Movie Genre Classification with Machine Learning Algorithms

The University of Melbourne

Semester 1 2020

1 Introduction

Students in the class COMP90049 were required to conduct a self-guided research task, focusing on movie genre classification using machine learning algorithms. The aim of the task was to find new knowledge in the field of film genre classification. The students were given a dataset for training, validation and testing. The MMTF-14K dataset, compiled by Deldjoo et al.[6], contains audio, visual and metadata taken from film trailers. The audio and visual data is in the form of floating-point numbers, making it impossible to interpret without knowledge of the methods used to compile the dataset.

1.1 Hypothesis Formulation

Prior to building a hypothesis or conducting any analysis on what influences movie genre predictability, it is important to note the preliminary thoughts and observations that were made about the dataset. These were central to shaping the course of the research. The tags feature was viewed as the feature with the highest likelihood of predicting genre. While some films had tags that did not infer anything about genre, for example simply "clv" or "dvd", others contained tags which directly stated the films genre. This formed the hypothesis that the tags feature was likely to be the strongest predictor of genre.

The extent to which the tags feature would outperform the other features was highly speculative, because prior to testing there was no understanding of how well the audio and visual features would predict film genre. The researcher was confident that the tag feature would outperform other metadata features because the tags

were more indicative of film genre than the film year or title because they are vague and non-specific.

Hypothesis: **The tags feature will predict film genre better than any other feature in the dataset.**

2 Related Works

There have been a number of studies conducted on predicting film genre with machine learning algorithms in the academic community. Table 1 summarises four papers and their findings; the common theme was to increase classifier accuracy, using visual and audio features. While a high degree of accuracy was achieved for each study, the number of genres which a film was classified amongst was very few, between 3 - 7. In a realistic scenario, the classifier would need to be able to classify a film's genre amongst a greater number of genres.

	H.Y. Huang[1]	**Z. Rasheed[2]**	**S.K.Jain[3]**	**Y.F. Huang[4]**
# of Genres	3	4	5	7
Features	Visual	Visual	Visual-Audio	Visual-Audio
Feature Sel.	N	N	N	Y
Classifier	MLP	Mean Shift	MLP	SVM
Accuracy	80.20%	83%	87.50%	91.90%

Table 1: Comparison of related studies

Most papers focusing on film genre classification have been concerned with the performance of different algorithms and optimisation. By comparison, the question of which features are the strongest predictors of film genre remains relatively under-researched. Coupled with the view that previous studies' results were derived with too few genres, it is warranted that further study be conducted on which kind of features are the strongest predictors of film genre.

2.1 Baseline: Zero R

The Zero R method gave a baseline of 15.095% accuracy. The Zero R method identifies the most frequent label and applies that label to every instance. It is expected that any reasonably well-developed algorithm would beat the baseline.

3 Methodology

A series of pre-processing techniques were utilised to get the data to a stage where it could be fed to the classification algorithms. The title and tag features were vectorized with the CountVectorizer [9] functionality. The CountVectorizer converts a collection of text data to a matrix of token counts. In the case of the tags feature, the vectorizer assigns a new column for each of the unique tags across the entire dataset. If a film contains a certain tag, then it will read "1" in the column assigned to that tag, and "0" otherwise. In order to assign the same columns to all the three datasets, they had to be concatenated, processed, and then split up into their original datasets.

The year feature and the YouTube Ids were discarded. Early assessment of their ability to predict genre warranted them unworthy of use. The YouTube Ids made no positive contribution to the prediction of film genre. To derive value from the YouTube Ids additional data would have to captured from each film's YouTube page, which was out of the scope for this research. The year feature was found to contribute negatively to genre prediction. Classifiers were trained and validated with the year feature alone, which produced poor results (Figure 1). The feature only just managed to beat the baseline in most cases. It was tested in conjunction with the tags feature, and either gave the same accuracy or worse. As a result, it was concluded that the year and YouTube Id features were poor predictors of film genre and so they were removed from further testing.

