COMP90049 Project 2 Report

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

1 Introduction

In this report, we will explore a movie dataset which contains metadata attributes such as "title" and "tag", and some pre-computed audio and visual attributes. Then, we will use this dataset to train machine learning models for predicting movies' genres. This dataset is derived from the MMTF-14K [1] and MovieLens [2] datasets.

We will use this derived dataset to train a Naive Bayes (NB) classifier and a multilayer perceptron (MLP) model, then compare them with Zero-R baseline model to examine their performance. In this report, three main hypotheses were proposed:

I. The simple NB classifier performance would not be too bad (accuracy difference within 10%) compared with MLP model given the categorical property of the "tag" attribute of our dataset.

   There are two sub-hypotheses to support this statement.

   A. The tag attribute would dominate the performance.

   B. NB classifier could perform well on the dataset with one dominant categorical attribute.

II. The audio and visual attributes could help improve movie genres' classification accuracy.

III. Complex MLP model would perform better

   A. Using "title" attribute could help the classification.

   B. Using complex hidden layer architecture could lead to better classification accuracy.

In the following section, we will examine the hypotheses by building models for testing. The models we built are based on the *Scikit-learn* module[3].

2 Literature Review

[1] pointed out the lack of key attributes of previous movie databases, the audio and visual attributes. They stated that, the same story could be shot into different movie genres. Therefore, the addition of the audio and visual attributes could reflect the real movie content better. In this report we would verify this statement by experiments.

3 Methods

In this section, we will show how the data be preprocessed and which machine learning model we would test.

3.1 Data Preprocessing

3.1.1 Drop Unnecessary Attributes

To achieve the best model performance, we first dropped the attribute which seems to have little contribution to genres classification.

The "movieId" and "YTID" were dropped because they are meaningless for genre classification.

Then we plotted the "number of movies" against the "year" of every genre to see the year attribute's influence on genre. To avoid lengthy, we only show two of them below. As Figure 1 and 2 indicate, these genres have a similar distribution and trend. Due to that, we also drop the attribute "year".

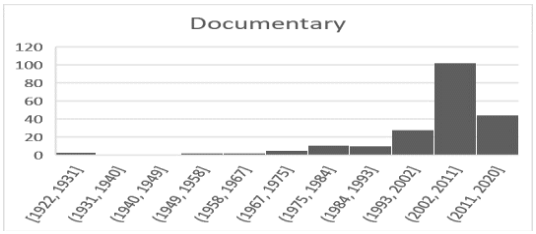

Figure 1- Genre "Documentary" Distribution

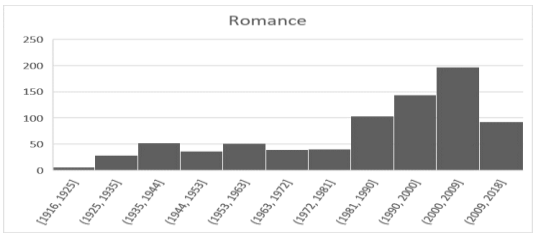

Figure 2- Genre "Romance" Distribution

3.1.2 Categorizing Audio-Visual Data

Due to the categorial property of NB classifier, we needed to categorize the continuous attributes into classes. First, we analysed the audio and visual data (see Table 1). Then, we divide each audio and visual value into 4 categories by their relative magnitude in training set. For example, value with the magnitude in 0~25% of the whole attribute's value range would be assigned to class 1.

	avf1	*avf2*	*avf3*
min	0	0	0.009692
25%	0.182695	0.234042	0.193495
50%	0.260192	0.315934	0.257108
75%	0.347781	0.410342	0.332876
max	0.881001	0.957703	0.929619

Table 1- Visual Attributes Analysis

After categorizing the continuous data, we could directly feed them into NB model.

3.1.3 Attributes Encoding

One-hot encoding and TF-IDF encoding are used on tag and title feature to examine the attributes' influence on classification accuracy.

