CS5052 Coursework 1: Apache Spark

Overview

This piece of coursework involves gaining some knowledge and experience by working with Apache Spark in order to analyse a large dataset.

Pair Programming

This coursework is to be completed in self-selected pairs. Please find a programming partner and email me your self-selected group by **Friday 19 February 2021**. In the event you don't find a partner by the deadline, I will assign you into a pair.

Competencies

- Have a basic understanding of the Python programming language
- Understand Apache Spark and its components

Practical Requirements

You will develop a console-based application in **Python and Apache Spark** in order to analyse a large data set. The dataset is from the MovieLens website, which contains "27,000,000 ratings and 1,100,000 tag applications applied to 58,000 movies by 280,000 users". The datasets can be downloaded from the following link: https://studres.cs.st-andrews.ac.uk/CS5052/Practicals/

It is advisable for you to try your application on the small dataset (~1MB) before using the large dataset (> 200 MB). However, your application must work on the large set and implement the functionality described in Parts 1, 2 and 3. Consider designing a user friendly interface for your application, e.g., do not output all data directly to the console.

Part 1

To gain a grade of 14.5 you should implement a set of **core features** on Apache Spark in order to:

- Read the dataset using Apache Spark.
- Store the dataset using the methods supported by Apache Spark.
- Search user by id, show the number of movies/genre that he/she has watched
    - Given a list of users, search all movies watched by each user
- Search movie by id/title, show the average rating, the number of users that have watched the movie

- Search genre, show all movies in that genre
    - Given a list of genres, search all movies belonging to each genre
- Search movies by year
- List the top n movies with highest rating, ordered by the rating
- List the top n movies with the highest number of watches, ordered by the number of watches

Part 2

In order to obtain a grade up to 16.5 you must implement the following two **intermediate features**:

- Find the favourite genre of a given user, or group of users. Consider and justify how you will define ‘favourite’.
- Compare the movie tastes of two users. Consider and justify how you will compare and present the data.

Part 3

In order to obtain a grade greater than 16.5 you must implement some more **advanced features**. Part 3 should build on the functionality provided by Part 2 and provide interesting reports and/or insights into the dataset. Interesting can be interpreted anyway you see fit, but should include some form of statistics and/or prediction. Some suggestions are below:

- Cluster users by movie taste.
- Visualisation and interaction of the data set, using external libraries
- Provide movie recommendations, e.g., user x liked movies A,B and C therefore they might like movies X,Y and Z.

Deliverables

Deadline: 9 April 2021

Every student must submit a single ZIP file containing the following artifacts

TO BE PREPARED AS A GROUP AND INCLUDED IN YOUR INDIVIDUAL SUBMISSION:

- A link to a repository containing your Spark solution. Ideally you should use the School’s Mercurial service, however a GitHub repository is also acceptable.
    - The repository must contain a README file describing how to run the application.
    - **Please do not include the dataset with your submission**
- A short video clip demonstrating the execution of your application. This can be a simple 5-minute screen capture with you talking over the video to describe the

functionality of your system. Treat it as a video which gives a quick overview of your solution and the functionality it supports.

TO BE PREPARED INDIVIDUALLY:

- A pdf report (not more than 3000 words) describing the design, implementation, and any difficulties you encountered. In particular, it should include:
    - Summary of supported core (Part 1), intermediate (Part 2) and advanced (Part 3) features you have implemented. Where appropriate add some discussion / justification as to why they were included.
    - Include a table which presents an overview of each of the features implemented in Part 1,2 and 3.
    - Any problems that you encountered and your solutions.
    - Any diagrams / charts you feel are necessary.
    - A reflective summary discussing the lessons learnt and experience that you have gained after finishing this practical.
- A brief description of your individual contributions to this practical and any potential issues you encountered. This should be submitted individually by every student.

Marking

See the standard mark descriptors in the School Student Handbook:

https://info.cs.st-andrews.ac.uk/student-handbook/learning-teaching/feedback.html#Mark_Descriptors

Lateness

The standard penalty for late submission applies (Scheme B: 1 mark per 8 hour period, or part thereof):

https://info.cs.st-andrews.ac.uk/student-handbook/learning-teaching/assessment.html#lateness-penalties

Good Academic Practice

The University policy on Good Academic Practice applies:
https://www.st-andrews.ac.uk/students/rules/academicpractice/