COMP9727: Recommender Systems
Project: Project Implementation and Evaluation

Due Date (Peer Assessment): Week 10, Wednesday, July 31, 4:00 p.m.
Due Date (Implementation): Week 10, Sunday, August 4, 5:00 p.m.
Due Date (Evaluation): Week 11, Friday, August 9, 5:00 p.m.

Value: 50%

The project is the major assessment item for this course, and is divided into a team-based project *implementation* in which students work in teams of 3–4 to build and evaluate a recommender system, and an individual *report* in which students expand on the team evaluation by reflecting on the whole process and proposing future directions for the work in the light of the material presented in the lectures in the second half of the term.

The project is expected to follow (and build on) an initial project design, and as such should focus on **all** aspects of building and evaluating a recommender system – the scope, dataset(s), method(s) and evaluation in the design (metrics and perhaps a small user study). Consider also the justification for your final design choices, such as why one particular style of recommender system is most suitable for the chosen scenario, or why one dataset is more useful than another.

Note that standard UNSW late penalties apply. In particular, there can be no late submissions for the project implementation as this is a team project.

Implementation

The project implementation should follow the style of the assignment and include the content outlined in the project design (possibly refined). Your team should include one main Jupyter notebook that shows your reasoning process through the construction and evaluation of your recommender system, including intermediate evaluations that went into the decision of the final recommender system proposal. It may be convenient to place auxiliary software modules in separate files, which can be submitted along with the main notebook (see below).

The mark for the project implementation is determined from a presentation held in Week 10, and is based on the sophistication of the recommender system, the thoroughness of the evaluation, the interpretation of the results, and the extent to which the project achieved its objectives.

Each member of a team generally receives the same mark for the project implementation, however marks are moderated according to individual contributions determined using peer assessments: in "dysfunctional teams" where member contributions are highly unequal, marks will be adjusted to reflect the level of contribution. The Moodle Team Evaluation tool will be used for each team member to assess (anonymously) the relative contributions of the team members. Students are required to enter their peer assessments using the tool by 4:00 p.m. the day of the first project presentations. Once the due date has passed, these percentages cannot be modified.

Presentation

Each team will present a 20-minute summary of their project and findings at a time to be arranged in Week 10. The structure of the presentation should follow that of the common part of the report (see below) and cover the project scope, dataset(s), method(s) developed, and evaluations of the methods and proposed recommender system using historical data and/or any user evaluation, including the rationale for any design decisions. It is recognized that there may not be enough

time for every detail to be included in this presentation, however additional information can be included in the individual report. Note that in constrast to the project pitch, this presentation is **not** meant to be a "sales pitch": your audience will play the role of your managers, who are interested primarily in results and outcomes.

As these presentations form part of the assessment for the project, only the project team and markers will be present. Your presentation may take the form of a Jupyter notebook (perhaps shortened version of the main one(s) submitted for the project), or a PowerPoint or pdf presentation. Include a copy of your team's presentation with your team project submission (see below for detailed instructions).

Report

The report is prepared individually, though should include elements common to each team member for the scope of the recommender system, dataset(s), methods and evaluation. The report should be a `.pdf` document of around 20 pages, though this will depend on the size and number of plots and tables.

For the **common** part, start with "due diligence" by including a "competitor analysis", by which is meant a brief description of any commercial systems that already exist for your team's recommendation scenario, discussion of their strengths and weaknesses, and comparison with your proposed system. Then summarize the general approach to the recommendation problem that you have found most successful, and include sufficient description and evaluation of methods to justify your results, making use of suitable dataset(s) and experiments.

For the **individual** part, add a reflection on the process of developing this project: what went well, what did not work (and why), and what would you do differently next time. Then include a section on how the recommender system could be extended to make use of the techniques described in the second half of term, i.e. social network recommendation, sequential/temporal recommendation, context-aware recommendation and/or LLMs, outlining what data would be needed to achieve this, and how realistic this is as a possibility. You may discuss the questions in general terms with other students, however the individual sections of your report must be written independently, and plagiarism detection software will be used to check that there has been no collusion on the writing.

