COMP90049 Machine Learning
Predicting Movie Genres

Ian Chuah

1 Introduction

The digital revolution has permanently transformed the way media is consumed. It has enabled websites to host large amounts of videos, and equipped consumers with the means to stream movies on demand. Manual task of filtering and regulating data, which is expensive and tedious can no longer keep up with the exponential proliferation of new content. This calls for a need to implement an automated filtration system to select relevant data based on user's preferences. Such recommender systems were utilized to great effect in the video services industry, such as Netflix and Youtube, (Makita & Lenskiy, 2016) where videos of interest are tailored to users based on their browsing habits. For these services, an automatic content-based movie classifier can be useful. For instance, videos with disturbing content are immediately flagged and removed; violent movies are detected and blocked from children.

In the following sections, we examine the suitability of several models in building a movie classifier from a given training dataset, namely Naïve Bayes, Multi-Level Perceptron and Zero-R as our baseline model. Section 2 discusses past studies conducted by researchers. In Section 3 we describe our approach for movie genre classification, in Section 4 we conduct performance study. Finally, a summary of work with some suggestions is provided in section 5.

2 Related Work

The study of classifying videos using ML algorithms were rather well researched, and successfully demonstrated in the past. The strategies employed varied from paper to paper, ranging from using low-level features that comprises the visual features, such as average shot length, color variance and lighting key by (Huang, Shih & Hsu 2007) and (Rasheed & Shah 2002) using 2-layer Neural network and Mean Shift Classification algorithm respectively. (Ji, Yang & Yu 2013) employed a 3D CNN model to interpret stacked video frames, an approach that stemmed from a conventional 2D CNN used for pictures. Others incorporated additional features such as audio features, as demonstrated by (Huang & Wang, 2012) using a SVM algorithm.

From the approaches above, there appears to be an emphasis on the real movie content, features which are real content features derived from the raw elements of the data, as opposed to the claim made by (Deldjoo, Constantin, Ionescu, Schedl & Cremonesi, 2018). Further, the papers are written prior to 2018, before Deldjoo's paper was published. It could be argued that metadata is just as important, if not more. We will later motivate this statement below, in related sections.

2.1 Dataset Description

For the purpose of this report, we will be using the dataset courtesy of (Deldjoo, Constantin, Ionescu, Schedl & Cremonesi, 2018) and (Harper & Konstan, 2015) containing textual and numerical attributes. This dataset contains real content features, which contains audio and visual features of movies, and pseudo content features, such as tags and Youtube links to movie trailers. It has in total 5240 videos, with numerical preprocessed audio and visual features, and textual metadata, comprising a total of 127 features. The target value is the genre, consists of 18 distinct labels, ranging from mainstream genres such as action, comedy, and romance to obscure ones such as war, sci-fi and western.

3 Proposed method

Upon initial observation, the biggest dilemma faced is to decide on what model to use due to the presence of a mixed-valued features. The multi-class classification problem means that traditionally binary classifiers such as Logistic Regression or Support Vector Machines (SVM) are unsuitable without extrapolating them to account for multi-dimensional problems. Thus, we use classification algorithms that can produce multi-dimensional outputs such as Naïve Bayes (NB) and Feedforward Neural Network. (FFNN)

3.1 Zero-R

Under the Zero-R decision rule, the model outputs the majority class in a dataset. It ignores the predictors (target) and make predictions based solely on the target. This trivial approach provides a good starting point for evaluating our main classifier performance, making it an effective baseline.

3.2 Generative Naïve Bayes (NB)

Naïve Bayes scales linearly with number of instances and has a simple learning method. The underlying conditional independence assumption of Naïve Bayes lends robustness to its classification model. A generative multinomial

Naïve Bayes is used here because it can classify an instance even if said instance has no common features with the training data. This makes it useful for classifying textual features, but not for numerical features in our dataset, obviously so because we can count words to predict class/label. To compensate for this, we use the gaussian version for the numerical features. However, the Gaussian model will only yield good estimates if the dataset closely resembles a normal distribution.

Under this model, the performance improves as the number of input features and training samples increases. The classifier's predictive ability increases as the occurrence of observing an instance in the absence of features decreases.

