Project Title	Movie Analytics
Technologies	BigData
Domain	Entertainment
Project Difficulties level	High

Problem Statement:

Develop a big data based project using spark and scala to answer few analytical questions on semi-structured dataset MovieLens dataset containing a million records.

Following task need to be implemented:

1. Use of Spark RDD
2. Spark SQL
3. Spark Data frame

Use spark shell using Scala API.

Perform necessary analysis to draw useful insights about the users and movies by leveraging different form of Spark APIs.

Analytical Queries:

1. What are the top 10 most viewed movies?
2. What are the distinct list of genres available?
3. How many movies for each genre?
4. How many movies are starting with numbers or letters (Example: Starting with 1/2/3../A/B/C..Z)?
5. List the latest released movies

Spark SQL

1. Create tables for movies.dat, users.dat and ratings.dat: Saving Tables from Spark SQL
2. Find the list of the oldest released movies.
3. How many movies are released each year?
4. How many number of movies are there for each rating?
5. How many users have rated each movie?

6. Wh rating for each movie?
7. What is the average rating for each movie?

Spark Data Frames

1. Prepare Movies data: Extracting the Year and Genre from the Text
2. Prepare Users data: Loading a double delimited csv file
3. Prepare Ratings data: Programmatically specifying a schema for the data frame
4. Import Data from URL: Scala
5. Save table without defining DDL in Hive
6. Broadcast Variable example
7. Accumulator example

Dataset: https://grouplens.org/datasets/movielens/1m/

Project Evaluation metrics:

Code:

- You are supposed to write a code in a modular fashion
- Safe: It can be used without causing harm.
- Testable: It can be tested at the code level.
- Maintainable: It can be maintained, even as your codebase grows.
- Portable: It works the same in every environment (operating system)
- You have to maintain your code on GitHub.
- You have to keep your GitHub repo public so that anyone can check your code.
- Proper readme file you have to maintain for any project development.
- You should include basic workflow and execution of the entire project in the readme file on GitHub
- Follow the coding standards: https://www.python.org/dev/peps/pep-0008/

Database:

- You are supposed to use a given dataset for this project which is a Cassandra database.
- https://astra.dev/ineuron

Cloud:

- You have to use any cloud platform for this entire solution hosting like AWS, Azure or GCP

API Details or User Interface:

- You have to expose your complete solution as an API or try to create a user interface for your model testing. Anything will be fine for us.

Logging:

- Logging is a must for every action performed by your code use the python logging library for this.

Ops Pipeline:

- If possible, you can try to use AI ops pipeline for project delivery Ex. DVC, MLflow , Sagemaker , Azure machine learning studio, Jenkins, Circle CI, Azure DevOps , TFX, Travis CI

Deployment:

- You can host your model in the cloud platform, edge devices, or maybe local, but with a proper justification of your system design.

Solutions Design:

- You have to submit complete solution design strategies in HLD and LLD document

System Architecture:

- You have to submit a system architecture design in your wireframe document and architecture document.

Latency for model response:

- You have to measure the response time of your model for a particular input of a dataset.

Optimization of solutions:

- Try to optimize your solution on code level, architecture level and mention all of these things in your final submission.
- Mention your test cases for your project.

Submission requirements:

High-level Document:

You have to create a high-level document design for your project. You can reference the HLD form below the link.

Sample link:

HLD Document Link

Low-level document:

You have to create a Low-level document design for your project; you can refer to the LLD from the below link.

Sample link

LLD Document Link

Architecture: You have to create an Architecture document design for your project; you can refer to the Architecture from the below link.

Sample link

Architecture sample link

Wireframe: You have to create a Wireframe document design for your project; refer to the Wireframe from the below link.

Demo link

Wireframe Document Link

Project code:

You have to submit your code GitHub repo in your dashboard when the final submission of your project.

Demo link

Project code sample link :

Detail project report:

You have to create a detailed project report and submit that document as per the given sample.

Demo link

DPR sample link

Project demo video:

You have to record a project demo video for at least 5 Minutes and submit that link as per the given demo.

Demo link

Project sample link :

The project LinkedIn a post:

You have to post your project detail on LinkedIn and submit that post link in your dashboard in your respective field.

Demo link

Linkedin post sample link :