MOVIE GENRE CLASSIFICATION

George Abraham, Arinjay Jain, Ayub Sharif
Northeastern University
jain.arinj@northeastern.edu, abraham.g@northeastern.edu, sharif.a@northeastern.edu

Abstract

With the amount of new movies coming out, the task of organizing all the data becomes difficult. With the classification of the genre of the movie, we can organize them easily and recommend them to users on that basis if needed. Our method requires either the poster of the movie or synopsis of the movie, the classification itself is being done using different algorithms on the IMDB dataset and IMDB poster dataset from Kaggle. We determine which of these methods is most accurate for the problem of classifying the movies.

Introduction

Movie genres are a way of categorizing movies into a set of categories that have common traits. These classifications are usually done manually by an expert. There is a lot of subjectivity when it comes to classifying movies, and two experts might even classify them in different categories. Even in our dataset, there were some movies that fit within joint categories such as 'action, drama' and not in either 'action' or 'drama'. The task is exponentially more complicated when you take into consideration the thousands of movies that streaming services deal with. It is beyond the capabilities of just a set of expert reviewers. We believe that this is a perfect application area for deep learning algorithms. There have been other works that used deep learning to classify movie genres, but they didn't do much comparisons between different algorithms, deep learning or not, and selecting the best one to perform this type of analysis.

Plot summaries are a useful tool used to predict the genre of a movie. It is what people read to get a sense of what the movie is about before deciding to even watch the movie. Given the descriptive information, summaries are a rich source to mine when predicting the genre of a movie. Another useful tool when garnering viewer interest in a movie is the poster. Good posters are often the primary element that captures the viewer's interest. It is in the best interest of designers to create posters that can accurately convey the theme, settings and other pertinent details to the prospective viewers. In our project, we also hope to explore if certain elements of a movie poster can be used to determine its genre.

Our project uses plot summaries as an input in two of our algorithms whereas posters are used in one of them. We evaluate deep learning algorithms to correctly classify the genres of the movie based on these inputs.

Background

In this section, we will briefly provide contextual information to better understand the models used in our project.

Logistic Regression

Logistic regression (Kleinbaum et al. 2002) is a statistical model that in its basic form uses a logistic function to model a dependent variable. At its core, it uses the logistic function to find a model that fits with the data points. This can be extended to model several classes of events such as determining whether a movie's genre is drama, action, romance etc.

Naive Bayes Classifier

These are a family of probabilistic classifiers based on applying Bayes' theorem with strong (naïve) independence assumptions between the features.

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)}$$

Support Vector Classifiers

These classifiers are non-probabilistic in nature that classify given data by separating them into hyperplanes. It usually is used for binary classification, but it can be extended for multiple classes as well (as done for this project).

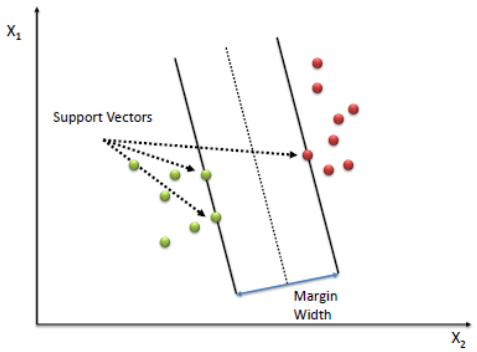

Figure 1: A SVM classifier with margin width and the decision boundary

Convolutional Neural Networks

Convolutional neural Networks are mostly used for image recognition and classification. These often consist of separate channels stacked on top of each other.
A classic CNN consists of:

- Convolutional layer:

Pixel data is convolved using filters and kernels. The filters slide across the pixel matrix taking the element wise product of filtering the image and summing the values.

- Activation Layer:

This layer is used to increase the non-Linearity in the CNN. Activation functions used include Relu, Tanh and Sigmoid.

- Pooling Layer:

This layer is used to down sample the features of the previous layer.

- Fully Connected Layer:

This is also called Flattening as the feature matrix is converted to a column before being fed to a neural network.

