|  |  |
| --- | --- |
| 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 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**

**Вариант № 6**

**Название:** Spark

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

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

Москва, 2024

**Цель:** изучить и освоить принципы работы со Spark в Kotlin.

**Задание**: сделать 10 выборок данных по выбранной предметной области.

В качестве предметной области был выбран датасет “UFC fighters statistic”.

Этот набор данных содержит список действующих бойцов UFC и их статистику на момент турнира UFC. Описание полей датасета представлено на рисунке 1. Часть датасета представлена на рисунке 2.

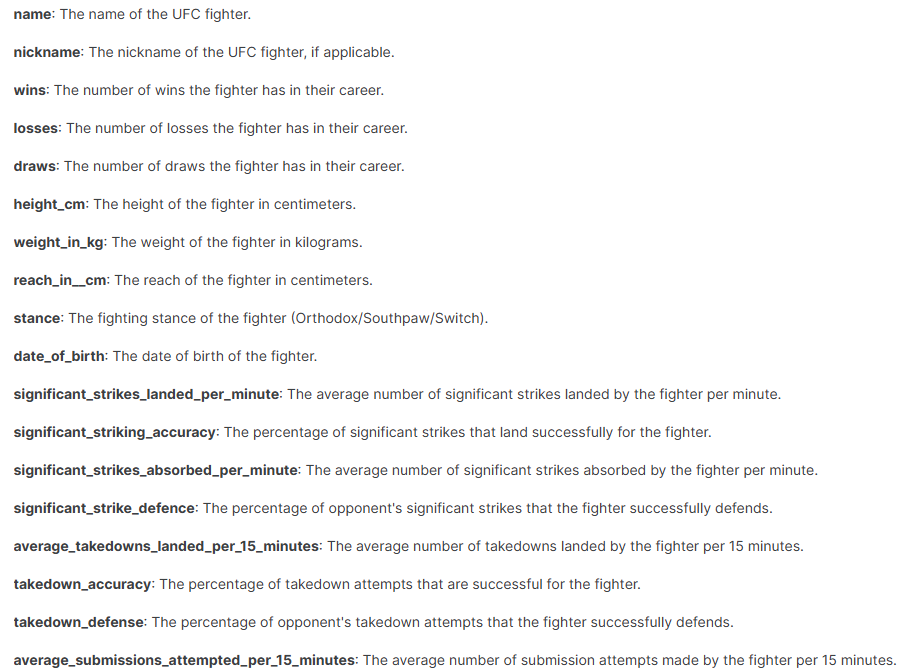


Рисунок 1 – Описание полей датасета

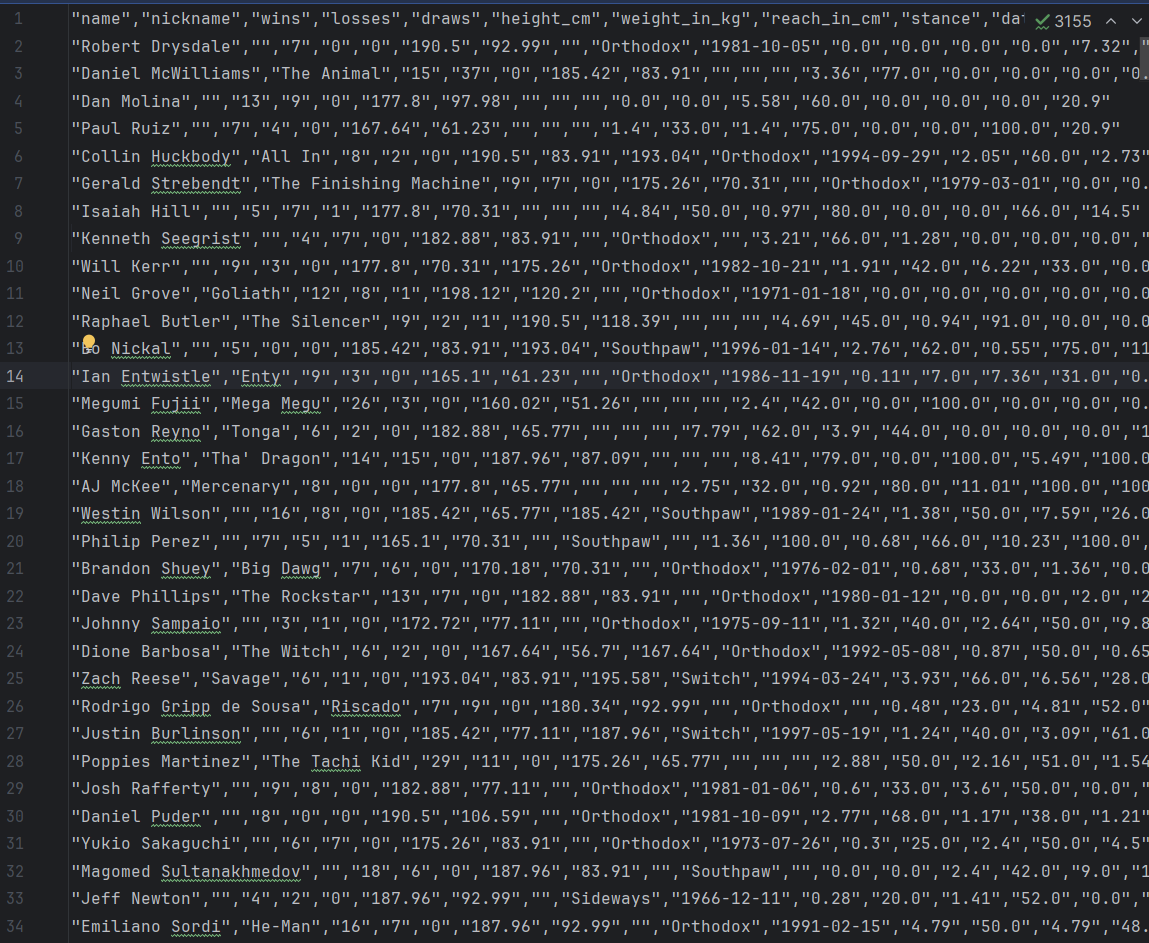


Рисунок 2 – Часть датасета

Код функции Main:

fun main() {

val spark = SparkSession.builder()

.appName("UFC Fighter Statistics")

.master("local[\*]")

.getOrCreate()

spark.sparkContext().setLogLevel("ERROR")

val dataset = spark.read().format("csv")

.option("header", "true")

.option("inferSchema", "true")

.load("ufc-fighters-statistics.csv")

dataset.createOrReplaceTempView("ufc\_stats")

println("Весь датасет:")

spark.sql("SELECT \* FROM ufc\_stats").show()

println("Бойцы с ростом выше 180 см:")

spark.sql("SELECT name, height\_cm FROM ufc\_stats WHERE height\_cm > 180").show()

println("Топ-5 бойцов по количеству побед:")

spark.sql("SELECT name, wins FROM ufc\_stats ORDER BY wins DESC LIMIT 5").show()

println("Бойцы без поражений:")