Submission

There are **three** separate items to submit, one as a team and two individual. **Please include your name and zid at the start of all documents.**

1. **Team Member Contributions.** Submit your **individual** peer assessment on Moodle by Wednesday, July 31, 4:00 p.m. Enter your best estimate of each team member's contribution to the implementation and evaluation of the recommender system, as the percentages cannot be adjusted after the submission date.

2. **Project**. Submit one `.zip` file **per team** containing (i) your software, including a main Jupyter notebook and all your software modules, though excluding Python and other libraries, and (ii) your presentation as a Jupyter notebook, PowerPoint presentation or pdf, on the CSE system by Sunday, August 4, 5:00 p.m., choosing an informative name such as `hotel-recommender.zip`, using the command:

   ```
   give cs9727 project <name>.zip
   ```

   The student submitting can check that the submission has been received using the command:

   ```
   9727 classrun -check project
   ```

3. **Project Evaluation.** Submit one `.pdf` file for your **individual** report on Moodle by Friday, August 9, 5:00 p.m.

Structure of the Presentation and Report

The following structure is suggested. Points need only be addressed if relevant.

1. **Problem.** Clearly describe the recommendation problem(s) and how the recommender system would be deployed within a larger (web or mobile) system. Include a basic "compeitor analysis" describing some current systems in use in this domain (if any), and their limitations. For your proposed recommender, what user inputs are available, what recommendations would be provided, and when? What user feedback (if any) would be obtained, and how would that be used by the recommender? How is the recommendation problem defined as a classification/prediction, rating/estimation or ranking problem? User interface mockups could be useful to illustrate the intended usage of the system.

2. **Dataset(s).** What dataset(s) did you use for the problem scenario(s)? Describe the basic characteristics of the dataset(s) and (if relevant) how you linked different datasets together. Give exploratory data analysis along the lines of the tutorials and assignment. What are the strengths and weaknesses of the dataset(s) in terms of addressing the chosen recommendation problem(s)? If you have used smaller subset(s) of available dataset(s), explain and justfify the reasons for this and the choice of each subset.

3. **Method(s).** Discuss the overall approach to the recommendation problem, and provide a rationale for choosing the set of methods for evaluation. Be clear about the type of method, whether content-based, collaborative filtering or hybrid. Explain all methods clearly giving values for all parameters, e.g. the value of k if using kNN, the similarity function, regularization parameter (even if this is the default value), neural network architectures, the number of recommendations produced, any preprocessing, the implementation or library used, and if using a method not discussed in class, a reference to the algorithm source (paper, library and/or code) used to implement that method. The objective is to describe your methods in such a way that they could be replicated, so do not leave out any necessary details.

4. **Experiments.** Clearly describe the experimental setup and the metrics chosen to evaluate performance of the recommender. For the experimental setup, explain the methodology, e.g. 80/20 training/test split or cross-validation (and how many folds) or for sequential prediction, whether the test set includes users not in the training set or *only* users not in the training set, or if you have tested multiple scenarios, whether the temporal ordering is important in the dataset and why, etc. Make sure that each method is evaluated in exactly the same way so the comparison between methods is fair and consistent. For metrics, there should be a variety of metrics for both the recommendation model (that focuses on historical data) and (if relevant) the recommender system (that focuses on user feedback and/or interactions). It may be appropriate to use top-N metrics (for some N) and/or per-user metrics, i.e. metrics averaged over a set of users. Give and justify the values of N used for top-N recommendation.

