NAME: MARIA GLORIA OBONO ONDO

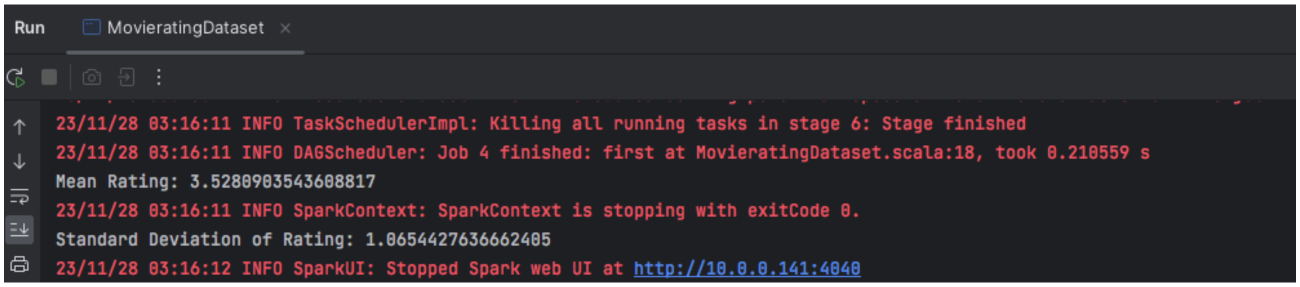
NUID: 002667315

Assignment 7: Analizing Movie Rating

I got the rating.csv file from: https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset?select=ratings.csv

CODE: MovieratingDataset.scala

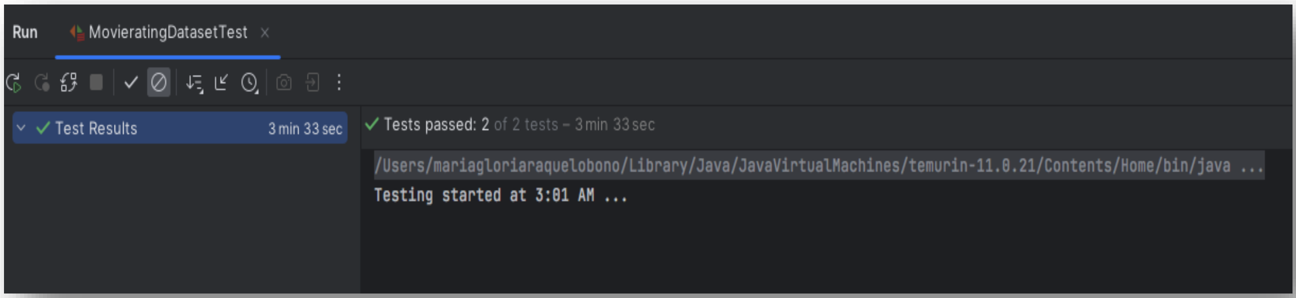
```
import org.apache.spark.sql.SparkSession
import org.apache.spark.sql.functions.{mean, stddev}

object MovieratingDataset {
  def main(args: Array[String]): Unit = {
    // Create a SparkSession
    val spark = SparkSession.builder()
      .appName("MovieratingDataset")
      .master("local[*]") // Change this to your cluster setup
      .getOrCreate()

    // Read the CSV file into a DataFrame
    val filePath = "/Users/mariagloriaraquelobono/Fall2023/Movie-
Analyzer/src/main/resources/ratings.csv"
    val movieData = spark.read.option("header", "true").csv(filePath)

    // Calculate mean rating and standard deviation
    val meanRating = movieData.select(mean("rating")).first().getDouble(0)
    val stdDevRating = movieData.select(stddev("rating")).first().getDouble(0)

    println(s"Mean Rating: $meanRating")
    println(s"Standard Deviation of Rating: $stdDevRating")

    // Stop the SparkSession
    spark.stop()
  }
}
```



```
23/11/28 03:16:11 INFO TaskSchedulerImpl: Killing all running tasks in stage 6: Stage finished
23/11/28 03:16:11 INFO DAGScheduler: Job 4 finished: first at MovieratingDataset.scala:18, took 0.210559 s
Mean Rating: 3.5280903543608817
23/11/28 03:16:11 INFO SparkContext: SparkContext is stopping with exitCode 0.
Standard Deviation of Rating: 1.0654427636662405
23/11/28 03:16:12 INFO SparkUI: Stopped Spark web UI at http://10.0.0.141:4040
```

CODE: MovieratingDatasetTest.scala

```
import org.apache.spark.sql.{SparkSession, DataFrame}
import org.apache.spark.sql.functions.{mean, stddev}
import org.scalatest.funsuite.AnyFunSuite

class MovieratingDatasetTest extends AnyFunSuite {

  def readRatingsCSV(spark: SparkSession, filePath: String): DataFrame = {
    // Read ratings.csv
    spark.read.option("header", "true").csv(filePath)
  }


  test("Test movie ratings analysis with merged data") {
    val spark = SparkSession.builder()
      .appName("MovieratingDatasetTest")
      .master("local[*]")
      .getOrCreate()

    import spark.implicits._

    // Replace these paths with your actual file paths

    val ratingFilePath = "/Users/mariagloriaraquelobono/Fall2023/Movie-
Analyzer/src/main/resources/ratings.csv"

    val ratingsData = readRatingsCSV(spark, ratingFilePath)

    val calculatedStats = ratingsData.agg(mean("rating").as("MeanRating"),
stddev("rating").as("StdDevRating")).head()
    val meanRating = calculatedStats.getAs[Double]("MeanRating")
    val stdDevRating = calculatedStats.getAs[Double]("StdDevRating")

    // Define expected values based on your test data
    val meanRatingExpected = 3.5280903543608817
    val stdDevRatingExpected = 1.0654427636662405
    assert(meanRating === meanRatingExpected)
    assert(stdDevRating === stdDevRatingExpected)

    spark.stop()
  }

  test("Test handling of empty ratings dataset") {
    val spark = SparkSession.builder()
      .appName("MovieratingDatasetTest")
      .master("local[*]")
      .getOrCreate()

    import spark.implicits._

    // Create an empty DataFrame to simulate an empty ratings dataset
    val emptyTestData = Seq.empty[(String, Double)].toDF("userId", "rating")

    val calculatedStats = emptyTestData.agg(mean("rating").as("MeanRating"),
stddev("rating").as("StdDevRating")).head()
    val meanRating = calculatedStats.getAs[Double]("MeanRating")
    val stdDevRating = calculatedStats.getAs[Double]("StdDevRating")

    // Define expected values for an empty dataset
    val meanRatingExpected = 0.0 // Expected mean rating for an empty dataset
    val stdDevRatingExpected = 0.0 // Expected standard deviation for an empty dataset

    assert(meanRating === meanRatingExpected)
    assert(stdDevRating === stdDevRatingExpected)

    spark.stop()
  }

}
```



Tests passed: 2 of 2 tests – 3 min 33 sec

```
/Users/mariagloriaraquelobono/Library/Java/JavaVirtualMachines/temurin-11.0.21/Contents/Home/bin/java ...
Testing started at 3:01 AM ...
```