3.2 Machine Learning Models

3.2.1 Baseline Model – Zero-R

We use the Zero-R model as the baseline. The most common class in the training dataset is "Romance". Using this genre as all the class of validation dataset leads to an accuracy of 17.06%.

3.2.2 Naive Bayes Model

To determine which NB mode is best for our dataset, I have plotted all the audio and visual data to see their distribution (Figure 3).

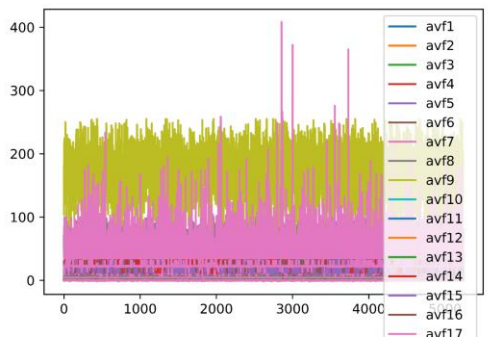

Figure 3- Audio and Video Data Distribution

Figure 3 indicates that the data is randomly distributed, not in Gaussian or other distribution. Due to that we can assume the basic NB classifier (*MultinomialNB* [4] in *Scikit-learn* module) would outperform other NB.

To examine my hypothesis and find out the best NB model for our task, we tried different NB models with categorized audio-visual data and one-hot encoding for tags on the validation data set (see Table 2).

Model Name	*Accuracy*
Complement NB	37.46 %
Multinomial NB	37.12 %
Categorical NB	29.77 %
Gaussian NB	9.36 %

Table 2- NB Model Comparison

As Table 2 shows, the Multinomial NB indeed outperform Gaussian and Categorical NB, because our data are neither in Gaussian distribution nor each attributed has its own categorical distribution. Moreover, we could find that the Complement NB (CNB) is slightly better than Multinomial NB. This is because our data set is not balanced; some classes such as "Romance" has many more instances than other classes. It will shrink the weight of classes with fewer instances in Multinomial NB due to the under-studied bias [5]. In Complement NB, a complement class formulation is used to balance the training instances used per estimate [5].

In the next section, I will use Complement NB (CNB) for experiments due to its highest accuracy rate.

3.2.3 Multilayer Perceptron

MLP model is more computational complex than NB model and we expected it would perform better. In section 4, we will discuss how to build the best MLP model for our movie classification task.

4 Experiment

In this section, we will design and implement some experiments to evaluate the hypotheses we made in section 1.

4.1 Influence of Attribute

We had tried different attributes combinations in our CNB and MLP model to find out which attribute is dominant. The validation set's accuracy of each case is shown in Table 3.

Model and Attributes	*Accuracy*
CNB with only tags	37.46 %
CNB with tags, visual, audio	37.79 %
CNB with only visual, audio	18.73 %
MLP with only tags	39.80 %
MLP with tags, visual, audio	40.47 %
MLP with only visual, audio	31.43 %

Table 3- Influence of Attribute

In this experiment, the MLP model's architecture was all fixed and TF-IDF was implemented on tag and title attributes.

4.2 MLP with Different Parameters

In this section, we would try different activation functions, solver, hidden layer architecture to compare the accuracy under different conditions. To gain consistent results, we feed in the same training and valid dataset with the same attributes and encoding, and only change one variable at one time.

4.2.1 Solver

We tried two different solvers: *sgd* and *adam*. The *sgd* solver uses stochastic gradient descent for optimization; and the *adam* solver is stochastic gradient descent with adaptive estimates of the lower-order moment [6]. The *adam* algorithm can computer individual adaptive learning rate for each attribute and usually work well on a large dataset [6].

These two MLP are both trained by using tag, visual, audio attributes, and with one hidden layer with 100 units, logistic activation function, and initial learning rate = 0.5.

We plotted the accuracy of training and validation set for visualization (Figure 4,5).