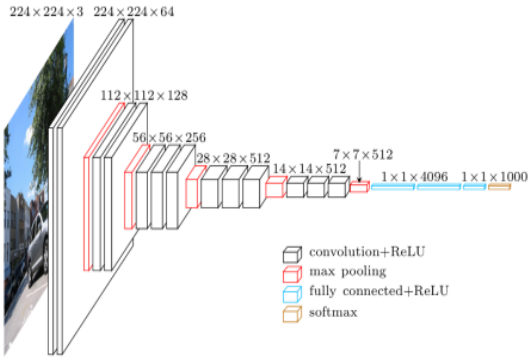

Figure 2: VGG16 Architecture (Example of CNN)

Training:
During the first pass, the model is initialized with random values for the layers.
The loss is then calculted using the mean square error formula:

$$MSE = \frac{i}{n}\sum_{i=1}^{n}(y_i - y'_i)$$

To minimize the loss, we must adjust the weights. The weights of the kernel are updated using the formula:

$$W'_x = W_x - a\left(\frac{\partial Error}{\partial W_x}\right)$$

Where W'_x is the new weight, W_x is the old weight, a is the learning rate and the differential term is the derivative of error with respect to weight.
Here the learning rate is chosen by the programmer.

Models used and compared are

Custom model

This model consists of 4 sets of Convolution layers followed by a maxpooling layer and features dropoff to reduce overfitting of data.

VGG16 model

The 16 in VGG16 refers to it having 16 layers that have weights. The VGG16 Architecture was developed and introduced by Karen Simonyan and Andrew Zisserman from the University of Oxford, in the year 2014.

VGG19 model

This is a variant of the VGG model featuring 19 layers (16 convolutional, 3 Fully connected, 5 max pool layers and 1 soft max layer).

Resnet50 model

This model features 48 convolution layers followed by 1 max pool and 1 average pool layer. This architecture features skip connections to reduce the effect of vanishing gradients in large networks.

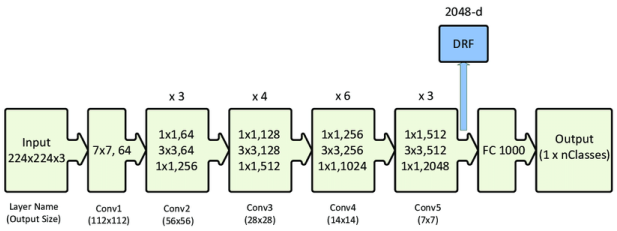

Figure 3: Resnet50 Architecture

InceptionV3

This model consists of symmetric and assymetric blocks of convolutional, maxpooling, average pooling and dropout layers. Batchnorm is used throughout the model.

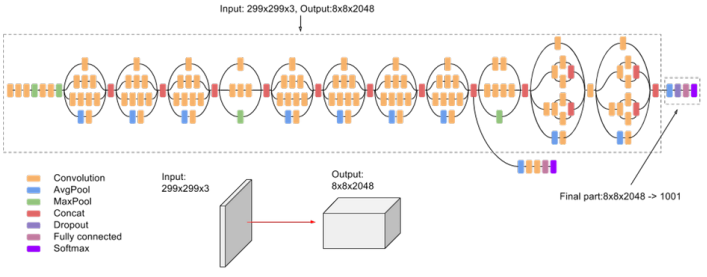

Figure 4: InceptionV3 Architecture

Long Short Term Memory

LSTMs are a type of recurrent neural network that is also capable of storing long term persistent information. A recurrent neural network can be used for persistent memory, but they cannot remember long term dependencies due to the vanishing gradient problem. The vanishing gradient problem occurs when training ML algorithms through gradient descent. As one goes deeper into the network, partial derivatives are used to compute the gradient. Gradients are responsible for how much the network learns during training, and therefore its performance. If the gradients are small or zero, then there is not enough training. LSTMs are designed to avoid long-term dependency problems. Here is a picture of a sample LSTM architecture.