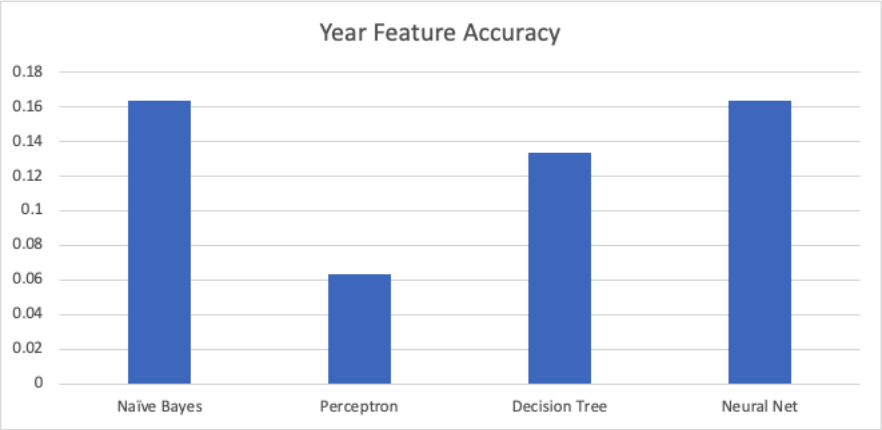


Figure 1: Classifier accuracy, trained and validated with the year feature

Computing time was an issue in some stages of the experimentation, and certain techniques were put in place to reduce the complexity of the task. After vectorization, the tags column became a 208-dimensional matrix whereas the title column

became a 6284-dimensionsional matrix. In conjunction with giving poor results (accuracy of 0.164 with MLP Classifier), it was necessary to conduct further processing on the titles matrix to reduce the computational power necessary to conduct experimentation, as well as to eliminate data that was not contributing to the classification process. Stop words were removed from the titles and each word was lemmatized. The number of columns in the vectorized titles matrix was reduced to 5308. This increased the accuracy of the MLP Classifier when testing to 0.177.

4 Results

Four different classifiers were tested over the validation dataset, and the results are shown in Table 2.

	Titles	**Tags**	**Visual**	**Audio**	**Visual & Audio**	**Average**
Naïve Bayes	0.0803	0.0803	0.1003	0.2107	0.1003	0.1144
Perceptron	0.1472	0.2776	0.0468	0.1271	0.0669	0.1331
Decision Tree	0.1204	0.3044	0.1304	0.1104	0.1572	0.1646
Neural Net	0.1605	0.3846	0.2107	0.194	0.2241	0.2348
Average	0.1425	0.2896	0.1351	0.1672	0.1445	

Table 2: Results of four classifiers trained and validated on the corresponding features

The results suggest that the hypothesis may be correct; the tags feature is the best predictor of film genre.

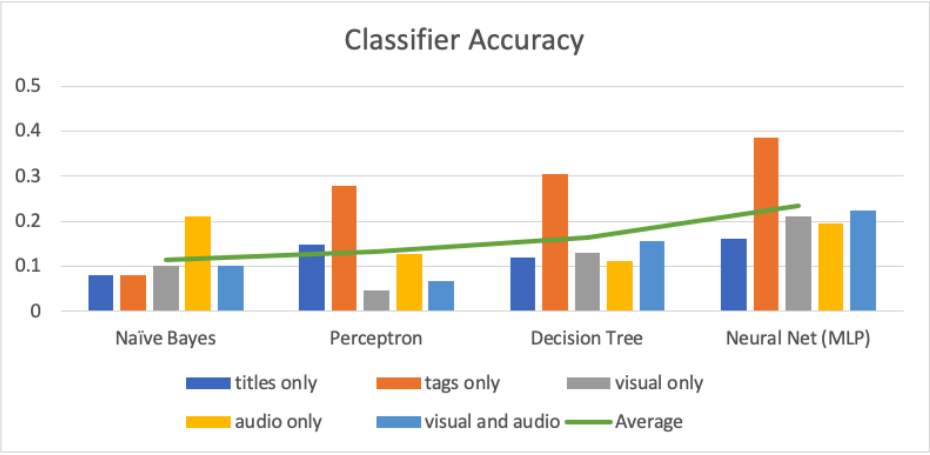


Figure 2: Classifier accuracy, trained and validated with each feature

The results show that the tags feature was the best predictor of genre for three of the four classifiers used, making a strong case that the hypothesis is true. In an attempt to verify this argument, further testing will be performed to try and disprove the hypothesis, and some insight will be given into why the classifiers gave their respective performances.