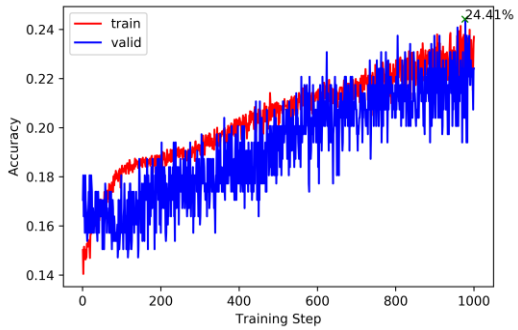

Figure 4a- MLP with *sgd* Solver (partial)

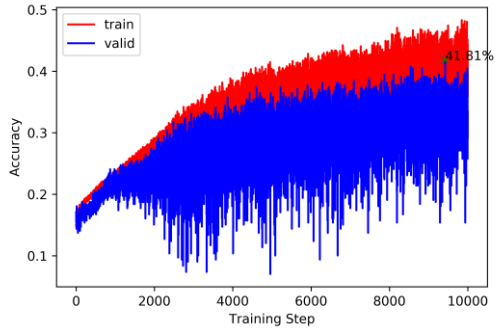

Figure 4b- MLP with *sgd* Solver

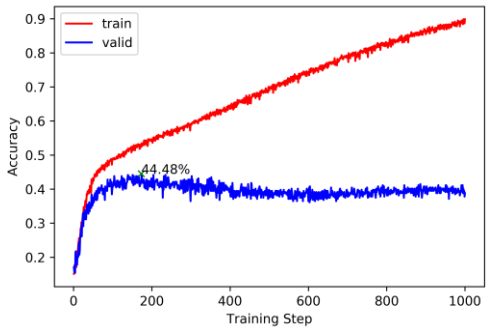

Figure 5- MLP with *adam* solver

As we can see in Figure 4, the accuracy of the validation set of *sgd* model fluctuated a lot and the training set converged slowly. In contrast to *sgd*, Figure 5 shows that the accuracy of the training set of *adam* model **converge faster** and the accuracy of the validation set is more **stable**. This agrees with our expectation that the *adam* solver works better on the large dataset. Therefore, we will use *adam* solver in the following tests.

4.2.2 Activation Function

In this section, all the MLP model are trained with the same parameters except the activation function. Four activation functions are discussed.

- identity: f(x) = x
- relu: f(x) = max(0,x)
- logistic: f(x) = 1 / (1 + exp(-x))
- tanh: f(x) = tanh(x)