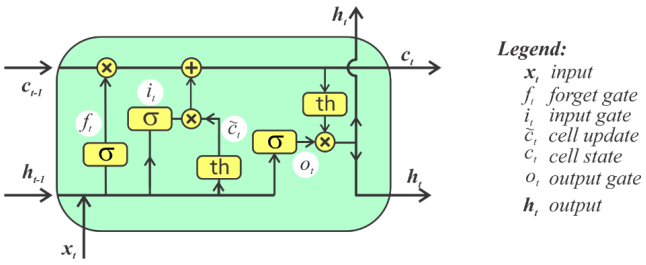

Figure 4: LSTM Architecture

Related Work

There have been multiple implementations of genre classification using deep learning. The majority of which are focused on music genre classification. They used spectrograms of the audio file as the input to the CNN for this task. Some literature focused on genre classification of movies based on non-visual material such as using the reviews and plot summaries. Classification based on movie posters is a less approached topic probably due to the difficulties associated with multi-label classification.

The authors in Barney et al. tested several methods for the classification including a ResNet 34 and a custom CNN architecture. The authors Chu et al. designed an architecture using CNN and YOLO. From the results obtained it would seem that the models used in Barney et al. are the state of the art in this problem. We seek to build up and compare the models used in this paper.

There has been a closely related previous attempt using LSTM to classify movie genres. Ertugrul and Karagoz from Middle East Technical University in Ankara, Turkey used bidirectional LSTM (Bi-LSTM) in their method for classification (Ertugrul and Karagoz 2018). They divided plot summaries into sentences and assigned the genre based on each sentence. This was used to train the Bi-LSTM network. They compared Bi-LSTM to basic Recurrent Neural Networks and Logistic Regression models. We didn't use a Bi-LSTM. We conducted our analysis using a basic LSTM model.

Project Description

Here, we formulate our task of genre classification:

Given a movie M, and its associated data in different modalities,X_i, using these modalities X_i independently or as a combination, predict the Genre of the movie M i.e. Y.

If this task was given to an expert human annotator, they might be able to classify the movie M using the domain knowledge about movie genres, but it would be a highly manual and time-consuming process. Our method involves using different machine learning and deep learning algorithms to see which method is the most accurate and fast for the classification of movie genres.

Data Sets

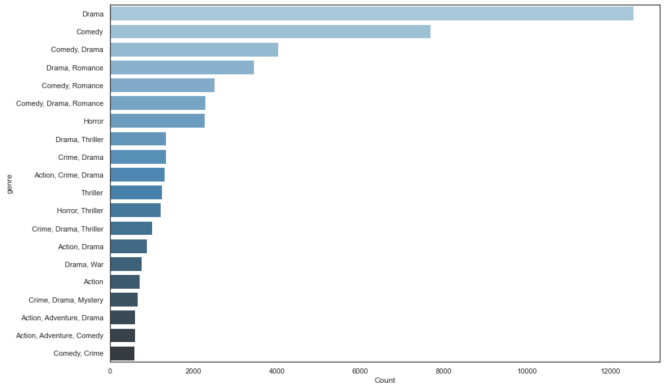

Figure 5: Initial Exploration of IMDB movie dataset for synopsis

The Kaggle Poster dataset consisted of about ~36000 posters. The Train/Validation/Training dataset split was ~ 10500/9500/1000.

The total number of unique genres was 28. And each movie was classified to up to 3 genres. The genres include: Action, Adult, Adventure, Animation, Biography, Comedy, Crime, Documentary, Drama, Family, Fantasy, Film-Noir, Game-Show, History, Horror, Music, Musical, Mystery, News, Reality-TV, Romance, Sci-Fi, Short, Sport, Talk-Show, Thriller, War and Western.

