School of Computing and Information Systems
The University of Melbourne
COMP90049, Introduction to Machine Learning, Semester 2 2020

Project 2: Music Genre Prediction from Audio, Metadata and Text Features!

Task:	Build a music genre classifier
Due:	Stage I: October 16 5PM
	Stage II: October 21 5PM
Submission:	Stage I: Report (PDF) to Turnitin; test outputs to Kaggle InClass Competition; code to LMS
	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 of a song on the basis of its audio, metadata and textual features. That is, given a list of songs, your job is to implement one or more Machine Learning model(s), train them using the training dataset, and evaluate using the test 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. At *Stage I*, the focus will be the implementation, experimentation, and the report, where you will demonstrate the knowledge that you have gained, in a manner that is accessible to a reasonably informed reader. At *Stage II*, you will review 2 reports written by your peers in Stage I and submit a review of about 200-400 words.

Deliverables

1. Stage I (By October 16 5PM):
   (a) The predicted labels of the test songs submitted to the Kaggle InClass Competition described below
   (b) (1) One or more programs, written in Python, which implement Machine Learning models, make predictions, and evaluate and (2) a README file that briefly details your implementation.
   (c) An **anonymous** written report, of 1800±10% words. Your name and student ID should **not** appear anywhere in the report, including the metadata (filename, etc.).
2. Stage II (By October 21 5PM):
   (a) Reviews of two reports written by other students, of 200-400 words each.

Dataset

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

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

- *train_features.csv*: Contains features of 7678 training instances.
- *train_labels.csv*: Contains a single genre label for each training instance
- *valid_features.csv*: Contains features of 450 validation instances.
- *valid_labels.csv*: Contains a single genre label for each validation instance.
- *test_features.csv*: Contains features of 428 test instances.

Each song in the data set is indexed with a unique `trackID`. 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 song, we provide its `title`, `loudness`, `tempo`, `key`, `mode`, `duration`, and `time_signature` .
- `Text features:` For each song, we provide a list of tags representing the words that appeared in the lyrics of the song and are human annotated (such as *'dance'*, *'love'*, or *'never*).
- `Audio features:` We provide 148 pre-computed audio features that were pre-extracted from the 30 or 60 second snippets of each track, and capture timbre, chroma, and 'Mel Frequency Cepstral Coefficients' (MFCC) aspects of the audio. Each feature is continuous and the values are not interpretable.

Each song is labelled with its genre, i.e., with a single label from one of 8 possible genre labels:

> 'Soul and Reggae', 'Pop', 'Punk', ' Jazz and Blues', 'Dance and Electronica', 'Folk', 'Classic Pop and Rock', 'Metal'

Task

You will develop Machine Learning models which predict the music genre based on a diverse set of features, capturing audio features of the track, as well as song metadata such as its title, key, duration, and textual tags representing the words in the song lyrics. You will implement and analyze different Machine Learning models in their performance; and explore the utility of the different types of features for music 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_lyrics* 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 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 existing Machine Learning libraries (such as sklearn) in your attempts at this project.

Submissions

All submission 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 models 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 InClass Competition URL will be announced on Canvas shortly.You will receive marks for submitting (at least) one set of predictions for the unlabelled test dataset into the competition. However, we wont 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 $1800 \pm 10\%$ words in length and provide a basic description 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[1]

3. Evaluation of your classifier(s) over the validation dataset

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

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

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 Bertin-Mahieux et al. (2011), as well as A. Schindler and A. Rauber (2012), 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.

Code

You must submit one or more programs, written in Python, which implement Machine Learning models, make predictions, and evaluate as well as a README file that briefly details your implementation. Note that you are strongly encouraged to use existing Machine learning libraries (such as sklearn) in your attempt at building a model. Your code must include the following functions:

- *train()*: where you build your Machine Learning model from the training data.
- *predict()*: where you use your trained model to predict a class for the validation data.
- *evaluate()*: where you will output the accuracy of your model based on an evaluation metric.

You may implement other functions as you require.

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)

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 music genre prediction.

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 (1800$\pm$10% words). You will include a short summary of related research.The marking rubric indicates what we will be looking for in each of these categories when marking.

Reviews: (8/40 marks available)

You will write a review for each of two reports written by other students; 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 InClass 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 closes, 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 closes, 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:

> T. Bertin-Mahieux, D. P.W. Ellis, B. Whitman, and P. Lamere. The million song dataset. In Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR), 2011.
>
> A. Schindler and A. Rauber. Capturing the temporal domain in Echonest Features for improved classification effectiveness. In Proceedings of the 10th International Workshop on Adaptive Multimedia Retrieval (AMR) , 2012.

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 (`rashidil@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

You are strongly encouraged to submit by the time and date specified above, however, if circumstances do not permit this, then the marks will be adjusted as follows:

- Each day (including working days and holidays) that the report is submitted after the due date (and time) specified above, 10% (4 marks) will be deducted from the marks available, up until 5 days has passed, after which regular submissions will no longer be accepted.
- Due to the end of semester, and the inherent inconvenience caused by late submission of reviews, any submission after the reviewing system closes will incur a flat 50% penalty (i.e. 4 of the 8 marks available); reviews submitted more than 5 days after the deadline will not be assessed.

Note that submitting the report late will mean that you may lose the opportunity for your report to participate in the reviewing process, which means that you will receive less feedback.

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 Universitys Academic Misconduct policy (http://academichonesty.unimelb.edu.au/policy.html) where inappropriate levels of collusion or plagiarism are deemed to have taken place.