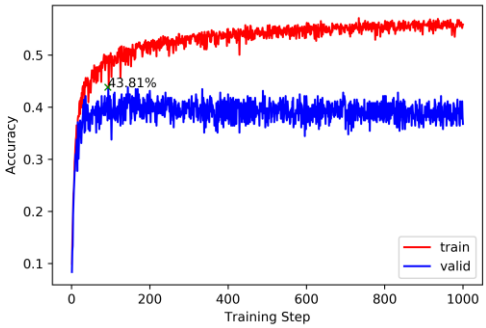

Figure 6- MLP with identity activation function

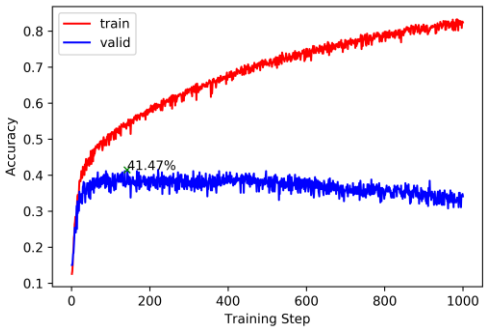

Figure 7- MLP with relu activation function

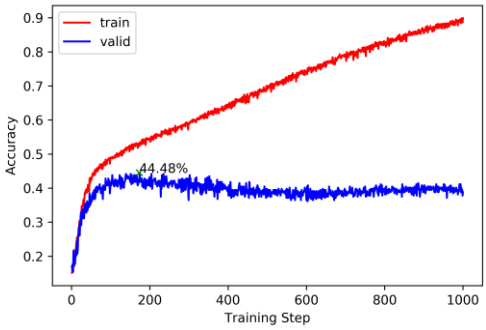

Figure 8- MLP with logistic activation function

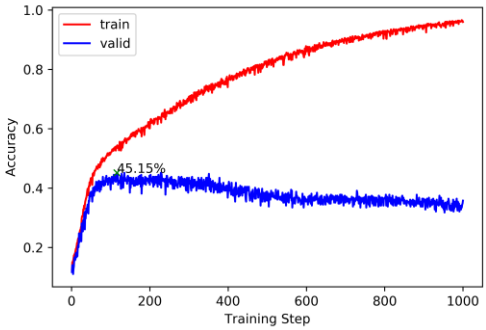

Figure 9- MLP with tanh activation function

As we can see in Figure 6 to 9, the identity activation function performed the worst. This meet the expectation that the **linear** identity activation could not perform non-linear transform on each layer; therefore, can not perform well on our non-linearly separable dataset.

The logistic and tanh activation functions perform relative well due to their high non-linear characteristic.

Moreover, we could observe in Figure 7 to 9, when we train our MLP models for too many steps, they tend to **overfit**; this leads to bad accuracy on the validation set. Among those three activations function, the model with logistic activation function has the most **stable** validation accuracy. Therefore, in the following tests, we will use logistic as activation function to minimize the overfitting effect.

4.2.3 Hidden Layer Architecture

In this test, we only change the hidden layer parameter. Logistic function and *adam* solver are used in all MLP model. Hidden layer parameter is expressed in parentheses, each number in parentheses represented the number of units in that layer. For example, (100,20,10) represents that there are 3 hidden layers with 100, 20, and 10 units in each of them.

