2021 FIT5212 Assignment 2

Due 11:55 PM, Tuesday, May 25, 2021

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 on Kaggle**

https://www.kaggle.com/t/6060c830a81d32ea0fd417d9826d3936 .

There is rich semi-structured data on social media platforms. This dataset for this task is collected from an online photo-sharing social media platform **Flickr**. People share photos with others on Flickr and make connections (friends). The task is to recommend a list of items (photos) to each user given rich information on this social media platform.

Rules

One account per participant.

- You cannot sign up on Kaggle using multiple accounts and therefore you cannot submit from multiple accounts.
- You may be disqualified for this competition if you use multiple accounts. Why? Please read this news about the ImageNet competition among big companies https://www.bbc.com/news/technology-33005728 .
- You can submit 5 submissions on Kaggle per day.

No private sharing among students

- Privately sharing code or data is not permitted. Violation of this can lead to serious academic punishments.

Submission:

To Kaggle

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

To Moodle:

1. A csv file, “**studentID_output.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, “**studentID_code.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 notebook should include both markdown explanation, codes, and outputs, so that we can read and mark.

3. A pdf file, “**stduentID_code.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, “**stduentID_report.pdf**”. This pdf should be **no more than 5 pages** which contain detailed analysis of the work. Detailed analysis and comparison for different algorithms should be included in the report. 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 complete the task. **Please include your kaggle name for this competition in your report.**

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

Description

You are given a graph, and you are required to perform the node clustering 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 clustering for the papers presented in the labels.txt (The first column is the node ID).

Node Clustering (30% Marks)

For node clustering, you are asked to cluster the nodes in the network into several categories, and evaluate the performance of different clustering algorithms. 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 clustering performance should be evaluated in terms of **normalized mutual information (NMI)** (https://scikit-learn.org/stable/modules/generated/sklearn.metrics.normalized_mutual_info_score.html). As running clustering algorithms such as k-means may result in different NMI scores with different seed values, it is recommended that you run each clustering algorithm multiple times (for instance 10 times) with different random seeds and then report the average performance.

Submission:

The codes should be finished in the jupyter notebook. Add a new section for Task 2 in your **studentID_code.ipynb,** and make a clear explanation in “**stduentID_report.pdf**” (**no more than 2 pages**).