School of Computing and Information Systems
The University of Melbourne
COMP90049, Introduction to Machine Learning, Semester 1, 2024

Assignment 2: How good is that movie?

Release:	Monday 15 April 2024
Due:	*Stage I*: Friday, 10 May 2024 at 5 PM *Stage II*: Friday, 17 May 2024 at 5 PM
Marks:	The Project will be marked out of 30 and will contribute 30% of your total mark.
Main Contact:	Hasti Samadi (hasti.samadi@unimelb.edu.au)

1 Overview

The goal of this project is to build and critically analyse Machine Learning methods, to predict the rating of some movies extracted from the TMDB database.

This assignment offers an opportunity to delve into machine learning concepts within a research context and enhance skills in data analysis and problem-solving.

The objective is to critically evaluate and analyse the efficacy of various machine learning algorithms for predicting movie ratings and to communicate your findings in an academic report. The technical aspect involves implementing machine learning algorithms to solve the task, while the report focuses on interpreting observations and drawing meaningful conclusions.

Your report should showcase your understanding of the subject matter in a manner accessible to a reasonably informed reader.

2 Deliverables

Stage I: *Model development and report writing (due May 10th):*

1. **Report (25 marks)**: An anonymous written report, of 2200 ($\pm$10%) words including a reference list, figure captions and tables. We are using the Canvas word counter as our word count reference. Your name and student ID should not appear anywhere in the report, including the metadata (filename, etc.). *You should submit your report as a **single PDF file TWICE.***
    a. *Once through Canvas/Turnitin (A2: Open-ended Research Report)*
    AND
    b. *Once through Canvas/Feedback Fruits (A2: Peer Review and Self-Reflection)*
2. **Output (2 marks)**: Rating predictions for the test instance dataset. *Submitted as a single CSV file through Kaggle in-class competition.* (Described in section 7).
3. **Code**: One or more programs, written in Python, including all the code necessary to reproduce the results in your report (model implementation, label prediction, and evaluation). Your code should be executable and have enough comments to make it understandable. You should also include a README file that briefly details your implementation. Please note that if you do not submit your code or your code is not functional, we will not mark your report. *Submitted as a zip file through*

Canvas (A2: Open-ended project code and comments)

Stage II: *Peer reviews (due May 17th):*

1. **Peer review:** reviews of two reports written by other students of 210-360 words each (Described in section 5)
2. **Reflection**: a written reflection piece of 400 words. (Described in section 5)

Both should be submitted via Canvas/Feedback Fruits - A2: Peer Review and Self Reflection

NOTE 1: Do **NOT** upload your report as part of a compressed archive (zip, tar, . . .) file or in a different format.

NOTE 2: Stage I submissions will be open from May 5. Stage II submissions will be open as soon as the reports are available (24 hours following the Stage I submission deadline).

3 Data

The information on movies is collected from the TMDB website, which is a platform that allows users to search its database of movies, rate them and write reviews. The data files for this project are available via Canvas and are described in a corresponding README file.

In our datasets, each movie contains:

- *Movie features*: such as `release_year`, `runtime`, `budget`, `revenue`, `original_language` and more.
- *Text features*: There are four text features in this dataset: `title`, `overview`, `tagline` and `production_companies`. We have used various text encoding methods to encode these features for you.

You are provided with:

- Labelled datasets: (Include movies' ratings as explained in section 3.1).
  - TMDB_train.csv: Consists of movie features of 100,000 movies with their rating labels that you can use to train your supervised Machine Learning models.
  - TMDB_evaluate.csv: Consists of movie features of 20,000 movies with their rating labels that you can use to evaluate your supervised Machine Learning models.
- Unlabelled datasets:
  - TMDB_unlabelled.csv: Consists of details of 254,701 movies that you can use to train your unsupervised or semi-supervised Machine Learning models.
  - TMDB_test.csv: Consists of details of 20,000 movies that you should use to TEST the performance of your Machine Learning models and report the result onto the Kaggle page.
- Pre-processed datasets:
  - TMDB_text_features_*.zip: The pre-processed text features for training and test sets, one zipped file for each text encoding method. Details about using these text features are provided in the README file.

3.1 Target Labels

These are the labels that your model should predict (y). We provide this label in two forms:

- the average rating (float; in the column named `average_rate` in the Train and Evaluate CSV files); and
- a categorical label indicating the rating band, where we binned the rating of the movies into 6 categories as follows (integer; in the column named `rate_category` in the Train and Evaluate CSV files).:
    - *average_rate < 4* → *0*
    - *4 <= average_rate < 5* → *1*
    - *5 <= average_rate < 6* → *2*
    - *6 <= average_rate < 7* → *3*
    - *7 <= average_rate < 8* → *4*
    - *average_rate >= 8* → *5*

You may use either of these label representations in your experiments, but different representations might call for different machine-learning approaches.

