School of Computing and Information Systems
The University of Melbourne
COMP90049, Introduction to Machine Learning, 2020

Project 2: Romance or Thriller? Movie Genre Prediction from Audio, Visual, and Text Features!

Task:	Build a movie genre classifier
Due:	Stage I: May 22 3pm
	Stage II: May 27 3pm
Submission:	Stage I: Report (PDF) to Turnitin; code to LMS ; test outputs to Kaggle InClass competition
	Stage II: Reviews (via Turnitin PeerMark)
Marks:	The Project will be marked out of 40, and will contribute 40% of your total mark.

Overview

The goal of this project is to build and critically analyse some supervised Machine Learning algorithms, to automatically identify the genre(s) of a movie on the basis of its audio, visual and textual (metadata) features. That is, given a list of movies, your job is to come up with one or more implemented Machine Learning model(s), which are trained using the training dataset, and evaluated using the validation and test dataset.

This project aims to reinforce the largely theoretical machine learning concepts around models, data, and evaluation covered in the lectures, by applying them to an open-ended problem. You will also have an opportunity to practice your general problem-solving skills, written communication skills, and creativity. This project has two stages.

The focus of this assignment will be the report, where you will demonstrate the knowledge that you have gained, in a manner that is accessible to a reasonably informed reader.

Deliverables

1. Stage I (By May 22 UTC+10):
   (a) The predicted labels of the test movies submitted to the Kaggle[1] in-class competition described below
   (b) One or more programs, written in Python, which implement machine learning methods that build the model, make predictions, and evaluate where necessary and a README file that briefly details your implementation.
   (c) An anonymous written report, of 1500-2000 words. Your name and student ID should not appear anywhere in the report, including the metadata (filename, etc.).
2. Stage II (By May 27 UTC+10):

[1] `https://www.kaggle.com/`

(a) Reviews of two reports written by other students, of 200-400 words each.

(b) A written reflection piece of (200-400 words).

Dataset

Each movie (instance) is represented through a large set of features (described in detail in the README), and listed in the *features.tsv* files. Each movie is labelled with a single genre tag, which is provided int the *labels.tsv* files.

The data files are available via the LMS. You will be provided with a set of training documents, validation documents and a set of test documents. The files are provided in *tsv* format, which stands for *tab-separated values*.

- *train_features.tsv*: Contains features of 5240 training instances.
- *train_labels.tsv*: Contains a single genre label for each training instance
- *valid_features.tsv*: Contains features of 299 validation instances.
- *valid_labels.tsv*: Contains a single genre label for each validation instance.
- *test_features.tsv*: Contains features of 298 test instances.

Each movie in the data set is indexed with a unique `movieID`. We provide three different types of features. Details are provided in the README file, as well as the references listed under **Terms of Use**:

- `Metadata features`: For each movie, we provide its `title`, its `year of release`, and a list of `tags` (like *'predictable'*, *'boring'*, or *'based_on_a_book'*) provided by human annotators.
- `Visual features`: We provide 107 pre-computed visual features, pre-extracted from the movie trailer. Each feature has a continuous floating point value. Note that these features are not interpretable.
- `Audio features`: We provide 20 pre-computed audio features capturing acoustic signals from the movie trailer. Again, each feature takes a continuous value, and the values are not interpretable.

In addition to those features, we also provide the following information which you may find useful in your error analysis (but are not required to use):

- A `youtube link` pointing to the movie trailer. We don't guarantee that all links are functional.

Each movie is labelled with its genre, i.e., with a single label from one of 18 possible genre labels:

'Action', 'Adventure', 'Animation', 'Children', 'Comedy', 'Crime', 'Documentary', 'Drama', 'Fantasy', 'Film_noir', 'Horror', 'Musical', 'Mystery', 'Romance', 'Sci_fi', 'Thriller', 'War', 'Western'

Task

You will develop machine learning models which predict the movie genre based on a diverse set of features, capturing acoustic and visual features of the movie trailers, as well as movie metadata such as its title and human-provided tags. You will implement and analyze different machine learning models in their performance; and explore the utility of the different types of features for movie genre prediction.

We will use a hold-out strategy to evaluate the trained model using a validation, and a test set:

1. **The training phase:** This will involve training your classifier(s) and parameter tuning where required. You will use the *train_features* and *train_labels* files.
2. **The validation phase:** This is where you observe the performance of the classifier(s). The validation data is labelled: you should run the classifier that you built in the training phase on this data to calculate one or more evaluation metrics to discuss in your report. This phase will help you to find the best model that can be used for the testing phase.
3. **The testing phase:** The test data is unlabeled; you should use your preferred model to produce a prediction for each test instance, and submit your predictions to the Kaggle website; we will use this output to confirm the observations of your approach.

N.B: Various machine learning techniques have been (or will be) discussed in this subject (Naive Bayes, Decision Trees, 0-R, etc.); many more exist. You are strongly encouraged to make use of machine learning software and/or existing libraries (such as `sklearn`) in your attempts at this project.

Submissions

All submissions will be via the Canvas. Stage I submissions will open one week before the due date. Stage II submissions will be open as soon as the reports are available, immediately following the Stage I submission deadline.