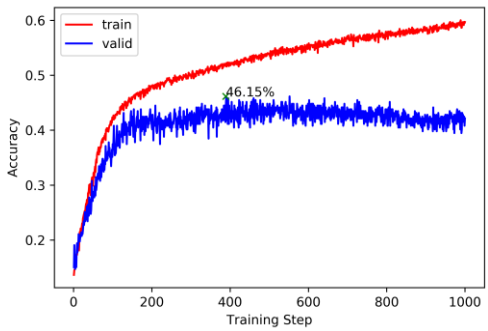

Figure 10- MLP with (20,) Hidden Layer

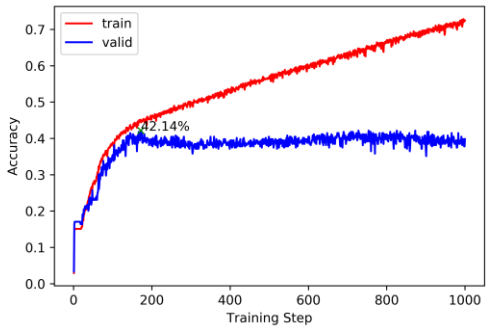

Figure 11- MLP with (40,20) Hidden Layer

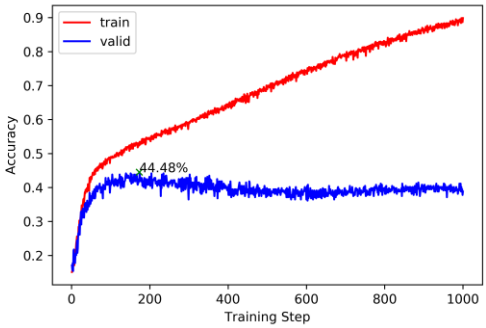

Figure 12- MLP with (100,) Hidden Layer

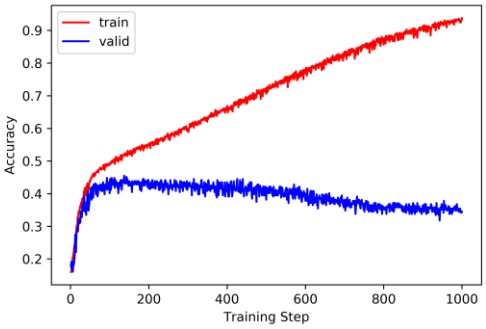

Figure 13- MLP with (144,) Hidden Layer

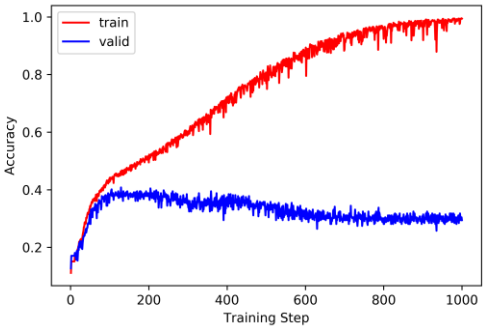

Figure 14- MLP with (144,72,36) Hidden Layer

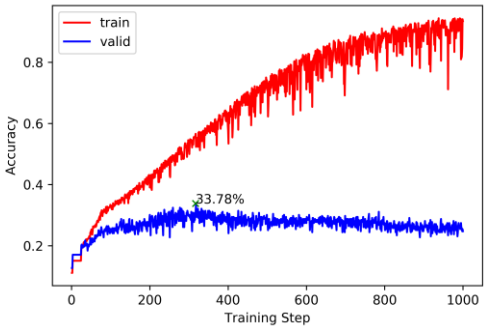

Figure 15- MLP with (200,100,50,25) Hidden Layer

As we can see in Figure 10 to 15, the more complex hidden layer architecture would leaded to faster convergence of train data set (the training accuracy increased faster); however, it also made the MLP model overfit faster (the valid accuracy decreased faster) and fluctuate more. There is a trade-off between converge speed and overfitting. Therefore, we chose (100,) as our hidden layer architecture because it could reach the max valid value soon and is not subjected to overfitting too much.

4.3 MLP with Different Attributes

In previous sections, all the MLP models were trained by using tag, audio, visual attributes. In this section, we would discuss could title attributes help the movie classification or not.

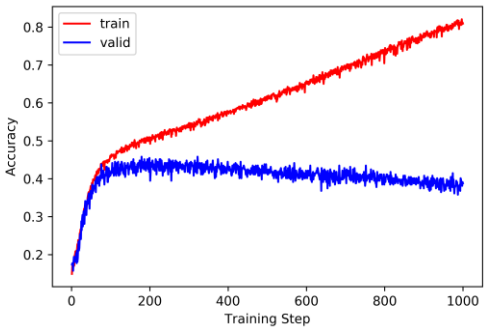

Figure 16- TF-IDF with Tag, Visual, Audio Attributes

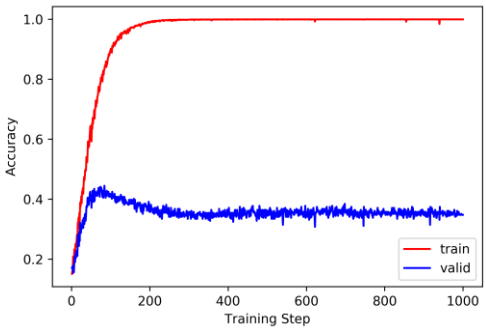

Figure 17- TF-IDF with Tag, Title, Visual, Audio Attributes

In Figure 16 and 17, these two MLP models were trained with the same parameters as previous sections indicate. Both MLP models used TF-IDF to encode text type attributes. The only difference is that in Figure 17, we also use the title attribute for training. We could observe that, with the TF-IDF encoded title attributes, the accuracy of training set increases much more quickly; however, the overfit situation also more severe, and the overall valid performance

was not improved.

