COMP90049 Assignment 2 Report: Movie Genre Prediction in Machine Learning

Anonymous

1 Introduction

Machine learning has been widely adopted in contemporary world and it's continuously growing and evolving exponentially. Nearly all applications and software systems utilize this technique more or less at some extent, and part of the most notable areas such as virtual personal assistance, product recommendation, social media services and movie genre prediction are hugely benefited, becoming more intelligence and more accurate than ever.

Among these areas, as the amount of online data increases rapidly, human have reached a point where we can't comprehend them all in a decent manner, therefore, classification techniques have been introduced to help movie genre classification. Many attributes of a movie such as visual, audio content, tag, scene, and caption can be utilised by the data scientists to process the data in a suitable way. This project aims to apply three different supervised machine learning methods - Neural Network, Naive Bayes, and Random Forest, to deeply discuss and evaluate their accuracy plus performance, and explore the reasons behind it.

2 Dataset and Data Pre-processing

The movie dataset established by Deldjoo, Yashar and Constantin (2018), Maxwell Harper and Joseph (2015) contains 5240 training instances and 299 validation instances with labels, 298 test instances without labels (refer Table 1), there are three different types of features for each - metadata features, visual features and audio features; as well as 18 possible genre labels which includes romance, thriller, crime, comedy, musical, documentary, drama, adventure, war, horror, children, film-noir, science fiction, mystery, fantasy, action, western and animation. Despite a YouTube link is also provided for error analysis purpose, but considering some of the links are not functional, therefore it will be ignored out of fairness. Hold-out strategy is selected to evaluate the trained model using a validation and a test set.

Dataset	Number of Movies	
Training	5240 instances with label (327 features in total)	avf: 107
		ivec: 20
		tags: 200
Validation	299 instances with label (327 features in total)	avf: 107
		ivec: 20
		tags: 200
Testing	298 instances with label (327 features in total)	avf: 107
		ivec: 20
		tags: 200

Table 1: Movie Dataset

Through the data pro-processing, features like *title*, $YTid$ and *year* are removed as they do not have any correlation with the predicted result. Additionally, audio features ($avf1 - avf107$) and visual features ($ivec1 - ivec20$) are transformed to NumPy array. For feature tag, one of the feature engineering techniques - one hot encoding is performed, that encodes all the tags to value 1 if current instance contains the tag, 0 otherwise.

3 Hypothesis

One main issue needs to be considered is what the mutual attributes are between movies that belong to the same genre. In general, action movies will have more action scene than romance movies, and romance movies will have

more intimacy scene than action movies. Base on such premise, this project assumes that the visual and audio features would show a correlation with the genre of a movie, and aims to verify whether the audio and video feature can be used to help identify the genre of a movie.

4 Methodology

Three supervised machine learning methods have been implemented, and in the following section, they will be deeply compared and evaluated.

4.1 Naive Bayes

In Naive Bayes classifier, we assume that a particular feature has zero correlation with other features in a class (Lewis, 1998). Therefore the joint distribution can be easily computed by multiplying each feature's probability. Because Naive Bayes assumes that features are independent, the probability is incorrect if this assumption is incorrect.

Assuming the output of the classifier has k class labels, which are $c_1, c_2, ..., c_k$ respectively, and y is rely on $x = (x_1, x_2, ..., x_n)$, for any given X, Bayes' Theorem can be represented as:

$$P(y = c_m|X) = \frac{P(y = c_m)P(X|y = c_m)}{P(X)}$$

Based on the assumption that features are independent from each other, therefore:

$$P(x_i|y = c_m, x1, ..., x_n) = P(x_i|y = c_m)$$

Hence:

$$P(y = c_m|X) = \frac{P(y = c_m)\prod_{i=1}^{n} P(x_i|y = c_m)}{P(X)}$$

When the training set is fixed, $P(X)$ is equivalent to a constant, therefore on the test set, we can solely compare the numerator of the above equation when using the attribute vector X to perform the classification.

4.2 Random Forest

Random Forest is one of the many machine learning methods used for supervised learning, which indicates for learning from labelled data as well as conducting predictions in terms of the learnt patterns. Random Forest is able to apply in both regression and classification tasks. It builds up multiple decision trees and combines them together to enhance the prediction to be more accurate and stable (Donges, 2019).

There are two stages in Random Forest algorithm. In stage one, Random Forest is created by randomly selecting k features from the total n features; calculate the node using the best split point among these selected features; using the best split technique to split the node into daughter node; the above processes are repeated until certain number of nodes are reached; finally, the whole above processes are repeated for n times to create n trees. In stage two, perform the prediction on the test set for each created decision tree; calculate the votes for each predicted target, and take the vote with the highest value as the final result.

4.3 Neural Network

Neural Network is consisted with multiple neurons that are connected in a certain pattern. Some of the rules of Neural Network:

- Neurons are distributed layer by layer, the left most layer is the input layer, the right most layer is the output layer, the layers between these two layers are hidden layers, as they are invisible for the observers.
- There is no connection between neurons of the same layer.
- Every connection has a weight
- Every neuron of layer n connected with all the neurons of layer n - 1.

calculating the output of the Neural Network according to the input, we need to first assign the value of each element of the input x to the corresponding neuron of the input layer, next, calculate the value of each neuron of each layer based on: $y = f_{\text{network}}(x)$. Finally, the output vector can be obtained by concatenating the value of each neuron in the output layer (Lecun, 2015).

