Mithali Manjunath – SEC01 (NUID 002879571)

Big Data System Engineering with Scala  
Fall 2024

Assignment 7 : Analyzing Movie Rating



**- GitHub Repo URL -**

**- List of Tasks Implemented**

1. read the CSV file using the Spark utility
2. created a method that accepts a DataFrame, calculates mean and standard deviation for all movies and returns the processed DataFrame

**- Code**

The CSV file - ratings.csv from The Movies Dataset in Kaggle is used.

**MovieRatingAnalysis.scala**

package edu.neu.coe.csye7200.csv

import org.apache.spark.sql.{SparkSession, DataFrame}

import org.apache.spark.sql.functions.\_

object MovieRatingAnalysis {

def main(args: Array[String]): Unit = {

val spark = SparkSession.builder()

.appName("Movie Rating Analysis")

.master("local[\*]")

.getOrCreate()

val filePath = "ratings.csv"

val movieDF = spark.read

.option("header", "true")

.option("inferSchema", "true")

.csv(filePath)

val resultDF = calculateStats(movieDF)

resultDF.show()

spark.stop()

}

def calculateStats(df: DataFrame): DataFrame = {

df.groupBy("movieId")

.agg(

avg("rating").alias("mean\_rating"),

stddev("rating").alias("stddev\_rating")

)

}

}

**MovieRatingAnalysisSpec.scala**

package edu.neu.coe.csye7200.csv

import org.scalatest.flatspec.AnyFlatSpec

import org.apache.spark.sql.SparkSession

import org.apache.spark.sql.Row

class MovieRatingAnalysisSpec extends AnyFlatSpec {

val spark = SparkSession.builder()

.appName("TestMovieRatingAnalysis")

.master("local[\*]")

.getOrCreate()

import spark.implicits.\_

"calculateStats" should "return correct mean and standard deviation" in {

val testData = Seq(

(1, 5.0), (1, 3.0), (1, 4.0),

(2, 2.0), (2, 4.0),

(10, 3.0), (10, 3.0), (10, 3.0)

).toDF("movieId", "rating")

val result = MovieRatingAnalysis.calculateStats(testData).collect()

assert(result.contains(Row(1, 4.0, 1.0)))

assert(result.contains(Row(2, 3.0, 1.4142135623730951)))

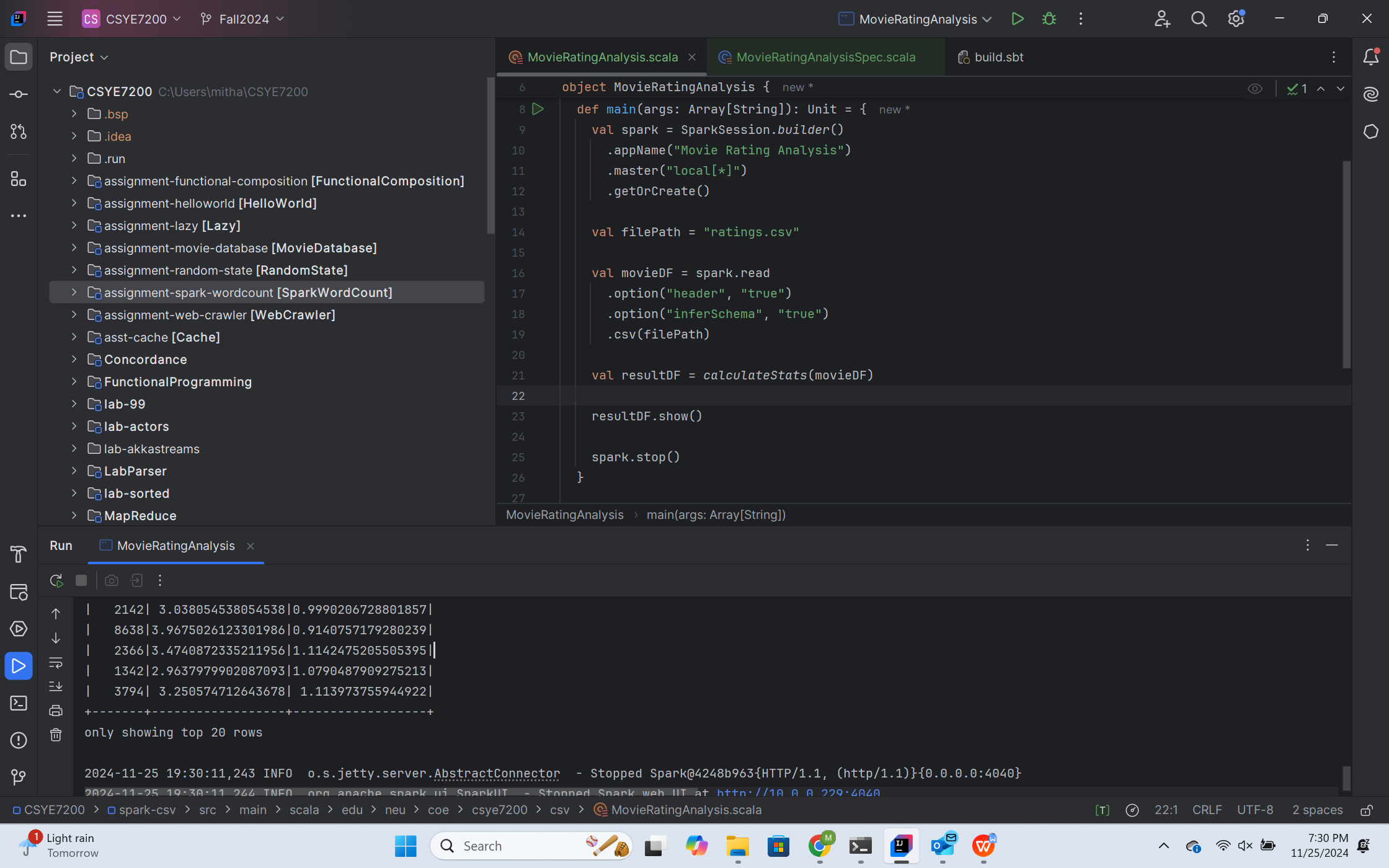
assert(result.contains(Row(10, 3.0, 0.0)))

}

}

**- Unit tests / Results**

Result of running the MovieRatingAnalysis.scala



Result of running the MovieRatingAnalysisSpec.scala