5. **Evaluation.** Discuss the results of the experiments with reference to the metrics for assessing the performance of both the recommendation model and resulting recommender system, taking into account how and when recommendations will be presented to users. If there are multiple metrics, say which are the most important or discuss the tradeoffs between the different metrics and identify your "best" model(s) for the recommendation problem(s). Also consider the computational requirements of the recommender system and especially any requirement for recommendations to be computed in near-real time, or for the model to be dynamically updated/retrained.

6. **Reflection.** Discuss what went well and what did not work well in the project. What would you do differently next time? In the light of the results obtained, would the proposed recommender system be commercially viable, or would more work be required for deploy this system? Considering the topics in the latter part of term (social network recommendation, sequential recommendation, context-aware recommendation and Large Language Models, which of these approaches could be useful extensions to the recommender system you have developed? Discuss the potential benefits and tradeoffs.

Assessment

Implementation

	<50	50–64	65–74	75–84	85–100
Problem (25%)	Problem not clearly defined.	Problem scope outlined in broad terms.	Problem scope defined precisely.	Problem scope defined precisely and extensively.	Novel problem defined precisely and completely.
Techniques (25%)	Methods not applied correctly.	Suitable basic method used.	Combination of basic methods used.	Complex method developed and implemented.	Novel method developed and implemented.
Experiments (25%)	Experiments minimal in scope.	Experiments adequate to justify conclusions.	Experiments provide some evidence for conclusions.	Experiments provide convincing evidence for conclusions.	Novel experimentation provides evidence for conclusions.
Analysis (25%)	Experimental analysis is incorrect.	Experiments sound and results correctly summarized.	Conclusions reached using sound reasoning and evidence.	Extensive experimental evidence fully supports conclusions.	Novel interpretations of results supported by evidence.

Evaluation

	<50	50–64	65–74	75–84	85–100
Content (25%)	Lacks content or conclusions poorly articulated.	Some conclusions drawn but analysis lacks depth.	Appropriate experiments used to support conclusions.	Experiments conducted according to sound methodology.	Novel insights made concerning experiments and results.
Structure (25%)	Poorly organized with little logical flow.	Adequately structured but connections between ideas not always clear.	Logically organized, connections between evidence and conclusions.	Logically organized, clearly drawn connections between evidence and conclusions.	Logically organized into a coherent, easy to understand argument.
Analysis (25%)	Experimental analysis is incorrect.	Experiments sound and results correctly summarized.	Conclusions reached using sound reasoning and evidence.	Extensive experimental evidence fully supports conclusions.	Novel interpretations of results supported by evidence.
Presentation (25%)	Too short or too long.	Required form and length but little original content.	Conclusions clearly articulated.	Clear explanations of arguments, evidence and conclusions.	Engaging style with clear explanations throughout.

Plagiarism

Remember that ALL work submitted for this project must be your own work (your own team's work) and no sharing or copying of code or answers between teams is allowed. The individual components of the report must be entirely your own work; you may discuss You may use code or datasets from the Internet only with suitable attribution of the source. You may not use ChatGPT or any similar software to generate any part of your explanations, evaluations or code. Do not use public code repositories on sites such as github or file sharing sites such as Google Drive to save any part of your work – make sure your code repository or cloud storage is private and do not share any links. This also applies after you have finished the course, as we do not want next year's students accessing your solution, and plagiarism penalties can still apply after the course has finished.

All submitted evaluation reports will be run through plagiarism detection software to detect similarities to other submissions, including from past years. You should **carefully** read the UNSW policy on academic integrity and plagiarism (linked from the course web page), noting, in particular, that *collusion* (working together on an assignment, or sharing parts of assignment solutions) is a form of plagiarism.

Finally, do not use any contract cheating "academies" or online "tutoring" services. This counts as serious misconduct with heavy penalties up to automatic failure of the course with 0 marks, and expulsion from the university for repeat offenders.

One more thing. There is a type of plagiarism known as "self plagiarism", which is also disallowed. All work submitted for this assignment must be original and no part of the work can be submitted in future, or have been submitted previously, for assessment in another course.