	Precision	**Recall**	**F1-score**	**Support**
Action	0	0	0	6
Adventure	0	0	0	2
Animation	0	0	0	3
Children	0.12	0.33	0.18	3
Comedy	0.39	0.34	0.37	38
Crime	0.4	0.4	0.4	5
Documentary	0.65	0.61	0.63	18
Drama	0.32	0.4	0.35	43
Fantasy	0.62	0.28	0.38	18
Film Noir	0.5	0.25	0.33	4
Horror	0.44	0.5	0.47	8
Musical	0.2	0.1	0.13	10
Mystery	1	0.06	0.11	18
Romance	0.28	0.51	0.36	51
Sci Fi	0.6	0.56	0.58	16
Thriller	0.39	0.54	0.45	28
War	0.88	0.33	0.48	21
Western	0	0	0	7

Table 3: Evaluation of the MLP classifier, trained and validated with the tags feature

To better understand the data and film predictability, Precision, Recall and F1-score are evaluated for the results of the MLP classifier's predictions on the tags feature (with default classifier parameters). The results show high variance in genre predictability. The classifier performed particularly poorly on genres with a small support, namely Action, Adventure, Animation and Western. These films represent a small number of the overall films in the training data, suggesting that the algorithm is biased towards predicting a genre which is more representative of the dataset.

Genres like War and Fantasy had high precision and low recall, meaning that if the algorithm predicted a film to be one of these genres it was often correct, although many films of these genres went undetected. This may be due to certain tags which are highly associated with these genres; if these certain tags are not present then the algorithm fails to detect the genre.

4.1 Classifier Analysis

The Gaussian Naïve Bayes and Perceptron classifiers achieved lower accuracy than the other classifiers on average. Both classifiers are typically intended for use with linearly separable data, thus a likely cause for their poor performance is that the data is linearly non-separable. The graphs in figure 3 show the non-separable nature of the data subsets used in the analyses; no single hyperplane can separate the two classes. In practice the data was higher in dimensionality, although to illustrate the non-separable property, two dimensions are compared.

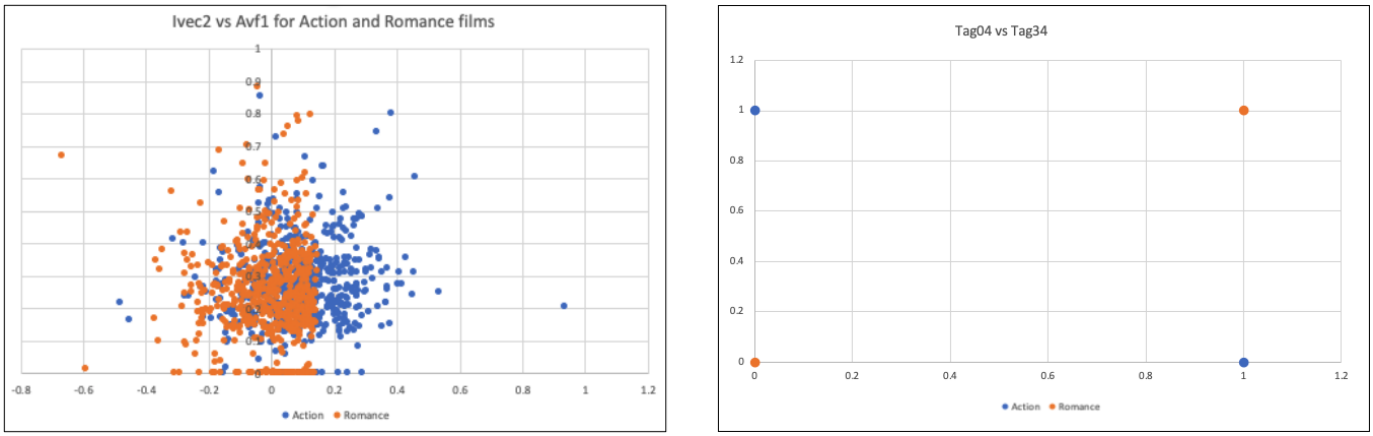


Figure 3: Linearly non-separable nature of data, shown by graphing two sub features against each other