3.3 Feedforward Neural Network (FNN)

A FFNN network, depending on the depth and width, can become algorithmically complex easily. Thus a Single Layer Feedforward Neural Network (SLFN) with a sigmoid activation function for the hidden layer and softmax activation at the output is used to reduce any possible time and storage related complications from arising. One layer is sufficient (Huang, Chen & Babri, 2000). For a problem of this complexity level, the dataset is small, so a shallow neural network is used to prevent overfitting. As the dataset increases, neural network tends to generalize better.

4 Experimental Results and Discussion

4.1 Data

Performing an initial analysis of the underlying distribution of the training set yielded the results below:

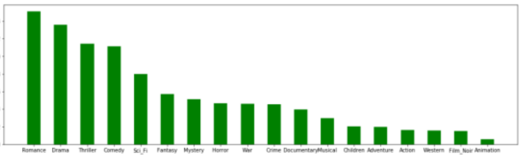


Figure 1 – Film distribution by genres from training set

Compared to a plot of global film genre production:

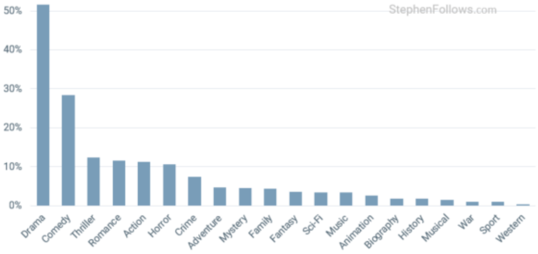


Figure 2 – Film distribution by genres (Follows 2018)

To simplify our analysis, it is assumed that **Figure 2** is an accurate representation of the true film genre distribution, and ideally our training data distribution should closely resemble it. Looking at the majority genres in training set, we have popular genres such as "Romance", "Drama" and "Thriller", in line with **Figure 2**. On the opposite spectrum, we have "Western", "Film-noir", and "Animation." **Figure 2** suggests that the least represented genres are "War", "Western" and "Musical" (we are neglecting "Sport" due to missing label in **Figure 1**). We see that the **Figure 1** and **2** matches closely in the majority label but drifts apart towards the middle and lower part of the distribution, implying that training set is biased. Thus, some underfitting may occur. The performance may be poorer for less represented movies and better for popular ones.

4.2 Feature Selection and Engineering

Two different ways of feature selection are explored here – First, we will use only the textual features, followed by a purely numerical feature set. Textual features are a better indicator for model performance because it does not suffer from curse of dimensionality. (Domingos 2012)

Some features may be useless for prediction. By manual inspection, we observe that features such as the Youtube Links, and movieId are just arbitrary assigned identification numbers that has no relation to our movie at all, so they are omitted.

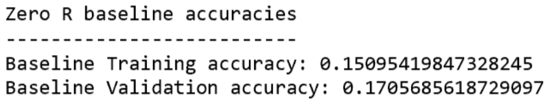
To improve the performance of our classifier, the features are vectorized using the word count and TF-IDF. TF-IDF inherently ranks the importance of word in a document so our models perform better using this method.

4.3 Performance Evaluation

The classifiers first learn the training set, and then tested on the training and validation set to estimate the training and validation accuracy.

4.3.1 Baseline

The performance of our baseline model is outlined below:



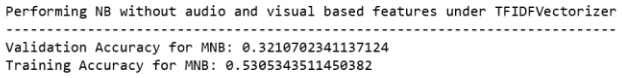
```
Zero R baseline accuracies
--------------------------
Baseline Training accuracy: 0.15095419847328245
Baseline Validation accuracy: 0.1705685618729097
```

Figure 3 – Zero R baseline accuracies

We see that our validation set accuracy is higher than the training accuracy, possibly indicating that the model generalizes well. However, this occurrence is by stroke of luck. A Zero R model by nature does not learn anything, it simply assigns the majority class as its target output. Coincidentally, the majority class established in the training accuracy "Romance" occurs more frequently in the validation dataset.

