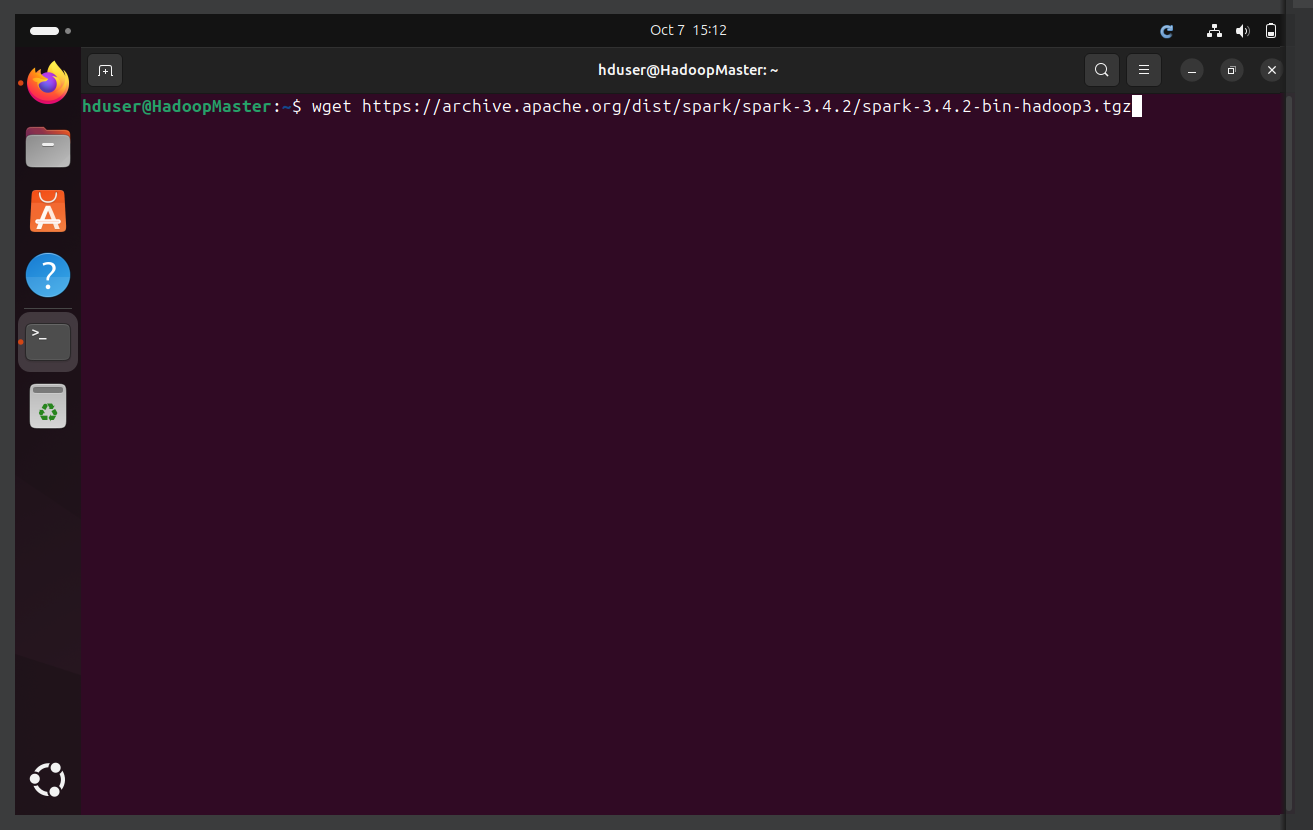


Рисунок Установка Spark

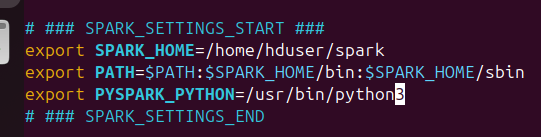


Рисунок Переменные окружения Spark

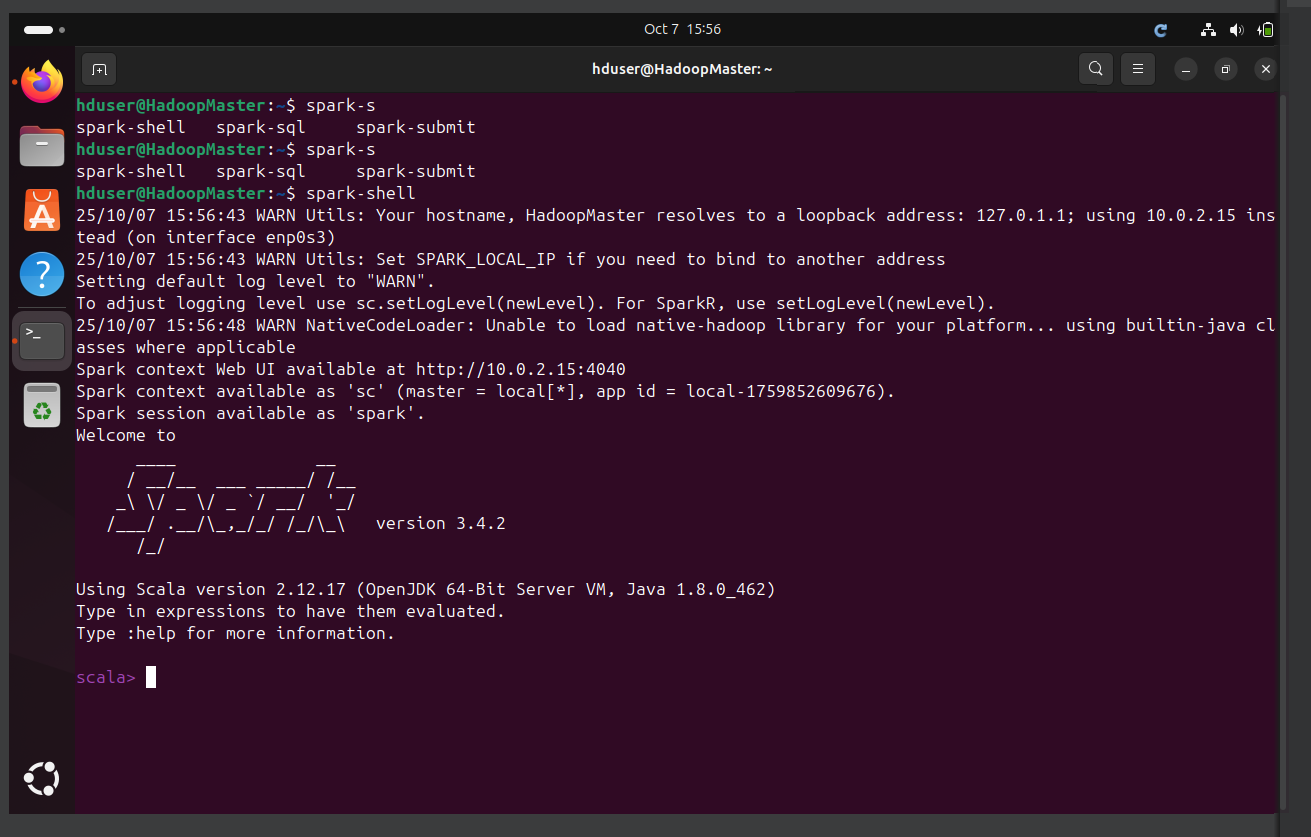


Рисунок Проверка работоспособности Spark

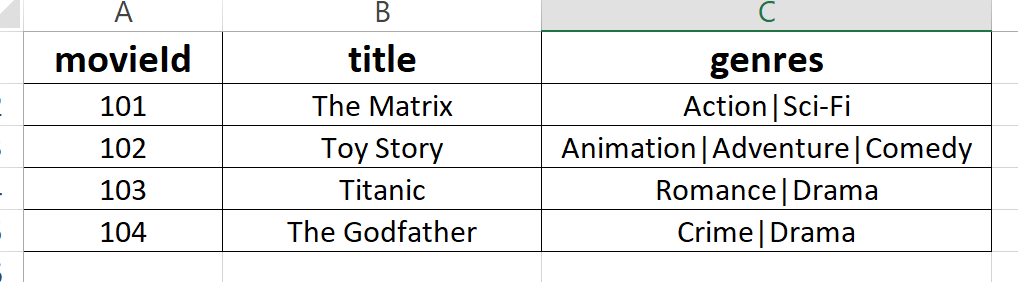


Рисунок movie.csv

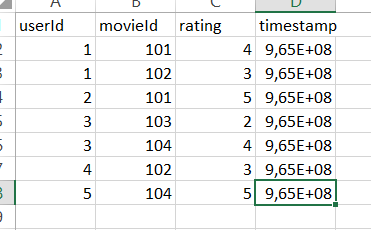


Рисунок rating.csv

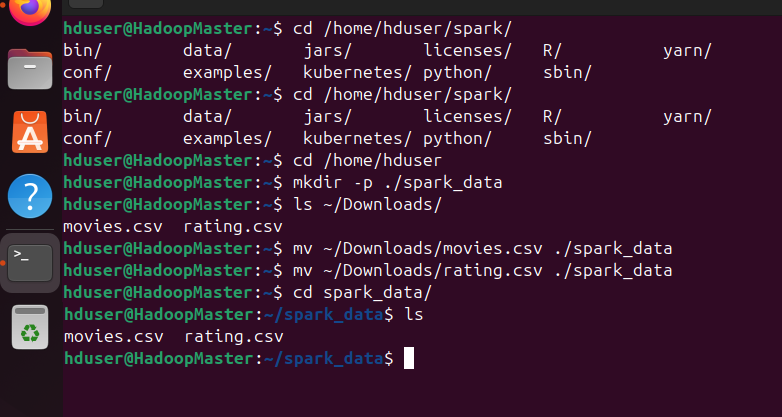


Рисунок Инициализация рабочего пространства для Spark

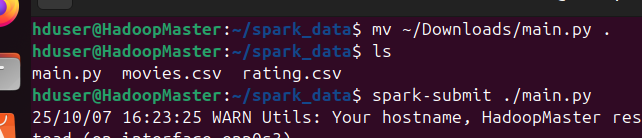


Рисунок Запукск скрипта Python

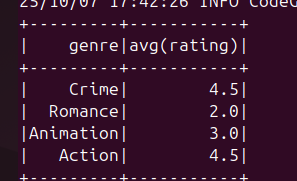


Рисунок Результат работы программы

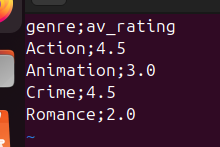


Рисунок Сохранённые результаты

