Final Project

This final project consists of two parts and is worth in total 36% of your grade and 36 total marks.

The first part is an individual assignment to complete a Kaggle competition. Each student will build a **recommender system model** to make predictions related to music reviews on *Amazon*. You will also be required to complete a write up describing your methods.

The second part is an open ended **research project**. You may augment your work from Part 1, or use what you learned to select a new dataset and task to investigate. You are free to explore something related to your research, but the project should be a distinct and isolated component – you must state which parts of the project were completed during the project timeframe. You may work in teams of up to 3 people. You will be required to complete a project report on your work in the form of an academic paper, as well as a final presentation.

Part 1 will be worth 10 marks, with the competition itself worth 5 marks and the writeup worth 5 marks. Part 2 will be worth 26 marks, with the research report worth 10 marks and the final presentation worth 15 marks. 1 mark will be awarded for registering your team on time. Details on how these marks will be graded are provided below.

Deadline: Dec. 9, 2020 at 11:59pm EST for Part 1. Dec. 10, 2020 for the presentation for Part 2. Dec. 15, 2020 at 11:59pm EST for the Part 2 written report.

Submission: You will need to submit one writeup for each part of this final project through Markus[1]. Kaggle submissions will be made through the provided links. You will also be required to provide the code used for both parts.

Late Submission: No late submissions will be accepted.

Computing: To install Python and required libraries, see the instructions on the course website[2]

Collaboration: Part 1 of this project is to be completed individually. Part 2 can be completed in a group, and we encourage groups of up to three. The process for team registration is detailed in Part 2.1.

[1] https://markus.teach.cs.toronto.edu/csc2515-2020-09

[2] https://www.cs.toronto.edu/~huang/courses/csc2515_2020f/index.html

Part 1. Building a Recommender System (10 marks)

1.1 Kaggle Competition (5 marks)

In this part of the assignment, you will build a **recommender system model** to predict ratings related to music reviews on *Amazon*. Specifically, given a (user, item) pair and associated review data, we want to predict the review's star rating as accurately as possible. Performance will be measured with MSE.

Solutions will be graded on Kaggle, with the competition closing at **11:59pm EST, Wednesday December 9** (note that the time reported on the competition webpage is in UTC!). The leaderboard will show your results on half of the test data, but your ultimate score will depend on your predictions across the whole dataset. This assignment should be completed **individually**. You must include your Kaggle team ID (click "download raw data" at the public leaderboard) in your submitted report, and the name on your Kaggle team (show on the leaderboards) must match your name on Markus. The link to the Kaggle competition will be posted on Quercus.

Files

train.json.zip 200,000 review to be used for training. It is not necessary to use *all* ratings for training, for example if doing so proves too computationally intensive.

reviewerID The ID of the user. This is a hashed user identifier from Amazon.

itemID The ID of the item. This is a hashed product identifier from Amazon.

reviewText The text of the review.

summary A short summary of the review.

overall The star rating of the user's review from 1 to 5.

price Price of the item.

reviewHash Hash of the review (essentially a unique identifier for the review).

unixReviewTime Time of the review in seconds since 1970.

reviewTime Plain-text representation of the review time.

category Category labels of the product being reviewed.

test.json.zip 10,000 reviews to be used for generating the final Kaggle submission. All fields are the same as in **train.json.zip** with the exception of the overall rating removed.

rating_pairs.csv Pairs (reviewerIDs and itemIDs) on which you are to predict ratings.

baselines.py A simple baseline that computes a user average and global average on training data, then uses this to predict on test data. This code is given to demonstrate how to properly format predictions for uploading to Kaggle. A submission made with this code corresponds to the 'naive baseline' submission on the leaderboard.

Please do not try to collect these reviews from Amazon, or to reverse-engineer the hashing function we used to anonymize the data. Doing so will not be easier than successfully completing the assignment. **We will require working code for all submissions to ensure no violation of the competition rules**.

Grading and Evaluation

Performing well on the task is worth 5 marks. Your Kaggle performance will be graded as follows:

- Your ability to obtain a solution which outperforms the leaderboard baselines on the unseen portion of the test data (4 marks). Obtaining full marks requires a solution which is substantially better than baseline performance.
- Obtain a solution which outperforms the baselines on the seen portion of the test data (i.e., the leaderboard). This is a consolation prize in case you overfit to the leaderboard. (1 mark).
- Students with submissions ranked in the top 10 will receive a single bonus mark.

To obtain good performance, you should not need to invent new approaches (though you are more than welcome to!) but rather you will be graded based on your ability to apply reasonable approaches to each of the given tasks. You will submit a zip file containing the code used to produce your submission to Markus. We will be checking submissions for similar or copied code and to verify competition rules were followed.