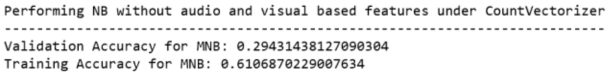
4.3.2 Performance of Vectorizers

I. (with Text based features only)



```
Performing NB without audio and visual based features under TFIDFVectorizer
---------------------------------------------------------------------------
Validation Accuracy for MNB: 0.3210702341137124
Training Accuracy for MNB: 0.5305343511450382
```

Figure 4 – Performance of NB model with textual features under TFIDF Vectorizer

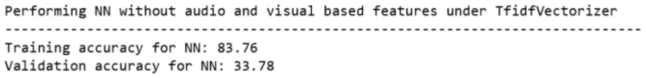


```
Performing NB without audio and visual based features under CountVectorizer
---------------------------------------------------------------------------
Validation Accuracy for MNB: 0.29431438127090304
Training Accuracy for MNB: 0.6106870229007634
```

Figure 5 – Performance of NB model with textual features under Count Vectorizer

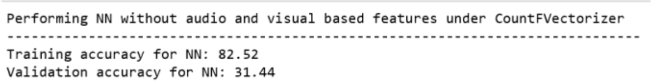
NB performs better when a TFIDF is used compared to a frequency count. One may expect that TFIDF values to be incompatible as NB classifies based on discrete features. This is because TFDIF is a term-weighting method through normalization, which better informs our model about the importance of features as compared to a raw word count.

For the sake of consistency, we only look at the case of a SLFN with fixed hyperparameters for both vectorization methods.



```
Performing NN without audio and visual based features under TfidfVectorizer
---------------------------------------------------------------------------
Training accuracy for NN: 83.76
Validation accuracy for NN: 33.78
```

Figure 7 – Performance of SLFN model with textual features under TFIDF Vectorizer with learning rate = 0.001



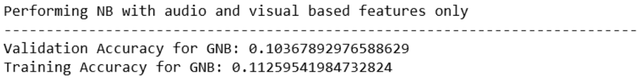
```
Performing NN without audio and visual based features under CountFVectorizer
---------------------------------------------------------------------------
Training accuracy for NN: 82.52
Validation accuracy for NN: 31.44
```

Figure 8 – Performance of SLFN model with textual features under TFIDF Vectorizer with learning rate = 0.001

Just like in the NB model, our SLFN model performs better when TFDIF vectorization is used. This is because TFIDF flags out the keywords that may be a strong indicator for the type of genre. Take the movie 'Happy Gilmore' for example, the tag 'hilarious' is a strong indicator of it being a Comedy, which is very unlikely to be found in most other non-comedy movie tags.

II. (with Numerical features only)

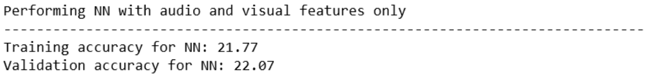
Since this involves continuous input variables, we use the Gaussian flavor of NB model:



```
Performing NB with audio and visual based features only
---------------------------------------------------------------------------
Validation Accuracy for GNB: 0.10367892976588629
Training Accuracy for GNB: 0.11259541984732824
```

Figure 9 – Performance of NB model with numerical features only (audio + visual)

We see that Gaussian NB model performs extremely poorly, failing to even clear the baseline. The model uses a normal approximation to estimate the target value, the closer the data resembles a normal distribution, the better the performance. As such, our Gaussian approximation provides a bad estimation because the numerical data is not normally distributed. Also, the values of the features dictate the type of sounds and images produced. As an arbitrary example, an audio feature with combination value [1,1,1] may produce a chirping sound, while [1,2,1] generates a percussion sound. In the context of movie scores, movies of similar genre utilize similar sound sequence to establish a desired mood that they want the audience to feel. (Zettl, 2017). In this case, it is obvious that the features are highly dependent, which violates the fundamental assumption of a NB model.

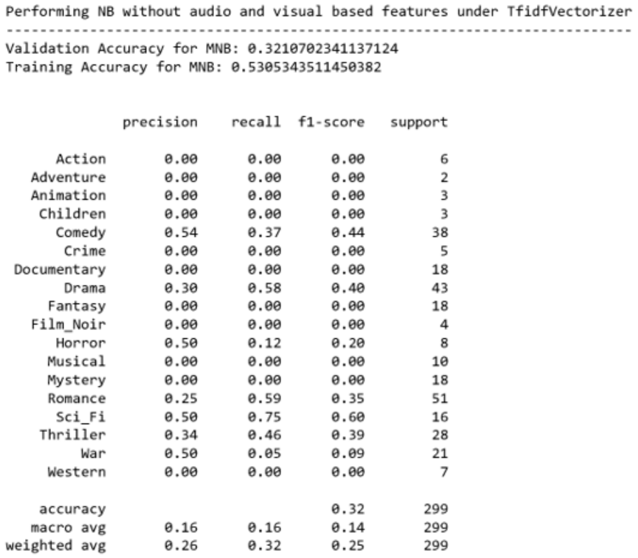


```
Performing NN with audio and visual features only
---------------------------------------------------------------------------
Training accuracy for NN: 21.77
Validation accuracy for NN: 22.07
```

