DBMS Project Report
Hive Warehouse on Hadoop

MWB

Group Members:

1)Mehul Chaturvedi(20CS30032)

2)Mir Mohammad Wasif(20CS10035)

3)Bismay Parija(20CS10015)

Introduction

Background Information:

In the era of digital media and streaming services, people are constantly seeking quality content to consume. With thousands of movies available, finding the right film to watch can be a daunting task. This is especially true when considering the diversity of preferences, genres, and movie eras. In such a scenario, data-driven recommendations and insights can greatly improve the user experience by providing a customised list of movies that cater to their interests.

Motivation for the Project:

The primary motivation for this project is to create a system that utilises data from the MovieLens dataset to offer users insights into movies based on various criteria. By developing functionalities such as finding top-rated movies by decade or getting movies by year, users can easily discover movies that match their preferences. This project aims to demonstrate the power of data analytics and the potential for enhanced user experiences in the entertainment domain.

Problem Statement:

The problem this project aims to address is the challenge of discovering relevant and high-quality movies in a vast and diverse pool of available content. To accomplish this, we need to efficiently process and analyse large volumes of data using the Hadoop ecosystem, specifically the Hive data warehouse.

By utilising the MovieLens dataset and implementing various functionalities to filter and rank movies within the Hive warehouse, the project seeks to provide a user-friendly solution that helps users quickly find movies that cater to their preferences, saving time and enhancing their overall experience. This approach showcases the power of big data technologies in providing valuable insights and recommendations for users.

Objective

Primary Objectives:

- To develop a user-friendly platform that utilises the Hadoop ecosystem, specifically the Hive data warehouse, to process and analyse the MovieLens dataset.
- To create various functionalities that enable users to filter and rank movies based on specific criteria, such as genre, year, and user ratings.
- To showcase the potential of big data technologies in providing valuable insights and recommendations that cater to users' preferences and enhance their overall experience.

Expected Outcomes:

- A robust and efficient platform that effectively utilises the Hadoop ecosystem to process and analyse large volumes of movie data.
- A set of functionalities that allow users to discover high-quality and relevant movies quickly and easily.
- An improved user experience, as users will be able to find movies that align with their preferences in a more efficient and streamlined manner.
- A demonstration of the potential of big data technologies in the entertainment industry, specifically in the context of movie discovery and recommendation.

Methodology

The methodology followed in this project consists of several key steps, as described below:

Overview of the data used (MovieLens dataset):

The MovieLens dataset is a widely-used dataset for movie recommendation systems, containing user ratings and movie metadata. It is utilised in this project as the primary source of data for generating movie recommendations based on various criteria.

Data preprocessing and cleaning steps:

The initial dataset is preprocessed and cleaned to ensure that it conforms to the desired format for analysis. This involves converting the release_date field to a DATE type, ensuring all fields have appropriate data types, and removing any inconsistencies.

Technologies used (Hive, Python, etc.):

The project utilises the Hadoop ecosystem, specifically the Hive data warehouse, for processing and analysing the dataset. Python is used as the primary programming language, with the PyHive library enabling communication between Python and Hive.

It is crucial to understand the underlying technologies and concepts used in this project. Here, we provide brief explanations of HDFS, Hive, building a cluster, and their roles in the project.

Hadoop Distributed File System (HDFS):

HDFS is the primary storage system used in the Hadoop ecosystem. It is a distributed file system designed to store and manage large volumes of data across a cluster of machines. HDFS provides fault tolerance, high availability, and scalability, making it ideal for big data processing. In this project, the MovieLens dataset is stored and managed in HDFS, allowing for efficient data access and processing by the Hive data warehouse.

Apache Hive:

Hive is a data warehouse solution built on top of Hadoop, designed to facilitate querying and managing large datasets stored in HDFS. It uses a SQL-like language called HiveQL, which allows users to perform complex data analysis and manipulation using familiar SQL syntax. Hive translates HiveQL queries into a series of MapReduce jobs that run on the Hadoop cluster. In this project, Hive is utilised to process and analyse the MovieLens dataset and generate movie recommendations based on user-specified criteria.

