Group Project Tutorial

Instructions:

- This is a group project. Each group can have up to 5 members. Each group elects a leader and submits the list of group members to the TA (email: 72405526@cityu-dg.edu.cn) by October 26th.
- Each group is required to deliver a 5-minute in-class presentation of your project. At the same time, each group submits a technical report, in which the process of data processing and analysis, findings and conclusions are required to be clearly stated. Technical report, source code submitted via canvas.
- The project scope is open-ended. You are encouraged to explore innovative ideas using any combination of data analysis, machine learning models, and visualization techniques.
- Should you have any question, please feel free to consult with the TAs directly after class.

Projects information:

For flexibility, you are free to explore either of the two projects listed below:

Project 1: Create a book recommendation system.

- **Overview**: Throughout this semester's coursework, we have explored knowledge related to Collaborative Filtering and Social Network Analysis within recommender systems, and now it is time to apply these concepts! I will provide you with a book recommendation dataset to develop a book recommendation system. This is an open-ended project, allowing you to pursue any explorations that activate your interest. For instance, you may implement Collaborative Filtering based on items, users, or content. The fundamental objective of this project is to create a basic recommendation function within the book recommendation system: given a book title/name, you must recommend five similar book titles/names. Beyond this, I encourage you to freely explore more innovative recommendation methods and content, and to present the implemented functionalities in an interactive format, such as a web application or mobile app.
- **Dataset**:
  - **Website**: https://www.kaggle.com/datasets/arashnic/book-recommendation-dataset/data
  - **Content**:
    - **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.
    - **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.
    - **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.
- I would like you to follow the steps below to conduct the experiment and write the report:
  - **Extract information about the dataset (20% of the total score)**:
    - Display the shape of the dataset.

        - Display the total number of null values in dataset.
        - Detect the null cols in the dataset.
        - Create the comparison dataset.
        - Display information about dataset.
        - Display number of unique values in each column in the dataset.
        - Display the data types of the dataset.
    - **Data Exploration and Preprocessing (20% of the total score)**:
        - Examine data characteristics, distributions, and check for missing values.
        - Process data based on its type, including handling missing values, feature encoding, standardization, or normalization.
        - Visualize the data distribution of interest.
    - **Algorithms Definition (30% of the total score)**: Define the recommendation algorithms to be utilized.
        - Note: You are required to employ a minimum of three recommendation algorithms. The algorithm consists of at least three algorithms:
            - One algorithm discussed in the lecture.
            - LightGBM: A Highly Efficient Gradient Boosting Decision Tree (https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf)
            - Choose at least one from the Reference Algorithms to reproduce.
        - Reference Algorithms:
            - LightFM: Metadata Embeddings for User and Item Cold-start Recommendations (https://arxiv.org/abs/1507.08439)
            - DCN V2: Improved Deep & Cross Network and Practical Lessons for Web-scale Learning to Rank Systems (https://arxiv.org/pdf/2008.13535)
            - DeepFM: A Factorization-Machine based Neural Network for CTR Prediction (https://www.ijcai.org/proceedings/2017/0239.pdf)
            - BPR: Bayesian Personalized Ranking from Implicit Feedback (https://arxiv.org/pdf/1205.2618)
            - DIN: Deep Interest Network for Click-Through Rate Prediction (https://arxiv.org/pdf/1706.06978)
    - **Book Recommendation (20% of the total score)**: Implement book recommendation using various defined recommendation algorithms.
    - Present your book recommendation system in the form of a web or mobile application **(10% of the total score)**.

Project 2: Simulate participation in a recommendation system challenge competition.

- **Overview**: While the public’s now listening to all kinds of music, algorithms still struggle in key areas. Without enough historical data, how would an algorithm know if listeners will like a new song or a new artist? And, how would it know what songs to recommend brand new users?WSDM has challenged the Kaggle ML community to help solve these problems and build a better music recommendation system. The dataset is from KKBOX, Asia’s leading music streaming service, holding the world’s most comprehensive Asia-Pop music library with over 30 million tracks. They currently use a collaborative filtering based algorithm with matrix factorization and word embedding in their recommendation system but believe new techniques could lead to better results.

- **Dataset**:
  - **Website**: https://www.kaggle.com/competitions/kkbox-music-recommendation-challenge/overview
  - **Key Features**:
    - **msno**: user id
    - **song_id**: song id
    - **source_system_tab**: the name of the tab where the event was triggered. System tabs are used to categorize KKBOX mobile apps functions. For example, tab my library contains functions to manipulate the local storage, and tab search contains functions relating to search.
    - **source_screen_name**: name of the layout a user sees.
    - **source_type**: an entry point a user first plays music on mobile apps. An entry point could be album, online-playlist, song .. etc.
    - **target**: this is the target variable. target=1 means there are recurring listening event(s) triggered within a month after the user's very first observable listening event, target=0 otherwise .
- I would like you to follow the steps below to conduct the experiment and write the report:
  - **Data Exploration and Preprocessing (30% of the total score)**:
    - Examine data characteristics, distributions, and check for missing values.
    - Process data based on its type, including handling missing values, feature encoding, standardization, or normalization.
  - **Dataset Splitting**: You can find train and test set in dataset website (https://www.kaggle.com/competitions/kkbox-music-recommendation-challenge/data).
  - **Model Definition (10% of the total score)**: Choose different recommentation models.
    - Note: You are required to employ a minimum of three recommendation algorithms. The algorithm consists of at least three algorithms:
      - One algorithm discussed in the lecture.
      - LightGBM: A Highly Efficient Gradient Boosting Decision Tree (https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf)
      - Choose at least one from the Reference Algorithms to reproduce.
    - Reference Algorithms:
      - LightFM: Metadata Embeddings for User and Item Cold-start Recommendations (https://arxiv.org/abs/1507.08439)
      - DCN V2: Improved Deep & Cross Network and Practical Lessons for Web-scale Learning to Rank Systems (https://arxiv.org/pdf/2008.13535)
      - DeepFM: A Factorization-Machine based Neural Network for CTR Prediction (https://www.ijcai.org/proceedings/2017/0239.pdf)
      - BPR: Bayesian Personalized Ranking from Implicit Feedback (https://arxiv.org/pdf/1205.2618)
      - DIN: Deep Interest Network for Click-Through Rate Prediction (https://arxiv.org/pdf/1706.06978)

- **Model Training (10% of the total score)**: Train each model on the training dataset, ensuring to set appropriate parameters for each model.
- **Model Evaluation**: Evaluate on area under the ROC curve between the predicted probability and the observed target.
- **Hyperparameter Tuning (10% of the total score)**: Explore different hyperparameters for tuning and select the best-performing model as the final model.
- **Model Selection (10% of the total score)**:
  - Select the best-performing model based on evaluation metrics.
  - Document the final model configuration, including hyperparameter values.
- **Result Interpretation (30% of the total score)**:
  - Summarize findings from model evaluation, highlighting strengths and weaknesses of the chosen models.
  - Provide recommendations for future work or improvements based on the analysis.