1.2 Written Report (5 marks)

You will also write a brief report about the approaches you took. Your report should be 12 pt font and be between 2 and 4 pages excluding references. This report will be submitted on Markus and is due by **11:59pm, Wednesday December 9**. **Remember to include your Kaggle team ID in this report or we will not be able to grade your submission.** Your report should cover the following sections:

1. Describe how you processed your data and what features you used. Your exploratory analysis here should motivate the model you use in the next section.
2. Describe your model. Explain and justify your decision to use the model you proposed. How will you optimize it? Did you run into any issues due to scalability, overfitting, etc.? What other models did you consider for comparison? What were your unsuccessful attempts along the way? What are the strengths and weaknesses of the different models being compared?
3. Describe your results and conclusions. How well does your model perform compared to alternatives, and what is the significance of the results? Which feature representations worked well and which do not? What is the interpretation of your model's parameters? Why did the proposed model succeed why others failed (or if it failed, why did it fail)?

Part 2. Open Ended Research Project (26 marks)

This is an **open-ended** project in which you are expected to write a detailed report documenting your results. This assignment may be conducted in groups of 1-3 people. The marking scheme is the same regardless of your group's size. Examples of datasets and projects that may be of interest in this assignment will be discussed in the lectures, though you may use any dataset you wish (including the ones we used for Part 1). For a selection of datasets, `https://cseweb.ucsd.edu/~jmcauley/datasets.html` is a good resource.

You are also welcome to to use a research project from your own graduate studies as your project, as long as you detail specifically what work was done during the course of this class and cover the items above in your research report as appropriate.

2.1 Registering your Team (1 mark)

In order to register your team, you will need to submit a pdf with your team member's names and student IDs on Markus. Make sure your submission on Markus is a group submission that contains all your team members. This assignment must be submitted by **November 27** so that we will have enough time to schedule the final presentations. If you are unable to find a team we have set up a Piazza group finder post to help.

2.2 Project Report (10 marks)

Your team will be required to produce a written report detailing your research. Make sure to specify the names of all of your group members when submitting. Your reports will follow the NeurIPS template and should be about four pages long, excluding references. Reports are due on Markus by **11:59pm EST, Tuesday December 15**. Make sure you include your teammates as team members of your submission.

Your reports will be graded on their coverage of the following five components. Examples of what might be included in these sections and previous assignment examples shall be described in more detail in class. Each of the five sections below will be worth 2 marks each for a total of 10 marks. **This grade will be shared by the whole group.**

1. Identify a dataset to study, and describe literature related to the problem you are studying. If you are using an existing dataset, where did it come from and how was it used? What other similar datasets have been studied in the past and how? What are the state-of-the-art methods currently employed to study this type of data? Are the conclusions from existing work similar to or different from your own findings?

2. Perform an exploratory analysis of the data. Describe the dataset, including its basic statistics and properties, and report any interesting findings. This exploratory analysis should motivate the design of your model in the following sections. Datasets should be reasonably large (e.g. more than 50,000 samples).

3. Identify the predictive task you will be studying on this dataset. Describe how you will evaluate your model at this predictive task, what relevant baselines can be used for comparison, and how you will assess the validity of your model's predictions. It's fine to use models that were described in class here (i.e., you don't have to invent anything new (though you may!)),

though you should explain and justify which model was appropriate for the task. It's also important in this section to carefully describe what features you will use and how you had to process the data to obtain them.

4. Describe your model. Explain and justify your decision to use the model you proposed. How will you optimize it? Did you run into any issues due to scalability, overfitting, etc.? What other models did you consider for comparison? What were your unsuccessful attempts along the way? What are the strengths and weaknesses of the different models being compared?

5. Describe your results and conclusions. How well does your model perform compared to alternatives, and what is the significance of the results? Which feature representations worked well and which do not? What is the interpretation of your model's parameters? Why did the proposed model succeed why others failed (or if it failed, why did it fail)? What would be some next steps for the project if you had more time?

Finally, your report should also include a link to a public GitHub repository containing the code used to produce your results. If you would like to not make your code public for any reason, you must instead submit your code as a zip file on Markus.

2.3 In-class Project Presentation (15 marks)

You will also be required to present your work in class. You will be graded as a group on how well you cover all aspects of your project and answer questions, as well as individually on your presentation skills. Presentations will be scheduled for **December 10**.

10 marks will be awarded as a group on how well your presentation covers the 5 components described above (related work, data, predictive task, model, and results). The other 5 marks will be awarded based on your individual ability to contribute to the presentation, discuss your work clearly and succinctly, and answer questions.