Big Data Application - Project proposal

Group information

ID	Name
21127170	Nguyễn Thế Thiện
21127326	Nguyễn Trần Trung Kiên
21127329	Châu Tấn Kiệt

Problem

Title: Real-time anime recommendation system based on user ratings.

Description:

Analyze real-time anime ratings data using Big Data and Machine Learning tools, in order to recommend animes a user has yet to watch, based on the user's rating history.

Input: User's rating history on watched animes.

Output: Ranking list for recommended animes fitting user's tastes.

Dataset

So far we do not have any intentions on using multiple datasets to perform further analyses.

We intend to use **Anime Dataset 2023** [1] by username Sajid from Kaggle, which is a collection of user and anime ratings on one of the largest anime databases and communities - MyAnimeList (myanimelist.net).

The dataset consists mainly of:

anime-dataset-2023.csv: Details of around 25K anime titles on MyAnimeList, including **related information**: anime name (original, translated), description, media form (types), number of episodes, airing time, age rating, producers, studio, etc. and **user-generated ratings**: such as average score, number of times marked favorites, number of users scored, number of users marked in personal anime lists (members), popularity, etc.

user-details-2023.csv: Details of around 730K users registered on MyAnimeList, including **personal information** such as gender, birthday, location, date joined, days spent watching anime, **activity scores**: number of days spent watching, number of anime completed / on hold / dropped, etc.

users-score-2023.csv: User ratings on anime titles, provided by 270K users on 16K anime titles, with a total of 24.3M samples, **99.48% sparsity rate** for only the observed users and anime titles.

final_animedataset.csv: Another dataset version containing user ratings and anime details, all in a single file, based on 2018 data (different from **users-score-2023.csv**). It could be used to rapidly and simply test models before putting more effort into further implementations.

Data storing and processing tool(s)

Redis [2]: It is one of the most popular NoSQL database used widely by companies, with reasonable pricing options for students and supports real-time data streaming.

Apache Spark [3] for real-time data processing, since we already have prior experience working with the tool, as well as its great (though slowly declining) relevance in the near future of this field.

Recommender system (RS)

As of now, the team has yet to decide if we should spend more effort into implementing a state-of-the-art Deep Learning RS.

Apache Spark MLlib RS [3] as it comes with Apache Spark mentioned above, and is a well-known standard Machine Learning library. It contains Alternating Least Squares (ALS) matrix factorization to learn latent factors.

ocelma's python-recsys [4] is also a well-known (even though old) library mainly uses Singular Value Decomposition (SVD).

Data visualization

Type(s)

Network graph in order to present the interactions between the observed users and items.

Tool(s)

NumPy Matplotlib [5] is a well-known and easy-to-use data visualization Python library for data scientists. Also it supports real-time data visualization.

NetworkX [6] is a graph generating Python library that can work with Matplotlib to visualize the observed data.

Tasks

Main tasks

1. **Data ingestion**: Set up Redis database with imported data from the dataset files, and set up data streaming connection.
2. **Data streaming & preprocessing**: Apache Spark Streaming simulates real-time data from Redis database, then clean and prepare the raw data before feeding into the recommendation system.
3. **Real-time RS model training**: Pre-built model from MLlib is trained by feeding real-time data, from Spark Streaming.
4. **RS in use**: Input user's rating history to predict a ranking list for recommended animes which user has not watched.
5. **Real-time dashboard**: Visualize analyzed data and predictions with NetworkX-assisted Matplotlib.

Plan timelines

Note: Date used here is in form (YY/)MM/DD, time used here is in 24-hour format: HH:MM.

The main plan is to divide the workload into 8 sprints throughout the span of 4 weeks, starting from 03/03 to 04/01, meaning 2 sprints per week.

1. Sprint 1-2 (03/04 - 03/10): Preparation.

- Set up Redis database from dataset files and configure real-time data streaming to Spark.
- Figure data preprocessing strategies, perform data preprocessing on dataset using Spark.
- Set up RS model from MLlib, learn its required input and output forms for training and testing.
- Figure out how to save RS model into a file for further training.

