|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика, ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ и системы  
 управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

МАГИСТЕРСКАЯ ПРОГРАММА **09.04.01/07 Интеллектуальные системы анализа,**

**обработки и интерпретации больших данных**

**Отчет**

**по лабораторной работе №10**

**Название:** Spark

**Дисциплина:** Языки программирования для работы с большими данными

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-22М |  |  | И.Л. Баришпол |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | П.В. Степанов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2023

**Задания**

1) Выбрать любой датасет на kaggle.com

2) Cделать 10 выборок данных по выбранной предметной области

package lab10

import org.apache.spark.sql.Dataset

import org.apache.spark.sql.Row

import org.apache.spark.sql.SparkSession

fun main(args: Array<String>) {

val spark: SparkSession = SparkSession

.builder()

.appName("Java Spark SQL basic example")

.config("spark.master", "local")

.getOrCreate()

val df: Dataset<Row> =

spark.read().option("header", "true").csv("rotten\_tomatoes\_movies.csv")

df.createOrReplaceTempView("rtm")

// Выбрать все фильмы с рейтингом > 4

spark.sql("SELECT \* FROM rtm WHERE rating = 5 AND tomatoMeter = 100").show()

// Выбрать все фильмы с рейтингом > 4 и рейтингом критиков > 90

spark.sql("SELECT \* FROM rtm WHERE rating = 5 AND tomatoMeter = 100 AND audienceScore = 100").show()

// Выбрать все фильмы с жанром "Comedy"

spark.sql("SELECT \* FROM rtm WHERE genre LIKE '%Comedy%'").show()

// Аггрегировать по жанрам и посчитать количество фильмов в каждом жанре. Сортировать по убыванию

spark.sql("SELECT genre, COUNT(\*) FROM rtm GROUP BY genre ORDER BY COUNT(\*) DESC").show()

// Вывести жанры 10 самых плохих фильмов

spark.sql("SELECT genre FROM rtm ORDER BY tomatoMeter ASC LIMIT 10").show()

// Вывести средний рейтинг фильмов по жанрам

spark.sql("SELECT genre, AVG(tomatoMeter) FROM rtm GROUP BY genre ORDER BY AVG(tomatoMeter) DESC").show()

// Вывести средний рейтинг фильмов по жанрам, у которых рейтинг критиков > 90

spark.sql("SELECT genre, AVG(tomatoMeter) FROM rtm WHERE tomatoMeter > 90 GROUP BY genre ORDER BY AVG(tomatoMeter) DESC")

.show()

// Вывести количество фильмов по годам (использовать из releaseDateTheaters первые 4 символа)

spark.sql("SELECT SUBSTRING(releaseDateTheaters, 1, 4) AS year, COUNT(\*) FROM rtm GROUP BY year ORDER BY year")

.show()

// Вывести года по убыванию среднего рейтинга фильмов

spark.sql("SELECT SUBSTRING(releaseDateTheaters, 1, 4) AS year, AVG(tomatoMeter) FROM rtm GROUP BY year ORDER BY AVG(tomatoMeter) DESC")

.show()

// Вывести количество фильмов по годам и жанрам

spark.sql("SELECT SUBSTRING(releaseDateTheaters, 1, 4) AS year, genre, COUNT(\*) FROM rtm GROUP BY year, genre ORDER BY year, genre")

.show()

}

**Вывод:** в ходе данной лабораторной работы были изучены принципы реализации высокопроизводительных вычислений через среду Spark в Kotlin. Был выбран датасет из открытого источника, после чего загружен в среду Spark как таблица. Далее были выведены 10 различных выборок с условиями и аггрегациями.