The Decision Tree marginally outperformed both the Naïve Bayes and Perceptron classifiers on average which may be due to its ability to classify linearly non-separable data. The Decision Tree classified the films with 30.4% accuracy when trained and validated on the tag feature, however fell short of the baseline on all other features except the combined visual and audio features. The problem of overfitting often leads to low accuracy with Decision Trees. Simply put, overfitting is when the classifier becomes too specific to the training data and cannot generalise well with new unseen data. Overfitting typically occurs when training Decision Trees on datasets with a large number of features and feature values, which is the case for the MMTF-14K dataset. Given the number of leaf nodes (between 2257 – 4219 for the four decision trees), overfitting most definitely occurred for each feature. Figure 4 shows no correlation between the size of the tree (number of leaves) and the accuracy of the Decision Tree, suggesting the reason for low accuracy is multifaceted, and may be caused by factors other than overfitting alone.

A common method to address overfitting is pruning, which is performed after the training is complete. Pruning removes decision nodes such that the overall accuracy

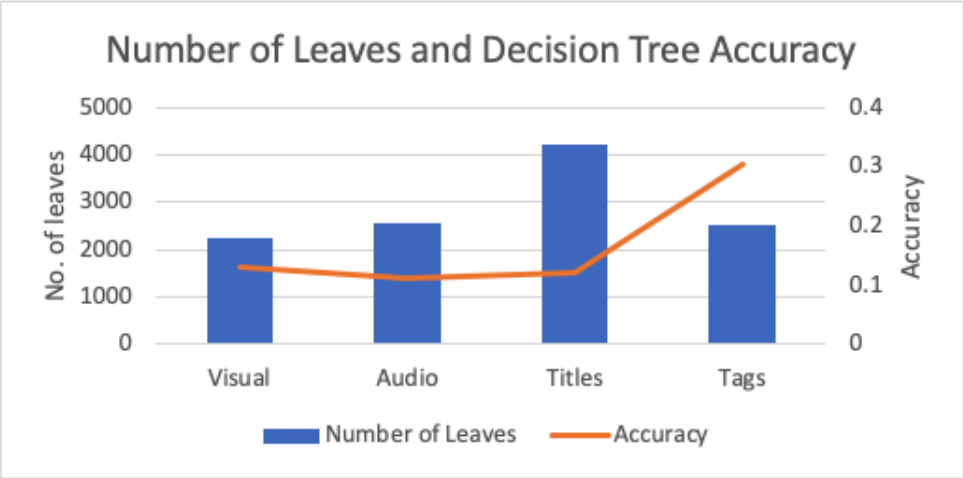
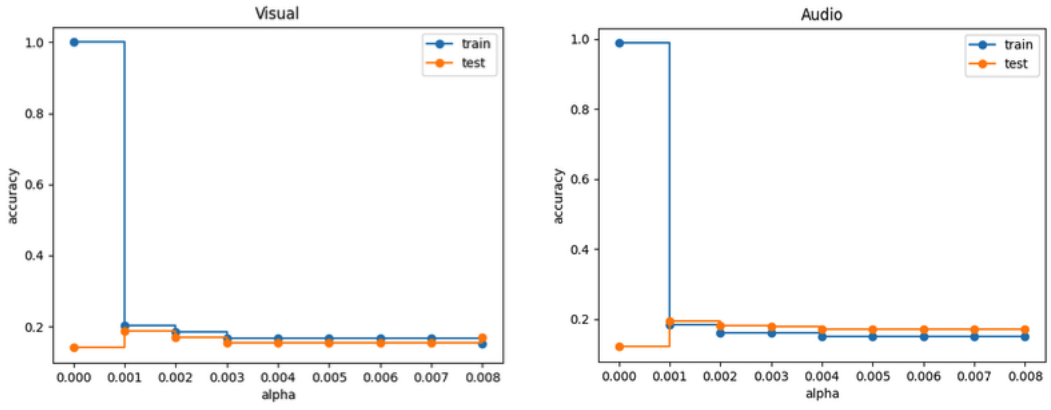