Building a cluster:

A Hadoop cluster is a collection of machines, called nodes, that work together to store, process, and manage large volumes of data. The cluster consists of a single

NameNode, which manages the file system metadata, and multiple DataNodes, which store and process the actual data. Building a Hadoop cluster involves the following steps:

a. Installing Hadoop and its dependencies on all nodes.
b. Configuring the Hadoop environment, such as setting up the HDFS directory structure and defining the NameNode and DataNodes.
c. Starting the Hadoop services, such as the NameNode, DataNode, and YARN ResourceManager.
d. Testing the cluster to ensure that it is functioning correctly and efficiently.

Role of HDFS, Hive, and the cluster in the project:

In this project, HDFS serves as the primary storage system for the MovieLens dataset, ensuring that the data is available for processing and analysis. The Hadoop cluster provides the computational resources necessary to process and analyse the data using Hive. Hive acts as the primary interface for querying and manipulating the data stored in HDFS, enabling the generation of movie recommendations based on various criteria.

By leveraging the power of HDFS, Hive, and the Hadoop cluster, this project is able to efficiently process and analyse large volumes of movie data, providing users with a comprehensive and customizable movie recommendation system.

Explanation of the SQL queries and Python functions for various tasks (top-rated movies by decade, movies by year, etc.):

The Python code provided contains various functions that enable the retrieval and display of movie recommendations based on user-specified criteria. The following is a brief overview of the key functions:

search_movies(search_string):
This query returns a list of movies whose titles contain the given search string. It uses the LIKE operator with wildcards (%) to match any movie titles containing the search string.

get_movie(movie_id):
This query fetches the movie details for a specific movie_id. It simply selects all columns from the movies table where the movie_id matches the input parameter.

get_top_rated_movies(genre, limit=10):
This query returns the top-rated movies in a specific genre. It joins the ratings and movies tables and filters the result by the given genre. Then, it calculates the average rating and number of ratings for each movie, and returns the top 'limit' movies based on their average rating, requiring at least 10 ratings.

get_most_popular_movies(limit=10):
This query returns the most popular movies based on the number of ratings. It joins the ratings and movies tables, groups the result by movie_id and title, and then orders the movies by the number of ratings, returning the top 'limit' movies.

get_movies_by_year(year, limit=10):
This query returns the most popular movies released in a specific year. It joins the ratings and movies tables, filters the result by the given year, groups the result by movie_id and title, and then orders the movies by the number of ratings, returning the top 'limit' movies.

get_movies_by_genre(genre, limit=10):
This query returns a list of movies in a specific genre. It selects the movie_id and title from the movies table, where the genres column contains the given genre, and limits the result to the specified number of movies.

get_similar_movies(movie_id, limit=5):
This query returns a list of movies that have the same genres as the movie with the specified movie_id. It first fetches the genres of the input movie, and then selects movies with the same genres, excluding the input movie itself. The result is ordered by the number of ratings, returning the top 'limit' similar movies.

get_top_rated_movies_by_demographics(demographic, value, limit=10):
This query returns the top-rated movies for a specific demographic value (e.g., age, gender, occupation). It joins the ratings, movies, and users tables, filters the result by the given demographic value, and calculates the average rating and number of ratings for each movie. The top 'limit' movies are returned based on their average rating, requiring at least 10 ratings.

top_rated_movies_by_decade(decade, limit):
This query returns the top-rated movies for a specific decade. It joins the ratings and movies tables, filters the result by the given decade, groups the result by movie_id and title, and calculates the average rating for each movie. The top 'limit' movies are returned based on their average rating.

get_genre_preferences_by_demographics(demographic, value, limit=5):
This query returns the most popular genres for a specific demographic value (e.g., age, gender, occupation). It joins the ratings, movies, and users tables, filters the