spark.sql("SELECT name, losses FROM ufc\_stats WHERE losses = 0").show()

println("Топ-5 бойцов по количеству значимых ударов в минуту:")

spark.sql("SELECT name, significant\_strikes\_landed\_per\_minute FROM ufc\_stats ORDER BY significant\_strikes\_landed\_per\_minute DESC LIMIT 5").show()

println("Бойцы со стойкой 'Orthodox':")

spark.sql("SELECT name, stance FROM ufc\_stats WHERE stance = 'Orthodox'").show()

println("Топ-5 бойцов с лучшей защитой от ударов:")

spark.sql("SELECT name, significant\_strike\_defence FROM ufc\_stats ORDER BY significant\_strike\_defence DESC LIMIT 5").show()

println("Топ-5 бойцов по среднему количеству тейкдаунов на 15 минут:")

spark.sql("SELECT name, average\_takedowns\_landed\_per\_15\_minutes FROM ufc\_stats ORDER BY average\_takedowns\_landed\_per\_15\_minutes DESC LIMIT 5").show()

println("Топ-5 бойцов с наибольшей точностью тейкдаунов:")

spark.sql("SELECT name, takedown\_accuracy FROM ufc\_stats ORDER BY takedown\_accuracy DESC LIMIT 5").show()

println("Топ-5 бойцов с лучшей обороной от тейкдаунов:")

spark.sql("SELECT name, takedown\_defense FROM ufc\_stats ORDER BY takedown\_defense DESC LIMIT 5").show()

println("Топ-5 бойцов по количеству попыток сабмишенов на 15 минут:")

spark.sql("SELECT name, average\_submissions\_attempted\_per\_15\_minutes FROM ufc\_stats ORDER BY average\_submissions\_attempted\_per\_15\_minutes DESC LIMIT 5").show()

}

Результат работы программы представлен на рисунке 3.

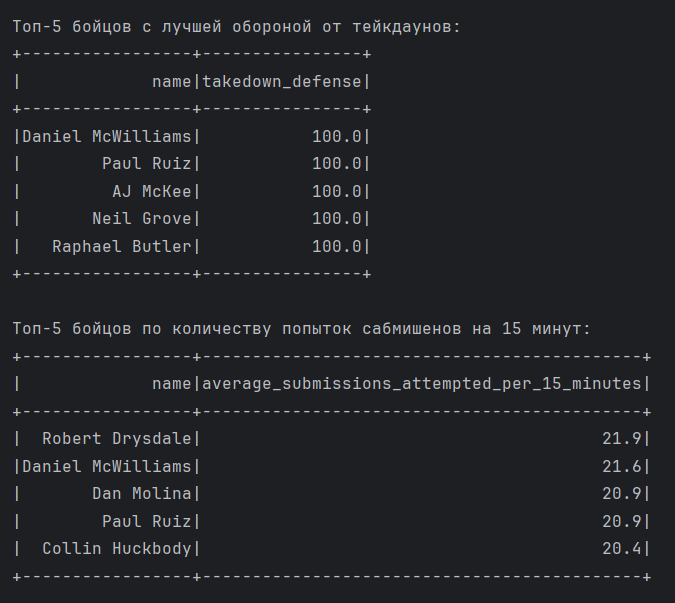


Рисунок 3 – Результат работы программы

**Вывод:** были освоены принципы работы со Spark в Kotlin.