Submissions: Stage I

Submission in Kaggle InClass Competition

To give you the possibility of evaluating your models on the test set, we will be setting up this project on Kaggle InClass competition. You can submit results on the test set there, and get immediate feedback on your model's performance. There is a Leaderboard, that will allow you to see how well you are doing as compared to other classmates participating on-line. The Kaggle in-class competition URL will be announced on Canvas shortly. You will receive marks for submitting (at least) one set of predictions for for the unlabelled test dataset into the competition. However, we won't mark your performance directly, as the focus of this assignment is on the quality of your critical analysis and your report, rather than the performance of your Machine Learning models.

Report

The report should be 1,500-2,000 words in length and should follow the structure of a short research paper. It should provide a basic descriptions of:

1. The task, and a short summary of some related literature
2. What you have done, including any learners that you have used, or features that you have engineered[2]
3. Evaluation of your classifier(s) over the validation dataset

You should also aim to have a more detailed discussion, which:

4. Contextualises the behaviour of the method(s), in terms of the theoretical properties we have identified in the lectures
5. Attempt some error analysis of the method(s)

And don't forget:

6. A bibliography, which includes Deldjoo et al. (2018), Harper et al. (2015), and other related work

Note that we are more interested in seeing evidence of you having thought about the task and determined reasons for the relative performance of different methods, rather than the raw scores of the different methods you select. 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.

We will provide LaTeXand RTF 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.

Submissions: Stage II

At stage II you have to submit the following reports:

Reviews

During the reviewing process, you will read two anonymous submissions by other students. This is to help you contemplate some other ways of approaching the Project, and to ensure that students get some extra feedback. For each paper, you should aim to write 200-400 words total, responding to three *'questions'*:

- Briefly summarise what the author has done in one paragraph (50-100 words)
- Indicate what you think that the author has done well, and why in one paragraph (100-200 words)
- Indicate what you think could have been improved, and why in one paragraph (50-100 words)

[2] This should be at a conceptual level; a detailed description of the code is not appropriate for the report.

Reflective writing

A comprehensive written reflection piece summarizing your critical reflection on this project within 200-400 words. This report is not anonymous.

Assessment Criteria

The Project will be marked out of 40, and is worth 40% of your overall mark for the subject. The mark breakdown will be: ;

Report Quality: (30/40 marks available)
You will explain the practical behaviour of your systems, referring to the theoretical behaviour of the Machine Learning methods where appropriate. You will support your observations with evidence, in terms of evaluation metrics, and, ideally, illustrative examples. You will derive some knowledge about the problem of movie genre classification.
You will produce a formal report, which is commensurate in style and structure with a (short) research paper. You must express your ideas clearly and concisely, and remain within the word limit (1500-2000 words). You will include a short summary of related research.You can use the marking rubric to indicate what we will be looking for in each of these categories when marking.
The report will be marked according to the accompanying rubric.

Reviews: (6/40 marks available)
You will write a review for each of two reports written by other students; you will follow the guidelines stated above.

Self-Reflection: (2/40 marks available)
A comprehensive written reflection piece summarizing your critical reflection on this project within 200-400words.You will follow the guidelines stated above.

Kaggle performance: (2/40 marks)
For submitting (at least) one set of model predictions to the Kaggle competition.

Using Kaggle

The Kaggle in-class competition URL will be announced on LMS shortly. To participate do the following:

- 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.
- Prior to competition close, you may select a final submission out of the ones submitted previously – by default the submission with highest public leaderboard score is selected by Kaggle.

- After competition close, public 30% test scores will be replaced with the private leaderboard 100% test scores.

Terms of Use

The data set is derived from the following resources:

> Deldjoo, Yashar and Constantin, Mihai Gabriel and Schedl, Markus and Ionescu, Bogdan and Cremonesi, Paolo. MMTF-14K: A Multifaceted Movie Trailer Feature Dataset for Recommendation and Retrieval. Proceedings of the 9th ACM Multimedia Systems Conference, MMSys 2018, Amsterdam, The Netherlands, June 12-15, 2018
>
> F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19 (December 2015)

These references **must** be cited in the bibliography. We reserve the right mark of any submission lacking these references with a 0, due to violation of the Terms of Use.

Please note that the dataset is a sample of actual data posted to the World Wide Web. As such, it may contain information that is in poor taste, or that could be construed as offensive. We would ask you, as much as possible, to look beyond this to the task at hand. If you object to these terms, please contact us (`lea.frermann@unimelb.edu.au`) as soon as possible.

Changes/Updates to the Project Specifications

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

Late Submission Policy

There will be **no extensions** granted, and **no late submissions** allowed. Submission will close at **TODO**. For students who are demonstrably unable to submit a full solution in time, we offer to reduce the weighting of the mark of this assignment towards the overall course grade (but you will still have to submit your solutions by the deadline).

Academic Misconduct

For most people, collaboration will form a natural part of the undertaking of this project. However, it is still an individual task, and so reuse of ideas or excessive influence in algorithm choice and development will be considered cheating. We will be checking submissions for originality and will invoke the University's Academic Misconduct policy (http://academichonesty.unimelb.edu.au/policy.html) where inappropriate levels of collusion or plagiarism are deemed to have taken place.