**Листинг программы:**

#!/usr/bin/env python3

from pyspark.sql import SparkSession

from pyspark.sql import functions as F

spark = SparkSession.builder.appName("GenreAverageRating").getOrCreate()

ratings\_path = "file:///home/hduser/spark\_data/rating.csv"

movies\_path  = "file:///home/hduser/spark\_data/movies.csv"

output\_path  = "file:///home/hduser/spark\_data/output"

ratings = spark.read.option("header", True).option("inferSchema", True).option("sep", ";").csv(ratings\_path)

movies  = spark.read.option("header", True).option("inferSchema", True).option("sep", ";").csv(movies\_path)

ratings = ratings.select("movieId", "rating")

movies  = movies.select("movieId", "genres")

movies = movies.withColumn("genre", F.split(F.col("genres"), "\\|").getItem(0)).select("movieId", "genre")

joined = ratings.join(movies, *on*="movieId", *how*="inner")

genre\_avg = joined.groupBy("genre").agg(F.avg("rating").alias("av\_rating\_raw"))

genre\_avg.show()

genre\_avg = genre\_avg.withColumn(

    "av\_rating",

    (F.floor(F.col("av\_rating\_raw") \* 10 + F.lit(0.5)) / F.lit(10))

)

result = genre\_avg.select("genre", "av\_rating").orderBy("genre")

result.show(*truncate*=False)

result.write.mode("overwrite").option("header", True).option("sep", ";").csv(output\_path)

spark.stop()

**Вывод:** в ходе выполнения работы были получены практические навыки по работе со Spark.