3.2 Features

To aid in your initial experiments, we have created different feature representations from the given datasets. You may use any subset of the text feature representations described below in your experiments, and you may also engineer your own features from the raw descriptions if you wish.

The provided representations are:

1. **BoW (Bag of Words)**

   We applied the *CountVectorizer* to transform the instances into vectors of *Token_ID* and their *count*. For example, with the use of *CountVectorizer* the title "Sudani from Nigeria", will be transformed into the following vector:

   [(39785, 1), (28688, 1)]

   Where 39785 is the token_ID for the word "Sudani" and 28688 is the Token_ID for the word "Nigeri". In this example and the provided files, we (1) removed all `stopwords[1]` and (2) only retained the 1000 words in the full data set with the highest COUNT values.

   . There are many other modifications you can use to experiment with different hypotheses you may have. For example, how 'removing very frequent and/or very infrequent words' can affect the behaviour of your Machine Learning models. There are many more examples.

2. **TFIDF**

   We applied term frequency-inverse document frequency pre-processing (*TfidfVectorizer*) to transform the text features as a vector of values that measure their importance using the following formula:

   $$w_{d,t} = f_{d,t} \times \log \frac{N}{f_t}$$

   Where $f_{d,t}$ is the frequency of term t in document d, f_t is the number of documents containing t, and N is the total number of documents in the collection. You can learn more about TFIDF in (Qaiser, Ali 2018).
   Using TFIDF the above example title will be transformed into the following vector:

[1] Stopwords are common words like "the," "and," "is," etc., which are filtered out from text data because they carry little meaningful information for natural language processing tasks.

[(39785, 0.707), (28688, 0.707)]

Similar to the Bag of Words method, you can use and edit this basic method to experiment with your ideas. In our provided dataset, we (1) removed all `stopwords` and (2) only retained the 1000 words in the full data set with the highest TFIDF values.

There are many other text vectorization methods that you can use (e.g. word2vec, Bert, etc.). You are welcome and encouraged to use as many vectorization methods as you choose. But please keep in mind that we are more interested in the depth of analysis and quality of interpretation in your report, NOT the variety or complexity of the methods you have used.

4 Stage I

4.1 Task Basics

You'll use the TMDB_train.csv dataset and, if you haven't pre-processed the data yourself, you can use the pre-processed files (TMDB_text_features_*.zip) to train machine learning models. Then, you'll evaluate these models using TMDB_evaluate.csv.

Depending on your research question, you might also use TMDB_unlabelled.csv to improve your models.

Once your models are ready, you'll use them to predict the 'rate_category' for all the movies in TMDB_test.csv. You'll submit these predictions to our Kaggle competition page.

4.2 Research Question

You should formulate a research question and develop machine learning algorithms and appropriate evaluation metrics to address the research question. Here are some sample research questions:

- Are big-budget films more popular than their low-budget counterparts?
- Is there any relationship between production companies and the rating of their movies?
- Is the rating of a movie correlated with its genre?
- Is the rating of a movie correlated with its title?
- Does the use of unlabelled data improve the performance of machine learning models in this dataset?
- How does the use of text features impact the performance of machine learning models in this dataset?
- Does using the ‘overview’ assist with the identification of very high-rated movies?

There are many more possible questions. You can choose to use any as your research question.

4.3 Feature Engineering

The process of engineering or selecting features that are useful for discriminating among your target class set is inherently poorly defined. Most machine learning assumes that the attributes are simply given, with no indication from where they came. The question as to which features are the best ones to use is ultimately an empirical one: just use the set that allows you to correctly classify the data.

In practice, the researcher uses their knowledge about the problem to select and construct “good” features. What aspects of a movie’s details might indicate its rating? You can find ideas in published papers, e.g., (Saraee, White et al. 2004).

It is optional for you to use the features provided (as they are), generate a new set of features or select a substitute of the features. Whatever method you choose, you have to use the features to train some models and run a few experiments on the given evaluation data.

