Movie Recommendation System

TEAM 2: Zongyao Li, 002191556 Xuliang Mei, 001305818 Jiaqi Wang 001023711

Use cases

Statistics

- Users can see some statistic information about the movies based on all the historical data. Eg:
  - All-time most popular movie list
  - Recent most popular movie list
  - Most popular movies in different genres
  - etc…

Personalized Recommendation

- Specify a user, list the movie recommendation which is personalized for the user

Similar Movie

- Specify a movie, list the movie recommendation which is composed by all the similar movie

Methodology

- For Statistic, we will first store all the data into **Spark**, then do all the statistics via **Spark SQL**
- For Personalized Recommendation, we will use **Collaborative Filtering**, take all the rating data into consideration.
- For Similar Movie, we will use **TD-IDF** algorithm to calculate the similarity between movies

Data sources

The data source is Kaggle's MovieLens 20M Dataset

The magnitude

genome_scores.csv: 11.7m	genome_tags.csv: 1128	link.csv: 27.3k	movie.csv: 27.3k	rating.csv:20.0m	tag.csv: 466k

Week 9: learn about the missing part of the new tech and adjust the Dev environment

Week10: define the data format, preprocess the data, add dependencies and construct the framework of the project

Week11: compose the recommendation algorithm, complete the recommendation algo and train models

Week12: set up the front-end(if have time) and manage the flow control

Week13: final adjustment

Milestones

Code Composition and Repository

Code Composition

- Scala:
  - Statistics
  - Collaborative filtering
  - Td-idf
  - Front-end display with scala play UI(maybe)

Repository

- Github Link: MovieRecommendation

Acceptance criteria

1s	Display all the statistic lists as soon as it’s requested
5s	Display the user personalized recommendation in reasonable time(maybe below 5s)
3s	Display the similar movies in reasonable time(maybe below 3s)

Goals of the project

- Take comprehensive use of collaborative filtering algorithms and td-idf to provide mixed recommendations.
- Get handy with Scala when building big-data systems
- Optimize the structure of the system and the algorithms to make it respond as fast as possible

Thanks!