IS4242
INTELLIGENT SYSTEMS & TECHNIQUES

Group Project

Due date: 5-Nov, 11:59 PM

Objective

- Synthesize what you have learned so far
- Predictive Analytics for Improving Customer Engagement
  - Building a recommendation engine

Data: MovieLens

- 100,000 ratings (1-5) from 943 users on 1682 movies.

- Each user has rated at least 20 movies.

- Demographic info. for the users (age, gender, occupation, zip)
  - Zip: locations within the US

- Collected over a period of seven months from September 19th, 1997, through April 22nd, 1998

Data: Files

- *u.data* -- The complete data set, 100,000 ratings by 943 users on 1682 items.
- Each user has rated at least 20 movies. Users and items are numbered consecutively from 1.
- The data is randomly ordered. This is a tab-separated list of: user id, item id, rating, timestamp.
- You can ignore the temporal aspect.

Data: Files

- *u.user* -- Demographic information about the users
  - Tab separated list of: user id, age, gender, occupation, zip code

- *u.item* -- Information about the items (movies)
  - Tab separated list of movie-id, movie-title, release-date, video-release-date, IMDb-URL, unknown, Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Mystery, etc.
  - The last 19 fields are the genres, a 1 indicates the movie is of that genre, a 0 indicates it is not; movies can be in several genres at once.

- The user ids and movie ids are the ones used in the u.data data set.

Training and Test Splits

- We have split the *u.data* into two sets: *ua.train*, and *ua.test* so that there are exactly 10 ratings per user in the test set.

- Train data: *ua.train*
  - Use this data for model building
- Test data: *ua.test*
  - Use this data for evaluation

- You may conduct pre-processing on the test data as well. However, make sure that this data is not used for training your model (if you do so, you will be awarded only 20% of the marks for the approach)

Target Variable and Features

- Binary Target Variable: Like or Dislike
  - Transform the ratings from a discrete range 1-5 into a binary variable
    - Like: Rating $\geq$ 4, Dislike: Rating <4

- Features:
  - Movies: Genre, Title, Movie release date, etc.
  - Users: Age, Gender, Occupation, etc.

Data Schematics

- The data may be quite sparse
- It may contain missing values
- Feature engineering may prove to be helpful
- There is no restriction on how you conduct data pre-processing, just make sure you are able to explain why each of the steps are performed

Approaches

- A total of 3 approaches

- In all 3 cases you are expected to do data exploration – pre-processing - feature engineering - model building, and evaluation steps

- During model building, you are strongly recommended to conduct hyper-parameter tuning using cross validation techniques (we will discuss these in the next tutorial)

- Evaluation may differ based on the task, this is discussed for each approach separately

Approach – 1: Neural Networks

- Pre-process the dataset to extract both user and movie features.
  - For example (you may follow one or none of these approaches):
    - You may drop variables in both user demographics and movie information that you think are meaningless to be considered as input features.
    - Filter out movies with missing information (if any).
    - Concatenate user and movie features as the final input

- Build a classifier model based on Multi-layer perceptron to predict the target variable (discussed in tutorial 9)
  - Fine-tuning may help you to obtain good results
    - Again, you are free to tune all or some of the aspects (number of neurons, number of hidden layers, activation function, etc.) of the neural network model

Approach 1: Model Evaluation

Evaluation over the test data

- Precision, Recall and F1-score
- You are encouraged to use lift value, cumulative lift value, Youden's index, etc. to identify appropriate threshold for classification
  - Make sure to explain the threshold you use

Approach 2: Collaborative Filtering

- Build a user-based collaborative filtering by applying SVD on the user-movie ratings data (discussed in tutorial 5)

- You are free to select the number of latent factors, number of closest users to consider while identifying top movies to recommend.
  - However, keep the top-movies to recommend to a maximum of 6 (i.e., K=6 in KNN).

- Evaluation over the *test data* (this is a hard problem; you may have low performance):
  - Percentage of correct predictions
  - Mean percentage of correct predictions for different values of K
  - You are also encouraged to use any other evaluation metrics you may find relevant
    - Make sure to explain your reasoning if you do so

Approach 3: Multi-Armed Bandits

- Data: Use the entire dataset (u.data), but focus on movies with significant user feedback
  - Specifically, select movies with greater than 200 ratings

- Divide the dataset into four groups based on gender and age as follows:
  - Male, Young (age $<$ 30)
  - Male, Adult (age $\geq$ 30)
  - Female, Young (age $<$ 30)
  - Female, Adult (age $\geq$ 30)

- This helps in reducing the heterogeneity across users, when the movies are randomized with in the group

Approach 3: Multi-Armed Bandits

- Implement Epsilon-Greedy Multi-Armed Bandit algorithm (discussed in week-6)
- Recommend top 5 movies for each of these groups
- Fine-tune the algorithm to optimize movie recommendations
  - Experiment with different values of parameters (epsilon values) to find the best configuration
- Evaluation: Percentage of overlap between top 5 recommended movies with five movies that have the highest percentage of likes in the focal group

Specific Tasks and Points

- Groups of 3:
  - Implement and document approaches 1, 2, and 3
  - 3*8 = 24 (implementation and results)
- Report: 6 points
- *Peer reviews* will be considered in providing individual grade

Report: Documentation

- It is important to explain each step you perform (in pre-processing, feature engineering, model training, evaluation, etc)

- Ask why you are performing each step and write the reason

- Explain what inference you draw or what you observe after each step (e.g., from descriptive statistics)

- Provide performance metrics (e.g., on test data) in a table for each approach

- Finally, compare the recommendation approaches by drawing on their limitations

Submission

- Per group:

  1. One (or more) Jupyter Notebook(s)
  2. Also submit the same notebook(s) converted to HTML
  3. Document (filename: *main*, format: word/pdf…) containing

     Names and IDs of all group members and a report

- Upload 1 zipped folder containing all files

- Deadline: 5-Nov, 11:59 PM

Resources

- You can use tutorial code or any online resources
  - Provide proper citations in your report (even for ChatGPT or GitHub-copilot)
    - https://apastyle.apa.org/blog/how-to-cite-chatgpt
  - Explanations should be in your own words

- Consultation Hours:
  - Wednesday 5:00 pm – 6:00 pm at COM3-02-31 or after the lecture/tutorial
  - You can also consult with Simian after the tutorials

NUS
National University
of Singapore

Thank You