CP421 Data Mining - Project
Due Date: Dec 6th, 2023 at 11:59 PM

Assignment Submission Guidelines

1. **File Naming**: To submit the materials for your project, first create a directory named *login_pj*. Inside this directory, include your project report (in Word or PDF format), any presentation materials, and your notebook. Once you've placed all these items in the directory, compress it into a .zip file. Please ensure you use the .zip format for compression. Upload this zipped file to MyLS. We request that each team submits only one file. Make sure to list all team members' names within the report.

2. **Late Submissions**: If you submit your assignment within 24 hours post the deadline, your grade will be reduced by 50%. Unfortunately, we cannot accept submissions made beyond 24 hours from the deadline, and such submissions will receive a grade of 0.

3. **Plagiarism Policy**: At WLU, we take academic integrity seriously. All submitted code will undergo a plagiarism check. Engaging in plagiarism can have significant academic consequences.

1 Deliverables

For your project, there are multiple components that contribute to your final project score. All evaluations will be conducted on a team basis.

- **Project Proposal**. Each team, with a maximum of 5 members, should submit a proposal before beginning their project. Describe your intended method in under 500 words. Our TA will review the topics to ensure diverse approaches among teams. Emphasize your unique contribution succinctly. **The deadline of submitting the proposal is Nov 13, 2023.**

- **Project Presentation**. Every team is allotted 15 minutes to present their project to their peers. The presentation should effectively communicate the core concepts, techniques, and outcomes of the project. This includes the problems tackled, the rationale behind the chosen analytical approaches, and pertinent assessments, contributions, and points of discussion. The presentations are scheduled for **Nov 29, Dec 4, and Dec 6**. Teams will be randomly assigned to these dates to ensure equal opportunity for each date. Once teams are established, the specific presentation dates for each will be disclosed.

- **Project Paper**: At the culmination of your project, a comprehensive paper is expected. This document should encapsulate the various stages of your project development, detailing data processing techniques, methodologies employed, and the insights garnered. Submissions for the project paper should be in either Word or PDF format.

- **Project Notebook**: This refers to a Jupyter notebook containing the actual code used to implement the proposed solution. It should be organized and annotated to clarify each step and decision made during the implementation. The notebook should provide a hands-on view of your data manipulation, analysis, and final solution.

2 Problem Definition

Over recent decades, as platforms like YouTube, Amazon, Netflix, and other web services have flourished, the influence of recommender systems in our daily lives has expanded significantly. These systems have become integral to a range of online experiences, from e-commerce (recommending products to potential buyers) to online advertising (curating content that aligns with user preferences). Essentially, recommender systems employ algorithms to suggest items that are likely to resonate with users. These items could range from movies to articles, products, and beyond, depending on the specific industry. For this project, leveraging the concepts taught in this course, you'll develop a recommendation system focused on suggesting books to users. You're encouraged to explore various approaches to enhance the efficiency and accuracy of your recommendation model.

2.1 General Description

You will be using the *Book Recommendation* dataset from Kaggle. The whole dataset was collected by Cai-Nicolas Ziegler in a 4-week crawl (August / September 2004) from the Book-Crossing community with kind permission from Ron Hornbaker, CTO of Humankind Systems. It contains 278,858 users (anonymized but with demographic information) providing 1,149,780 ratings (explicit / implicit) about 271,379 books. The complete dataset can be found at `https://www.kaggle.com/arashnic/book-recommendation-dataset`. **While developing your method, you are free to use some or all of the data files.**

2.2 Formatting and Encoding

The dataset files contains 3 files.

1. **Users**. Contains the users. Note that user IDs (User-ID) have been anonymized and map to integers. Demographic data is provided (Location, Age) if available. Otherwise, these fields contain NULL-values.
2. **Books** Books are identified by their respective ISBN. Invalid ISBNs have already been removed from the dataset. Moreover, some content-based information is given (Book-Title, Book-Author, Year-Of-Publication, Publisher), obtained from Amazon Web Services. Note that in case of several authors, only the first is provided. URLs linking to cover images are also given, appearing in three different flavours (Image-URL-S, Image-URL-M, Image-URL-L), i.e., small, medium, large. These URLs point to the Amazon web site.
3. **Ratings** Contains the book rating information. Ratings (Book-Rating) are either explicit, expressed on a scale from 1-10 (higher values denoting higher appreciation), or implicit, expressed by 0.

3 Evaluation and Experiments

To demonstrate the effectiveness of your proposal, you are supposed to compare your method with some baselines. You can start with some simple methods listed below.

- Use the global mean to do the prediction;
- Use the user means as the prediction;
- Use the item means as the prediction;
- Use the classic method discussed in class.

Training and testing data To perform an evaluation, you need to first split the data into training and testing datasets. Using the training data, you can develop a recommender system with your proposed method. For the testing data, you can randomly select some cells from the user-item matrix, e.g., 20%, and assume that they are unknown. Then by feeding the testing data into the recommender system, it can estimate ratings for the missing cells. Finally, you can compare the system outputs against the real values.
Performance Evaluation You will use the root mean-square error (RMSE) to measure the performance. You can also use other evaluation metrics, such as Precision, Recall, and F-measure, as they see fit.

4 Grading

The project grade will be distributed as follows:

- **Project Paper**: 20% of the total grade.
- **Project Presentation**: 30% of the total grade.
- **Quality of the Project**: 50% of the total grade.

5 Sample Ideas and Some State-of-the-arts

Emphasis will be placed on innovation during the evaluation process. Relying solely on pre-existing packages like *surprise*, or employing widely-recognized methods such as traditional content-based recommendation or collaborative filtering, will not suffice. Projects of this nature can expect a reduced score. For inspiration, refer to the state-of-the-art ideas outlined below.

1. Hybrid Methods

- **Hybrid Collaborative-Content models**: Combine collaborative and content-based filtering.
- **Weighted**: Weighted sum of scores from different methods.
- **Feature Combination**: Combine features from different sources into one algorithm.

2. Deep Learning Approaches
    - **AutoEncoders**: Compress the user-item interaction matrix.
    - **Neural Collaborative Filtering (NCF)**: Combine Generalized Matrix Factorization (GMF) and Multi-Layer Perceptron (MLP).
    - **RNNs**: Use for session-based recommendations.
    - **BERT for Recommendations**: Adapt user interaction sequences as sentences.
3. Knowledge-Based Recommendations
    - Rely on explicit knowledge about users and items. Might ask users for more information about preferences.
4. Context-Aware Recommendations
    - Consider context (e.g., time of day, location, mood) in recommendations.
5. Factorization Machines
    - General predictor for real valued feature vectors. Especially effective for sparse datasets and modeling feature interactions.
6. Reinforcement Learning for Recommendations
    - Treat recommendations as a reinforcement learning problem where recommendations are actions and user feedback is the reward.
7. Graph-based Recommendations
    - Use graph databases and algorithms like node embeddings, especially when entity relationships matter.
8. Hybrid Deep Models
    - Combine CNNs for image-based recommendations with RNNs for sequence-based recommendations.