5 Evaluation

In this section, classical machine learning evaluation indicators - accuracy, precision, recall, f1-score are introduced for result analysis.

- Accuracy: the number of correct predictions over all predictions that the model made.
- Precision: the percentage of positive cases that are correctly identified.
- Recall: the percentage of actual positive cases that are correctly identified.
- F1-score: the single metric to measure both precision and recall.

These indicators can be calculated as follow:

$$Precision(class_m) = \frac{TP(class_m)}{TP(class_m) + FP(class_m)}$$

$$Recall(class_m) = \frac{TP(class_m)}{TP(class_m) + FN(class_m)}$$

$$F1(class_m) = 2 * \frac{precision * recall}{precision + recall}$$

6 Results Analysis

The overall and detailed (As 18 types of labels are too many to show in this report, therefore we will only take the first five) distribution of aforementioned metrics are shown in Figure 1 and Figure 2.

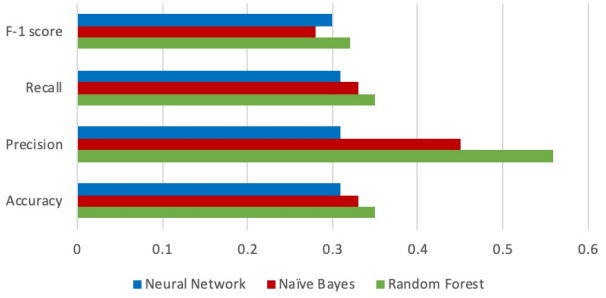

Figure 1: Result of Three ML Models

Neuron Network can achieve up to 98% accuracy for training set with 3 x 80 hidden layers. By adjusting the number of neurons in hidden layers, it has been discovered that the more the neurons, the more accurate the result is, however, the performance does not differ much when the number of neurons reach 70 or more per layer. Random Forest's 53% accuracy is also acceptable, Naive Bayes which is the lowest, only achieved 36% accuracy. However, one notable fact is that Neural Network is the lowest (31%) when it comes to validation set, three models' performance is more or less around 33%. The phenomenon that the

Class\Model	Random Forest		Naïve Bayes	Neural Network
Thriller	Precision	0.27	0.28	0.33
	Recall	0.57	0.57	0.43
	F1	0.36	0.37	0.38
	Accuracy	0.35	0.33	0.31
Romance	Precision	0.43	0.62	0.32
	Recall	0.31	0.16	0.24
	F1	0.36	0.25	0.27
	Accuracy	0.35	0.33	0.31
Documentary	Precision	0.67	0.38	0.41
	Recall	0.22	0.78	0.5
	F1	0.33	0.51	0.45
	Accuracy	0.35	0.33	0.31
Comeduy	Precision	0.36	0.28	0.36
	Recall	0.58	0.77	0.37
	F1	0.44	0.41	0.36
	Accuracy	0.35	0.33	0.31
Musical	Precision	0.33	0.51	0.25
	Recall	0.1	0.19	0.1
	F1	0.15	0.28	0.14
	Accuracy	0.35	0.33	0.31

Figure 2: ML Model Detailed Performance

validation accuracy is much lower than the training accuracy, is most likely due to the size of the dataset, directly comparing these two sets with huge difference in size (5240 VS 299) is committed unfair.

Conducting dimensionality deduction by only utilizing *tag* feature, the result shows that all three models' performance have been negatively affected for a significant portion as shown in Figure 3, whereas Naive Bayes appears to be more sensitive to this reduction than the other two, accuracy has dropped from 33% to 22%. The fact that leads to this cause is likely due to Naive Bayes is derived under the assumption that the features are independent of each other, but often the actual data is difficult satisfy this condition. Therefore, as

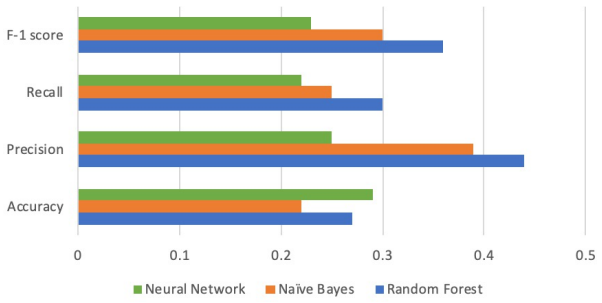

Figure 3: Performance Without Audio/Visual Features

we assumed previously, the implementation of visual and audio attributes improves the accuracy of the whole system.

7 Conclusion

In summary, this report assessed the performance of Naive Bayes, Random Forest and Neural Network under the scenario of identifying movie genre, Random Forest outperforms the rest of two models in terms of overall accuracy. It can be concluded that the visual and audio attributes have a positive effect on the movie genre classification. For future improvement, feature engineering and a training set with larger size will be considered for future attempt.

References

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

Donges, N. (2019). *A complete guide to the random forest algorithm,* Buildin. Retrieved 15 May 2020, from https://builtin.com/data-science/random-forest-algorithm

LeCun, Y., Bengio, Y. and Hinton, G., (2015). Deep learning. *Nature*, 521(7553), pp.436-444

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)

Lewis D.D. (1998) Naive (Bayes) at forty: *The independence assumption in information retrieval*. In: Nédellec C., Rouveirol C. (eds) Machine Learning: ECML-98. ECML 1998. Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence), vol 1398. Springer, Berlin, Heidelberg