Figure 10 – Performance of SLFN model with numerical features only (audio + visual) with learning rate = 0.001

The performance rating of SLFN dropped too, reinforcing the fact that textual feature classification far outweighs its numerical counterpart.

4.3.3 Classification Report



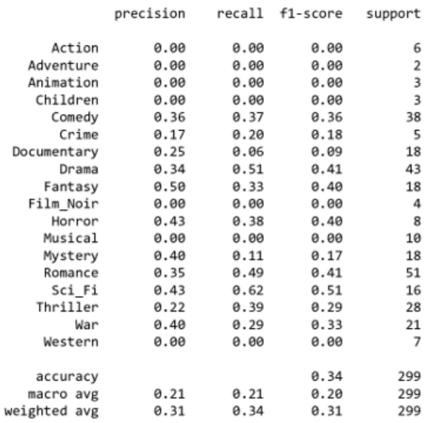
```
Performing NB without audio and visual based features under TfidfVectorizer
---------------------------------------------------------------------------
Validation Accuracy for MNB: 0.3210702341137124
Training Accuracy for MNB: 0.5305343511450382


              precision    recall  f1-score   support

      Action       0.00      0.00      0.00         6
   Adventure       0.00      0.00      0.00         2
   Animation       0.00      0.00      0.00         3
    Children       0.00      0.00      0.00         3
      Comedy       0.54      0.37      0.44        38
       Crime       0.00      0.00      0.00         5
 Documentary       0.00      0.00      0.00        18
       Drama       0.30      0.58      0.40        43
     Fantasy       0.00      0.00      0.00        18
   Film_Noir       0.00      0.00      0.00         4
      Horror       0.50      0.12      0.20         8
     Musical       0.00      0.00      0.00        10
     Mystery       0.00      0.00      0.00        18
     Romance       0.25      0.59      0.35        51
      Sci_Fi       0.50      0.75      0.60        16
    Thriller       0.34      0.46      0.39        28
         War       0.50      0.05      0.09        21
     Western       0.00      0.00      0.00         7

    accuracy                           0.32       299
   macro avg       0.16      0.16      0.14       299
weighted avg       0.26      0.32      0.25       299
```

Figure 11 – Final Results for NB model



```
              precision    recall  f1-score   support

      Action       0.00      0.00      0.00         6
   Adventure       0.00      0.00      0.00         2
   Animation       0.00      0.00      0.00         3
    Children       0.00      0.00      0.00         3
      Comedy       0.36      0.37      0.36        38
       Crime       0.17      0.20      0.18         5
 Documentary       0.25      0.06      0.09        18
       Drama       0.34      0.51      0.41        43
     Fantasy       0.50      0.33      0.40        18
   Film_Noir       0.00      0.00      0.00         4
      Horror       0.43      0.38      0.40         8
     Musical       0.00      0.00      0.00        10
     Mystery       0.40      0.11      0.17        18
     Romance       0.35      0.49      0.41        51
      Sci_Fi       0.43      0.62      0.51        16
    Thriller       0.22      0.39      0.29        28
         War       0.40      0.29      0.33        21
     Western       0.00      0.00      0.00         7

    accuracy                           0.34       299
   macro avg       0.21      0.21      0.20       299
weighted avg       0.31      0.34      0.31       299
```

Figure 12 – Final Results for SLFN model

From above, our NB model unexpectedly perform poorly on several popular categories like Drama and Romance. Hypothetically, it should perform better in those categories given that there are more target labels to learn from. One possible explanation is that using the TFIDF vectorization method, the contribution of higher frequency words towards predictions is diminished due to normalization, causing underfitting to occur. The absence of variety in our predictions in NB model indicates so. An interesting observation made is that both models score well when it comes to predicting Sci-Fi genre. This is due to the textual features that are associated with Sci-Fi are very distinct, thus easily distinguishable. For example, in the movie "The Matrix" the presence of a word like "computer" strongly points to a Sci-fi genre.