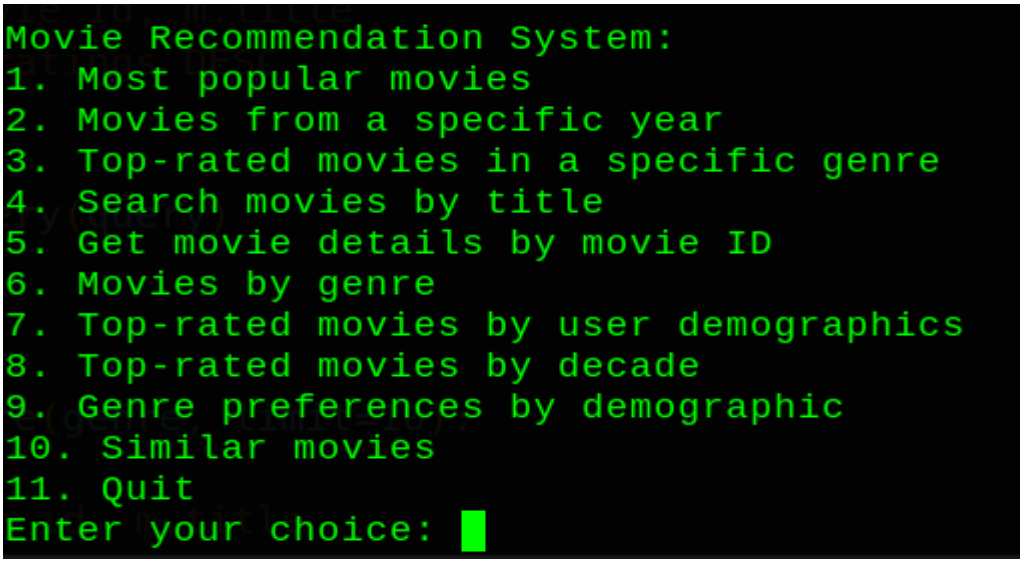
result by the given demographic value, and groups the result by the genres column. The top 'limit' genres are returned based on the number of ratings.

Each function contains an SQL query that extracts the relevant data from the Hive tables, processes it based on the user-specified criteria, and returns the result in a Pandas DataFrame. The main_menu() function provides an interactive interface for the user to access these functionalities.

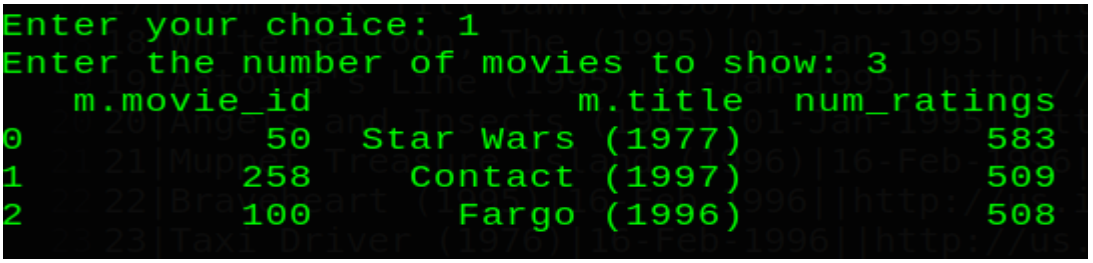
These functions and SQL queries are designed to work seamlessly together, enabling users to quickly and easily discover movies that match their preferences and interests.

Results

1. Home Screen :



2. 1st choice:



3. 2nd choice:

```
Enter your choice: 2
Enter the year: 1996
Enter the number of movies to show: 2
   m.movie_id                      m.title  num_ratings
0         286  English Patient, The (1996)          481
1         288                Scream (1996)          478
```

3. 3rd choice:

```
Enter your choice: 3
Enter the genre: Action
Enter the number of movies to show: 4
   m.movie_id                    m.title  avg_rating  num_ratings
0          50           Star Wars (1977)    4.358491          583
1         127      Godfather, The (1972)    4.283293          413
2         174  Raiders of the Lost Ark (1981)    4.252381          420
3         313            Titanic (1997)    4.245714          350
```

3. 4th choice:

```
Enter your choice: 4
Enter a search string: Usual Suspects, The (1995)
   movies.movie_id             movies.title movies.release_date   movies.genres
0               12  Usual Suspects, The (1995)       14-Aug-1995  Crime|Thriller
```

3. 5th choice:

```
Enter your choice: 5
Enter the movie ID: 1
   movies.movie_id     movies.title movies.release_date            movies.genres
0                1  Toy Story (1995)       01-Jan-1995  Animation|Children|Comedy
```

3. 6th choice:

```
Enter your choice: 6
Enter the genre: Comedy
Enter the number of movies to show: 5
   m.movie_id                                  m.title
0           1                       Toy Story (1995)
1           4                      Get Shorty (1995)
2           8                           Babe (1995)
3          13                 Mighty Aphrodite (1995)
4          16  French Twist (Gazon maudit) (1995)
```

3. 7th choice:

```
Enter your choice: 7
Enter the demographic (age, gender, or occupation): age
Enter the demographic value: 18
Enter the number of movies to show: 2
   m.movie_id          m.title  avg_rating  num_ratings
0         313   Titanic (1997)    4.818182           11
1          50  Star Wars (1977)    4.769231           13
```

