FIT5212 Assignment 2

Due 11:55 PM, Wednesday, June 10, 2020

In this assignment, you are requested to finish 2 tasks. This assignment accounts for 25% of the total marks for this unit.

Task 1: Recommender System Challenge (70% Marks)

Description

You are required to complete an **in-class challenge in Kaggle** **https://www.kaggle.com/t/6d26bd75418742f5b2d2c605840e4637** .

This dataset is collected from an online social network platform. People interact with others by making friends and posting reviews and ratings for different items. Please recommend a list of items to each user.

Data

The user item interaction data is the main data for this challenge. This data is further split into training, validation, and test sets.

- Training data. The training dataset contains a set of interactions between users and items. If a user engages with an item, then there will be a record in the dataset.
- Test data. Each user is provided with a list with 100 candidates in the test dataset, you will need to check the candidate list and recommend the top 10 items for each user.
- Validation data. Similar to the Test data, the difference is that the last column in the Validation dataset is the ground truth of the ratings. You can use this dataset to tune your model.

Please train your recommender systems and generate the outputs for the test data.

Optional Data

In additional to the interaction data (training data), you are also provided with the following information:

1. User information.
2. Item information.
3. Social information.

Please note that the user, item, social information are optional data for this challenge. You don't have to use it. It is your own choice to determine if the information should be used or not.

More information can be found here
https://www.kaggle.com/t/6d26bd75418742f5b2d2c605840e4637

.

Requirements:

1. Participate in the challenge and make your submission. Maximum submission in Kaggle is 2 submissions per day.
2. This is an individual assignment. You have to finish it on your own.
3. In addition to the challenge, you have to finish a report on this challenge and submit it to Moodle.

Submission:

To Kaggle

- Kaggle submission, you need to submit your result on Kaggle.

To Moodle:

1. A csv file, "**studentID.csv**". Please replace studentID as your own student ID. The content should be the same as the file you have submitted to Kaggle. This file should be submitted in Moodle. We will double check the files you have submitted to Kaggle and Moodle. If the two files are not the same (i.e., the file submitted to Moodle cannot get the same score in Kaggle), your result is invalid, and you will fail the assignment.
2. A jupyter notebook, "**code_studentID.ipynb**". This notebook should show how you finish the task. Ideally you should show what sort of algorithms you have considered, what kind of information you have used, and the reason for your choice of the corresponding algorithm to achieve the results you submitted to Kaggle. **Comparison for different algorithms should be included in this jupyter notebook.** And detailed analysis of the results are encouraged. The notebook should be self contained. If you have used other algorithms/packages which are not covered in this lecture, you should give a detailed introduction to that algorithm/package. If a third party package is used, this package should be a well-known package and easy to install (e.g., install within a single command).
   This nodebook should include both markdown explanation, codes, and outputs, so that we can read and mark.
3. A pdf file, "**code_stduentID.pdf**". This pdf is generated by cleaning all the output in the jupyter notebook and exporting as a pdf file. This pdf will be passed in Turnitin for plagiarism check.
4. A pdf report, "**report_stduentID.pdf**". This pdf contains more detailed analysis of the work. This pdf will be passed in Turnitin for plagiarism check.

Marking:

- The kaggle leaderboard only shows your scores on 50% of the test data. Your final score will be marked based on your csv file submitted to Moodle for the whole test dataset.
- The methodology and report for Task 1 is set to 30% of the total mark for this assignment, and the prediction score accounts for 40% . So please prepare a good report and clearly describe your method to achieve the task.

Task 2: Node Classification in Graphs (30% Marks)

Description

You are given a graph, and you are required to perform the node classification in this graph dataset.

Dataset Description

You are given a citation network. In this network, each node is paper, an edge indicates the relationship between two papers. As the network has extremely sparse network structure, we also provide text information for each paper, i.e., the title of each paper. The files in the dataset include:

File Name	Description
docs.txt	title information of each node in a network, each line represents a node (paper). The first item in each line is the node ID
adjedges.txt	neighbor nodes of each node in a network. The first item in each line is the node ID, and the rest items are nodes that have a link to the first node. Node that if only one item in a line, it means that the node has no links to other nodes
labels.txt	class labels of a node. Each line represents a node id and its class label

The task is to perform the node classification for the papers presented in the labels.txt (The first column is the node ID).

Node Classification (30% Marks)

For node classification, you are asked to classify the nodes in the network into several categories, and evaluate the performance of different classification algorithms. Please split the network as training and test set yourself. The training percentage is 20%. As the network contains different information, including node content and graph structure information, you should make necessary comparisons and recommend a good algorithm for this task. At least one embedding approach (text embedding or network embedding) should be used. You should justify the use of different graph information as well as the recommended algorithm for this task. Detailed result analysis is important to this task.

Please set the random seeds for reproducible results. This can be done with:

```
import numpy as np
np.random.seed(0)
import torch
torch.manual_seed(0)
```

See more from here https://pytorch.org/docs/stable/notes/randomness.html.

The node classification performance should be evaluated in terms of accuracy (the percentage of correctly classified samples).

Submission:
The codes should be finished in the jupyter notebook. Add a new section for Task 2 in you **code_studentID.ipynb,** and make a clear explanation in “**report_stduentID.pdf**”.