2. Sprint 3-4 (03/11 - 03/17): Tool testing and systematic setups.

- Perform real-time data processing using Redis and Spark, with data visualization using NetworkX and Matplotlib.
- Set up a basic user interface to apply the use of RS.
- Test RS model training on small scale with multiple batches, with saving and loading RS model.

3. Sprint 5-6 (03/18 - 03/24): Main event.

- **Perform real-time RS model training on dataset.**
- Research and experiment documentation.
- Application of RS model into the problem.

4. Sprint 7-8 (03/25 - 03/31): Project conclusion.

- Research and experiment documentation and presentation with Canva [7].
- Graphical demonstration.

5. Sprint 9+ (04/01 - 04/12): Backup.

Assignments

Note: Date used here is in form (YY/)MM/DD, time used here is in 24-hour format: HH:MM.

All works must be draft-documented in text files (md, txt, pdf, docs) upon finished working.

Sprint no.	Who	Job(s)	Tool(s)	Start	Due	Note
1	Kiên	Set up real-time data streaming from database	Redis, Spark	03/04 06:00	03/06 21:00	Write down how to setup
1	Thiện	Figure data preprocessing strategies	Spark	03/04 06:00	03/06 21:00	Explain the strategies

Sprint no.	Who	Job(s)	Tool(s)	Start	Due	Note
1	Kiệt	Choose and set up RS model, learn its inputs from Spark and outputs	Spark, MLlib	03/04 06:00	03/06 21:00	Explain why choosing the model and how it works briefly
2	Kiên	Configure real-time data streaming from Redis to Spark	Redis, Spark	03/07 06:00	03/10 21:00	Test Spark Streaming techniques if needed.
2	Thiện	Perform data preprocessing on dataset	Spark	03/07 06:00	03/10 21:00	Technique results and plotting if necessary.
2	Kiệt	Figure out how to save RS model into a file for further training	MLlib	03/07 06:00	03/10 21:00	Test and write down how to save and load.
3-4	Kiệt	Perform real-time data processing with data visualization	Redis, Spark, NetworkX, Matplotlib	03/11 06:00	03/17 21:00	
3-4	Kiên	Set up basic UI	C++, Python, JavaScript (?)	03/11 06:00	03/17 21:00	
3-4	Thiện	Test RS model training on small scale with multiple batches, with saving and loading RS model	Spark, MLlib	03/11 06:00	03/17 21:00	
5-6	Kiên	Perform real-time RS model training on dataset	MLlib, Redis, Spark	03/18 06:00	03/21 21:00	
5-6	Kiệt	Research and experiment documentation	LaTeX	03/18 06:00	03/24 21:00	
6	Thiện	Application of RS model into the problem	C++, Python, JavaScript (?)	03/22 06:00	03/24 21:00	
7-8	Kiên	Graphical demonstration	Screen recorder	03/25 06:00	03/31 21:00	

Sprint no.	Who	Job(s)	Tool(s)	Start	Due	Note
7-8	Kiệt	Documentation finishing	LaTeX	03/25 06:00	03/31 21:00	
7-8	Thiện	Presentation with Canva	Canva	03/25 06:00	03/31 21:00	

References

[1] Sajid Uddin (2023). Anime Dataset 2023. *Kaggle: Your Machine Learning and Data Science Community*. https://www.kaggle.com/datasets/dbdmobile/myanimelist-dataset?resource=download

[2] Salvatore Sanfilippo (2009). Redis 7.4.2 (2025). https://redis.io

[3] Matei Zaharia (2014). Apache Spark 3.5.4 (2024). https://spark.apache.org/

[4] Oscar Celma, Daniel Eisner, et al. (2011). python-recsys: A python library for implementing a recommender system. https://github.com/ocelma/python-recsys

[5] John D. Hunter (2003). Matplotlib 3.10.0 (2024). https://matplotlib.org/

[6] Aric Hagberg, Pieter Swart, Dan Schult (2005). NetworkX 3.4.2 (2024). https://networkx.org/

[7] Melanie Perkins, Cliff Obrecht, Cameron Adams (2013). Canva. https://canva.com