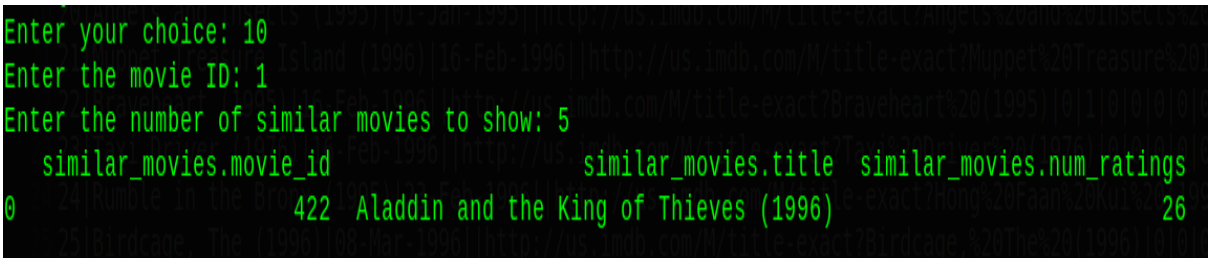
3. 8th choice:

```
Enter your choice: 8
Enter the starting year of the decade (e.g., 1980 for 1980s): 1990
Enter the number of movies to show: 3
   m.movie_id                                      m.title  average_rating
0        1189                        Prefontaine (1997)             5.0
1        1653  Entertaining Angels: The Dorothy Day Story (1996)        5.0
2         814                 Great Day in Harlem, A (1994)             5.0
```

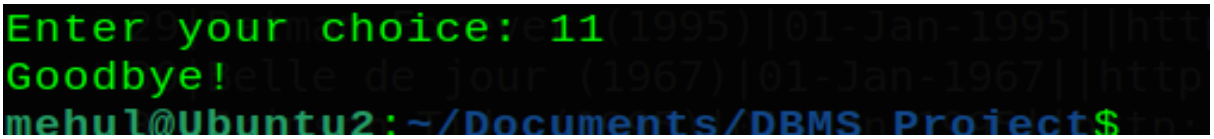
10. 9th choice:

```
Enter your choice: 9
Enter the demographic (age, gender, or occupation): age
Enter the demographic value: 18
Enter the number of genres to show: 1
  m.genres  num_ratings
0    Drama          257
```

11. 10th choice:



12. Exit:



References:

1.Hadoop: The Definitive Guide by Tom White. O'Reilly Media, 2012. [Book] (https://www.oreilly.com/library/view/hadoop-the-definitive/9781449328917/)
This book provides a comprehensive introduction to Hadoop and its ecosystem, including HDFS, MapReduce, and other related technologies.

2. Learning Apache Hive Essentials by Dayong Du. Packt Publishing, 2015. [Book] (https://www.packtpub.com/product/learning-hive/9781787282393)
This book covers the basics of Hive, along with advanced features and use cases, helping you understand how to use Hive for big data processing.

3.Apache Hadoop: Official Documentation [Online Resource] (https://hadoop.apache.org/docs/current/)
The official documentation for Apache Hadoop provides a wealth of information on Hadoop, HDFS, and related projects.

3.Apache Hive: Official Documentation [Online Resource] (https://cwiki.apache.org/confluence/display/Hive/Home)
The official documentation for Apache Hive offers in-depth information on Hive features, functions, and best practices.

5.The Hadoop Ecosystem Table by Javi Roman. [Blog Post] (https://hadoopecosystemtable.github.io/)

This resource provides a comprehensive list of Hadoop-related technologies, including Hive, HDFS, and many others, with brief descriptions and links to their official websites.

6.MovieLens Dataset by GroupLens [Dataset]
(https://grouplens.org/datasets/movielens/)
The MovieLens dataset is a collection of movie ratings from the MovieLens website, which has been widely used for movie recommendation research and big data projects.

7.Hadoop and Hive for Big Data Processing [Research Paper]
(https://ieeexplore.ieee.org/document/6398010)
This research paper discusses the use of Hadoop and Hive for big data processing, providing insights into the architecture and use cases of these technologies.

8.Tutorials Point Hive Installation
https://www.tutorialspoint.com/hive/hive_installation.htm#