4.4 Analysing Machine Learning Models

Various machine learning techniques have been (or will be) discussed in this subject (0R, 1R, Naive Bayes, Decision Trees, k-NN, Logistic Regression, Neural Networks, etc.); many more exist. You may use any machine learning method you consider suitable for this problem. *You are strongly encouraged to make use of machine learning software and/or existing libraries (such as sklearn) in your attempts at this project.*

In this stage, your task has two phases:

- *The training-evaluation phase:* The holdout approach should be applied to the training data provided. Check section 4.6 for the minimal expectations in this phase.
- *The test phase:* the trained classifiers will be evaluated on the provided test data. The predicted labels of test cases should be submitted as part of the Stage I deliverable on Kaggle. Check section 7 for details.

Based on your research question and after training different models and running a few experiments, you are expected to develop some knowledge of why you are reaching the results you do and some hypotheses of how you can change these results. Here are a few examples:

Example 1

> Hypothesis: *removing the 'release_year' from the features can reduce the noise and increase the performance of x and y models*.
>
> Test: *compare the performance of x and y models before and after 'release_year' removal and comment on the observation. Did the observation support the hypothesis? Why or why not?*

Example 2

> Hypothesis: The *machine learning model A is working faster than model B has something to do with the structure of the instances in this dataset*.
>
> Test: *Change the structure of the instances (by sub-sampling or feature engineering or any other way) and test the performance of the two models before and after the changes. Did the observation support the hypothesis? Why or why not?*

You should then test these hypotheses with more experiments. When explaining your results, you are expected to use examples from the dataset as well as theories and findings from the lectures and published literature. You are also expected to use appropriate visualization tools (e.g., tables or diagrams) to communicate your findings professionally and academically.

4.5 Report

Your main submission for this assignment is your report. The report should follow the structure of a short research paper, as will be discussed in the guest lecture on Academic Writing. It should describe your approach and observations, both in engineering features, and the machine learning algorithms you tried. Its main aim is to provide the reader with knowledge about the problem, in particular critical analysis of your results and discoveries. The internal structure of well-known classifiers (discussed in the subject) should only be mentioned if it is important for connecting the theory to your practical observations.

The following is the expected structure of the report for this assignment.

- *Introduction*: a short description of the problem, data set and research question. Your report should clearly state your research question. Remember addressing more than one research question does not necessarily lead to higher marks. We value the depth and quality of your critical analysis of methods and results over simply covering more content or materials.
- *Literature review:* a short summary of some related literature, including the data set reference and at least two additional relevant research papers of your choice.
- *Method:* Introduce the used feature(s), and the rationale behind including them. Explain and justify the Machine Learning models you have used and their hyperparameters. You also need to explain your evaluation method(s) and metric(s) you have used (and why you have used them). *This should be at a conceptual level; a detailed description of the code is not appropriate for this report. The description should be similar to what you would see in a machine learning conference paper.*
- *Results*: Present the results, in terms of evaluation metric(s) and, ideally, illustrative examples and diagrams.
- *Discussion / Critical Analysis*: Contextualise the systems' behaviour, based on the understanding of the subject materials (*This is the most important part of the task in this assignment*).

  Contextualise implies that we are more interested in seeing evidence that you have thought about the task and determining reasons for the relative performance of different methods, rather than the raw scores of the different methods you selected. This is not to say that you should ignore the relative performance of different runs over the data, but rather that you should think beyond simple numbers to the reasons that underlie them and connect them back to your research question. You can also add complementary experiments and their results in this section.
- *Conclusion*: Demonstrate your identified knowledge about the problem and suggest the next steps.
- A *bibliography*, references to any related work you used in your project. You are encouraged to use the APA 7 citation style but may use different styles as long as you are consistent throughout your report.

We will provide LATEX and docx-style files that we would prefer that you use in writing the report. Reports are to be submitted in the form of a single PDF file. If a report is submitted in any format other than PDF, we reserve the right to return the report with a mark of 0.

Your name and student ID should not appear anywhere in the report, including any metadata (filename, etc.). If we find any such information, we reserve the right to return the report with a mark of 0.

4.6 Number of the models

You should minimally implement and analyse in your report one baseline, and at least two different machine learning models.

Reminder: We are more interested in your critical analysis of methods and results than the raw performance of your models.

You may not be able to arrive at a definitive answer to your research question, which is perfectly fine. However, you should analyse and discuss your (possibly negative) results in depth.

5 Stage II

Task Basics

Once you've submitted your anonymized report through Feedback Fruit AND the main assignment page, you'll receive two reports from your classmates to review. You'll need to read them carefully and provide a peer review for each. Additionally, you'll write a self-reflection report about your own submission.

