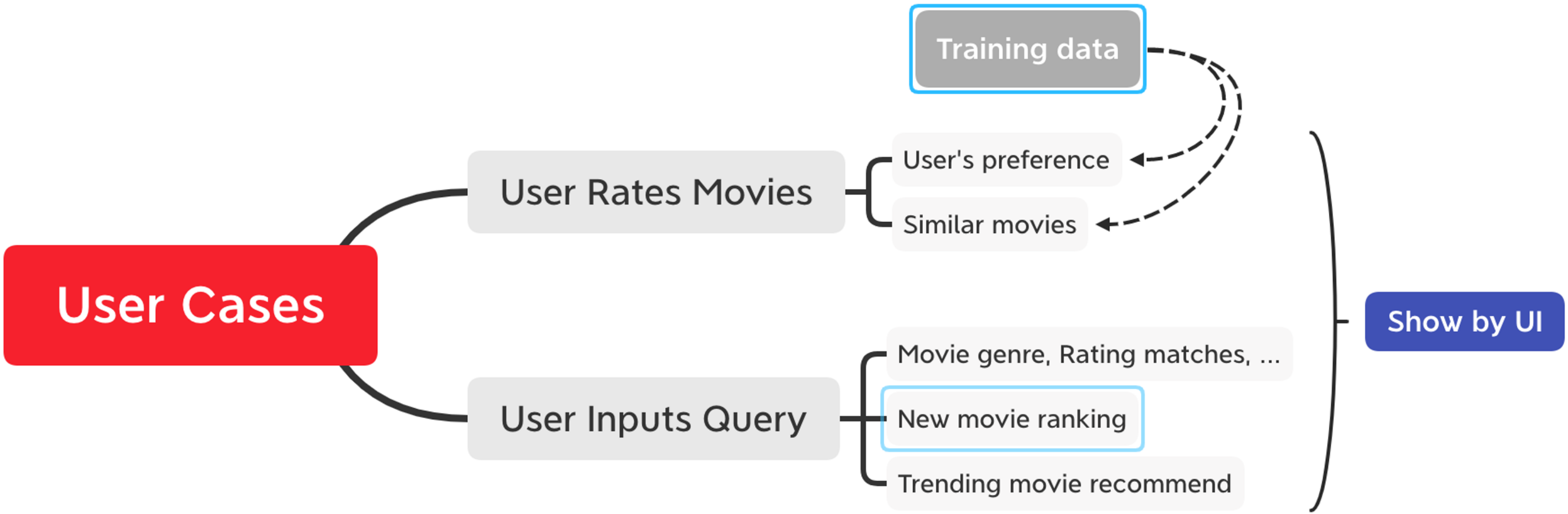
Movie Recommendation

Team III

Mengzhe ZHANG

Rongqi SUN

Yue FANG

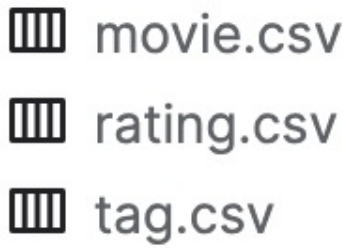

User Cases
User Rates Movies
User Inputs Query
Training data
User's preference
Similar movies
Movie genre, Rating matches, ...
New movie ranking
Trending movie recommend
Show by UI

Methodology

- **Statistics:**
  - Use Spark Core + Spark SQL deal dataset
    - Load dataset with MongoDB through Spark SQL
- **Recommendation:**
  - Algorithms use ALS to summarize user's preference and similar movies
  - Use Spark Core + Spark MLlib to implement recommendation methods

Data Source

- **MovieLens 20M Dataset**
  - https://www.kaggle.com/grouplens/movielens-20m-dataset
  - **It contains 20000263 ratings and 465564 tag applications across 27278 movies.**
  - **These data were created by 138493 users between January 09, 1995 and March 31, 2015.**
  - **Users were selected at random for inclusion. All selected users had rated at least 20 movies.**



Milestones

Milestones	Time
Data cleaning and processing Unit Test	3.22 – 3.28
Recommendation methods implementing Unit Test	3.29 – 4.7
Setup UI Implement visualization	4.8 – 4.14
Final model and use cases testing System Test	4.15 – 4.21

Repository

- Scala:
- Recommendation Part
- User Interface Use Play
- MongDB for dataset using SparkSQL
- Repo: https://github.com/tracy626/CSYE7200_FinalProj_Team3

Acceptance criteria

- User is able to:
  - Rate movie and get feedback about similar movies recommendation according to rating history (predict user's preference)
  - Ask for recommendation of new movies and get list by rank
  - Get trending movie recommendation
  - Get User-Based Recommend result in less than 3 seconds
  - Get Statistics Recommend result in less than 1 second
  - …

Goals of the Project

- Clean and process raw dataset
- Analyze movies rating with other features
- Input recommendation and analysis results to database
- Create UI for recommendation system
- Create reactive page for user to filter recommendation results