Figure 4: No apparent correlation between tree size and accuracy

is not reduced. The "alpha" value is the measure of granularity of the pruning. As alpha increases, accuracy is reduced from the training data as the model becomes more generalised and less specific to the training data [6]. The graphs below (figure 5) illustrate this for each of the features (except titles, which was too computation heavy to execute), and show that an alpha value between 0.001 – 0.002 achieves a higher accuracy for each feature. The accuracy improvement from pruning was very small overall, suggesting that either the Decision Tree classifier is sub optimal for film genre classification, the features on their own are poor predictors of genre, or the necessary parameters for higher accuracy have not been explored. The results further suggest that the title, audio and visual features may be poorer predictors of film genre, and that the tag feature is the strongest predictor of genre.



The Multi-Layered Perceptron (Neural Network) achieved the best results of all the

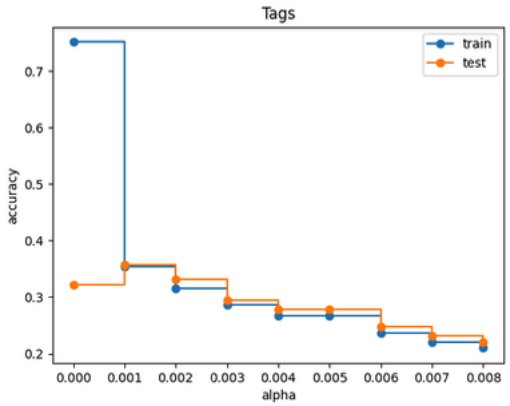


Figure 5: As the value of alpha is increased (more of the tree is pruned), the training accuracy decreases while the validation accuracy increases until a point

classifiers across all features except audio. This is likely due to its ability to handle linearly non-separable data. In an attempt to increase accuracy, the solver and alpha parameters are altered for different values (results shown in figure 6). The results did not provide any major insight, although they indicated that a higher alpha value and selection of the adam solver achieved higher accuracy, so these parameters were used for further analyses.

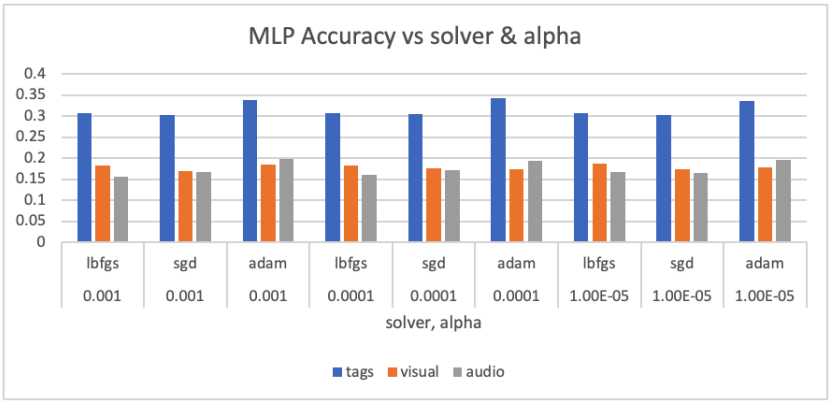


Figure 6: solver = adam, alpha = 0.001 yielded the highest accuracy

Due to computation constraints, it is more efficient to parse the data through the neural network in mini batches rather than parsing all 5240 training instances at once. The problem with smaller batches is that the stochastic gradient descent becomes a noisy operation, as smaller subsets of the data often point the optimiser away from the minima due to subsets of data being non-representative of the entire dataset. Larger batches are known to produce poor results because the gradient

descent optimizer often settles at sharp minima, which are poor at generalizing for the test data[5], however this was not the case for the tags or audio features. The graph below shows how larger batch size slightly increased the accuracy of the classifier, suggesting that gradient descent issues were only present for the visual feature.