Peer Review

During the reviewing process, you will read two anonymous submissions by your classmates via Feedback Fruits. This is to help you contemplate some other ways of approaching the project and to ensure that every student receives some extra feedback. You should aim to write 210-360 words per review, responding to three 'questions':

- Briefly summarise what the author has done in one paragraph (70-120 words)
- Indicate what you think that the author has done well, and why in one paragraph (70-120 words)
- Indicate what you think could have been improved, and why in one paragraph (70-120 words).

Please be courteous and professional in the reviewing process. A brief guideline for reviewers can be found here.

Self-Reflection

A comprehensive written reflection piece summarising your critical reflection on the following topics in 400($\pm$10%) words.

- the process of completing this project
- things that you are satisfied with and those that can be improved in your Stage I deliverables, e.g. modelling, evaluation, analysis, and discussion.

The reflection report is individual and not anonymous. Everyone must submit their own reflection on Feedback Fruits.

6 Assessment Criteria

The Project will be marked out of 30 and is worth 30% of your overall mark for the subject. The mark breakdown will be:

Report Quality: (25/30 marks)

You can consult the marking rubric on the Canvas/A2: Open-ended Research Report page which indicates detailed categories that we will be looking for in the report.

Kaggle: (2/30 marks)

The performance of the classifier *(1 mark)* is for submitting (at least) one set of model predictions to the Kaggle competition; and *(1 mark)* to get a reasonable accuracy, e.i., better than our threshold.

Reviews: (2/30 marks)

You will write a review for each of the two reports written by other students; you will follow the guidelines stated above.

Reflection: **(1/30 mark)**

You will write a self-reflection review for your report following the guidelines stated above.

You must submit your code that supports the results presented in your report. If you do not submit an executable code that supports your findings, you will receive **zero** marks for the report section.

Since all the documents exist on the World Wide Web, it is inconvenient but possible to "cheat" and identify some of the class labels from the test data using non-machine learning methods. If there is any evidence of this, the performance of the classifier will be ignored, and you will instead receive a mark of **zero** for this component.

7 Using Kaggle

To give you the possibility of evaluating your models, even more, we will be setting up a Kaggle In-Class competition. You can submit results on the test set there and get immediate feedback on the performance of your system. There is a Leaderboard, that will allow you to see how well you are doing as compared to other classmates participating online.

The Kaggle in-class competition URL is available on Canvas. To participate in the competition:

- Each student should create a Kaggle account (unless they have one already) using your Student-ID
- You may make up to 8 submissions per day. An example submission file can be found on the Kaggle site.
- Submissions will be evaluated by Kaggle for accuracy, against just 30% of the test data, forming the public leaderboard.
- Before the competition closes, you may select a final submission out of the ones submitted previously – by default the submission with the highest public leaderboard score is selected by Kaggle.
- After the competition closes, the public 30% test scores will be replaced with the private leaderboard 100% test scores.

8 Assignment Policies

8.1 Changes/Updates to the Assignment Specifications

We will use Canvas to advertise any (hopefully small-scale) changes or clarifications in the assignment specifications. Any addendums made to the assignment specifications via Canvas will supersede the information contained in this version of the specifications.

8.2 Late Submissions

Late submissions in stage I will bring disruption to the reviewing process. **There will be no extensions granted, and no late submissions allowed to ensure a smooth run of the Stage II process.** Submission will close at 5 pm on May 10th.

You are strongly encouraged to submit by the date and time specified above. For students who are demonstrably unable to submit a full solution in time, we may offer a solution but note that you may be unable to benefit from the peer review process in that case. A solution will be sought on a case-by-case basis. Please email Hasti (hasti.samadi@unimelb.edu.au) with documentation of the reasons for the delay.

8.3 Academic Honesty

While it is acceptable to discuss the assignment with others in general terms, excessive collaboration with students outside of your group is considered cheating. Your submissions will be examined for originality and will invoke the University's Academic Misconduct Policy[2] where either inappropriate levels of collaboration or plagiarism are deemed to have taken place.

We highly recommend (re)taking the academic honesty training module in this subject's Canvas. We will be checking submissions for originality and will invoke the University's Academic Misconduct policy where inappropriate levels of collusion or plagiarism are deemed to have taken place.

Reference:

QAISER, S. and ALI, R., 2018. Text mining: use of TF-IDF to examine the relevance of words to documents. International Journal of Computer Applications, 181(1), pp. 25-29.

SARAEE, M., WHITE, S. and ECCLESTON, J., 2004. A data mining approach to analysis and prediction of movie ratings. WIT Transactions on information and communication technologies, 33.

[2] https://academicintegrity.unimelb.edu.au/home