In the TF-IDF encoding, the title attributes were extracted from the training set, and we dropped the words with less than three alphabets; nevertheless, there were still more than 6000 title attributes. It was trivial so it made the model overfit. In contrast, there were only 208 tag attributes, which could represent the genres better.

Moreover, due to the large number of title attributes, adding title attributes for training led to long training time. For example, training with title attributes for 1000 steps took 2961 secs, while training with tag attributes for 1000 steps only took 355 secs.

From the time and memory efficiency aspect and the validation performance, we could conclude that adding the title attributes would not make the model better.

5 Analysis & Discussion

5.1 Hypothesis I Evaluation

As we could see in section 4.1, the CNB and MLP model with only visual and audio attributes have low accuracy. Especially in CNB, the accuracy is only 18.73%, which is close to the baseline Zero-R model. This points out that, the audio and visual attributes are not the dominant attribute.

When we added the "tag" attributes for training, the accuracy of CNB had a dramatic improvement. And, even with only tag attributes, the CNB could perform almost the same well. This points out that the "tag" attribute is the dominant attribute. Base on the accuracy with "tag", visual, and audio attributes, there is only a 6.62% performance difference between CNB and MLP.

This satisfies our hypothesis 1, when there is a dominant categorical attribute (i.e. "tag"), the performance of NB model would not be too bad compared with MLP.

5.2 Hypothesis II Evaluation

Look at section 4.1 again, Table 3 shows both CNB and MLP's accuracy are increased when the visual and audio attributes are added into training set. Although the incremental is not noticeable, we could conclude that the visual and audio attributes can indeed help classify the movie genres. This proves that the MMTF-14K [1] is more powerful for movie genre prediction than other traditional movie databases.

5.3 Hypothesis III Evaluation

From section 4.2, we could conclude that the best MLP model for this task is:

- Using tags, audio, visual attributes for training
- Using *adam* solver
- Using one hidden layer with 100 units is enough
- Using logistic activation function

Therefore, more complex MLP would not lead to better performance on the validation set.

6 Conclusions

In this report, we have proved that when the dataset is dominated by one categorical attribute, the NB classifier could achieve similar performance to MLP. We also showed that the audio and visual attributes could indeed improve the model performance on both the training set and validation set. Moreover, we found one of the best MLP model's configurations and proved that over simplified or complicated would not lead to good performance.

References

[1] Y. Deldjoo, M. G. Constantin, B. Ionescu, M. Schedl, and P. Cremonesi, "MMTF-14K: a multifaceted movie trailer feature dataset for recommendation and retrieval," presented at the Proceedings of the 9th ACM Multimedia Systems Conference, Amsterdam, Netherlands, 2018. [Online]. Available: https://doi.org/10.1145/3204949.3208141.

[2] F. M. Harper and J. A. Konstan, "The MovieLens Datasets: History and Context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, p. Article 19, 2015, doi: 10.1145/2827872.

[3] F. Pedregosa, "Scikit-learn: Machine learning in Python," *The Journal of Machine Learning Research*, vol. 12, p. 2825, 2011. [Online]. Available: HTTP://unimelb.hosted.exlibrisgroup.com/sfxlcl41?sid=google&auinit=F&aulast=Pedregosa&atitle=Scikit-learn%3A%20Machine%20learning%20in%20Python&title=The%20Journal%20of%20Machine%20Learning%20Research&volume=12&date=2011&spage=

2825&issn=1532-4435.
[4] "1.9. Naive Bayes — scikit-learn 0.23.1 documentation." https://scikit-learn.org/stable/modules/naive_bayes.html#naive-bayes (accessed.
[5] J. D. Rennie, L. Shih, J. Teevan, and D. R. Karger, "Tackling the poor assumptions of naive bayes text classifiers," in *Proceedings of the 20th international conference on machine learning (ICML-03)*, 2003, pp. 616-623.
[6] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv*, 2014/12/22/ 2014. [Online]. Available: https://arxiv.org/abs/1412.6980.