Comparing the average precision and recalls across both models, the SLFN model out-performs NB. This is because the vectorization introduced additional dimensions of our dataset. The big data generated is very appropriate as training inputs for our neural network.

5 Conclusion and Future Work

It was identified that using only textual features is far superior to numerical features in movie genre classification, even though it makes up of 11% of our dataset. Feeding relevant features is more important than more features in improving model performance. Our experiments show that a single layer neural network is better than Multinomial Naïve Bayes model. To use the audio and visual features to great effect, relevant domain knowledge in image and audio processing is recommended to devise methods that can quantitatively analyze them, as demonstrated in Section 2. For future research, K cross validation could be used to obtain a better approximation of the model performance. Also, explore options such as using a stacked ensemble or deeper learning models to obtain improved results.

6 References

Makita, E., & Lenskiy, A. (2016). A Multinomial Probabilistic Model for Movie Genre Predictions. International Journal Of Machine Learning And Computing, 6(2), 97-100. doi: 10.18178/ijmlc.2016.6.2.580

H. Huang, W. Shih & W. Hsu, (2007). Movie Classification Using Visual Effect Features, *2007 IEEE Workshop on Signal Processing Systems*, Shanghai, China, 2007, pp. 295-300, doi: 10.1109/SIPS.2007.4387561.

Deldjoo, Y., Constantin, M., Ionescu, B., Schedl, M., & Cremonesi, P. (2018). MMTF-14K. Proceedings Of The 9Th ACM Multimedia Systems Conference. doi: 10.1145/3204949.3208141

Ji, S., Yang, M., & Yu, K. (2013). 3D convolutional neural networks for human action recognition. IEEE transactions on pattern analysis and machine intelligence, 35(1), 221–231. https://doi.org/10.1109/TPAMI.2012.59

Harper, F.M., & Konstan, J.A. (2015). The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.*, *5*, 19:1-19:19.

Domingos, Pedro. (2012). A Few Useful Things to Know About Machine Learning. Commun. ACM. 55. 78–87. 10.1145/2347736.2347755.

Follows, S. (2018). Genre trends in global film production. Retrieved 21 May 2020, from https://stephenfollows.com/genre-trends-global-film-production/

Z. Rasheed & M. Shah (2002). Movie genre classification by exploiting audio-visual features of previews, *Object recognition supported by user interaction for service robots*, Quebec City, Quebec, Canada, 2002, pp. 1086-1089 vol.2, doi: 10.1109/ICPR.2002.1048494.

Huang YF. & Wang SH. (2012) Movie Genre Classification Using SVM with Audio and Video Features. In: Huang R., Ghorbani A.A., Pasi G., Yamaguchi T., Yen N.Y., Jin B. (eds) Active Media Technology. AMT 2012. Lecture Notes in Computer Science, vol 7669. Springer, Berlin, Heidelberg

Guang-Bin Huang, Yan-Qiu Chen, & Babri, H. (2000). Classification ability of single hidden layer feedforward neural networks. IEEE Transactions On Neural Networks, 11(3), 799-801. doi: 10.1109/72.846750

Zettl, H. (2017). Sight Sound Motion, Applied Media Aesthetics (8th ed.). Cengage

(Word Count: 2240 words incl. Tables and References)

Z. Rasheed & M. Shah (2002). Movie genre classification by exploiting audio-visual features of previews, *Object recognition supported by user interaction for service robots*, Quebec City, Quebec, Canada, 2002, pp. 1086-1089 vol.2, doi: 10.1109/ICPR.2002.1048494.

Huang YF. & Wang SH. (2012) Movie Genre Classification Using SVM with Audio and Video Features. In: Huang R., Ghorbani A.A., Pasi G., Yamaguchi T., Yen N.Y., Jin B. (eds) Active Media Technology. AMT 2012. Lecture Notes in Computer Science, vol 7669. Springer, Berlin, Heidelberg