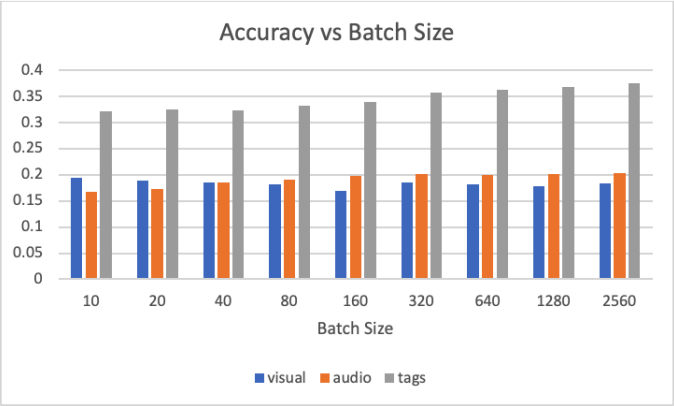


Figure 7: As the batch size increases, the node weight updating process is more informed. Greater accuracy is yielded as a result

5 Conclusion

The task of predicting film genre from audio, visual and metadata is complex and is yet to be mastered by machine learning techniques. Brief analysis of related works showed that the majority of research into the topic has been focused on classifier optimisation, classifier comparison, and visual and audio data processing techniques. Visual and audio features have been shown to predict a film's genre with high accuracy in some cases, although the techniques used in this research were not able to achieve the same result. With the objective of generating new knowledge in the field of film genre prediction, the research aimed to dig deeper into which feature was most predictive of genre. The results of the various analyses conducted were aligned with the hypothesis that the tag feature is most predictive.

It is likely there are combinations of features or pre-processing techniques that were not utilised which would have yielded better results. To produce a stronger argument that the tags feature is the best predictor of film genre, further research into feature combination and data analysis would be necessary. An understanding of the extraction methodology for the visual and audio features would also complement further research.

6 References

1. Huang, H.Y., Shih, W.S., Hsu, W.H.: Movie classification using visual effect features. In: Proc. the IEEE Workshop on Signal Processing Systems, pp. 295–300 (2007)

2. Z. Rasheed, Y. Sheikh, and M. Shah. 2005. On the use of computable features for film classification. IEEE Trans. Cir. and Sys. for Video Technol. 15, 52–64 (2005)

3. Jain, S.K., Jadon, R.S.: Movies genres classifier using neural network. In: Proc. the 24th International Symposium on Computer and Information Sciences, pp. 575–580 (2009)

4. Huang, Yin-Fu and Wang, Shih-Hao. Movie Genre Classification Using SVM with Audio and Video Features. (2012)

5. N. Sh. Keskar, Dh. Mudigere, J. Nocedal, M. Smelyanskiy and P.T.- P. Tang, "on Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima (2017)

6. Elomaa T. The Biases of Decision Tree Pruning Strategies. In: Hand D.J., Kok J.N., Berthold M.R. (eds) Advances in Intelligent Data Analysis. IDA 1999. Lecture Notes in Computer Science, vol 1642. (1999)

The dataset is derived from the following sources:

7. Deldjoo, Yashar and Constantin, Mihai Gabriel and Schedl, Markus and Ionescu, Bogdan and Cremonesi, Paolo. MMTF-14K: A Multifaceted Movie Trailer Feature Dataset for Recommenda- tion and Retrieval. Proceedings of the 9th ACM Multimedia Systems Conference, MMSys 2018, Amsterdam, The Netherlands, June 12-15, (2018)

8. 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)

The following packages were used in system implemenation:

9. Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, (2011)

10. NLTK: Bird, Steven, Edward Loper and Ewan Klein, Natural Language Processing with Python. O'Reilly Media Inc. (2009)