Figure 6: Example poster from the dataset (Action/Sci Fi)

Experiments

LSTM Experiments

We ran three experiments in our LSTM model. For all of our experiments, we ran the LSTM model on a training dataset and a test dataset. We got graphs that showed the accuracy of the model based on these two datasets as well as the losses from the model. Accuracy helps measure how well the model performs and is expressed as a percentage. It includes the number of predictions that were equal to the actual value. Accuracy is an easy way of interpreting performance. Loss, on the other hand, takes into consideration the variation of the predicted value and actual value. It sums off these errors for each training or testing set. It is not as easy to interpret as the accuracy graph, but definitely provides another layer of telling the performance story. In each of the different experiments, we changed a condition to see how much it would impact the model performance. In the first experiment, instead of looking at all the genres. We chose only the single category genre because we thought the model would do a better job handling this dataset as it wouldn't have to deal with the ambiguity of deciding between say a joint category. For example it wouldn't have to decide between placing a movie in "romance" or "romance, comedy". It turned out that the accuracy wasn't great. The average accuracy percentage for the test dataset for experiment 1 was 13.16%. The accuracy graph was also very unstable for experiment 1, as a matter of fact, all of the experiments have extreme unstable accuracy graphs but we will get to that later.

In the second experiment, instead of classifying into single category genres, we also included multi-category genres as well. We kept all other model features such as learning rate the same as the first experiment. We found our average accuracy score to rise up to 36.34% when we accounted for multi-category genres.

In our third and final experiment, we chose to classify into single category genres just like the first experiment. In this experiment, we chose to change the learning rate to see if it would create more stability in our accuracy graph. Larger learning rates can be tied to unstable training, so we decided to test a smaller learning rate. We decreased the learning rate from 0.146% to 0.00000146% to see if a smaller learning rate would have more stable accuracy results. It turns out that this wasn't the case. The average accuracy was also worse than experiment 1. In experiment 3, the average accuracy was 6.798% compared to 13.16% in experiment 1.

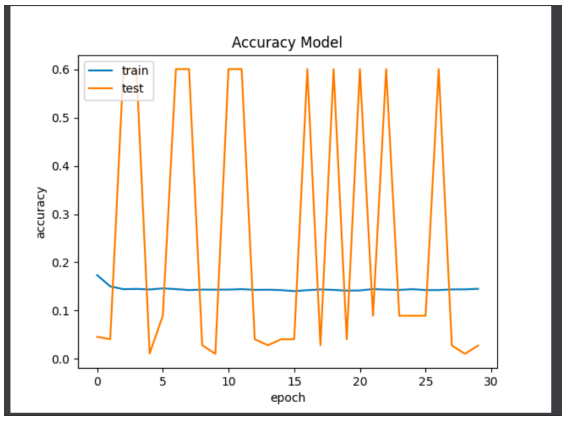

Figure 7: Accuracy model for experiment 1

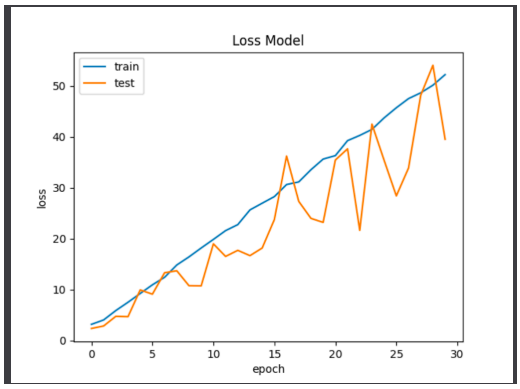

Figure 8: Loss model for experiment 1

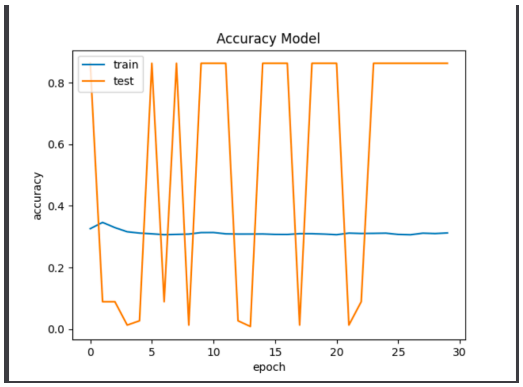

Figure 9: Accuracy model for experiment 2

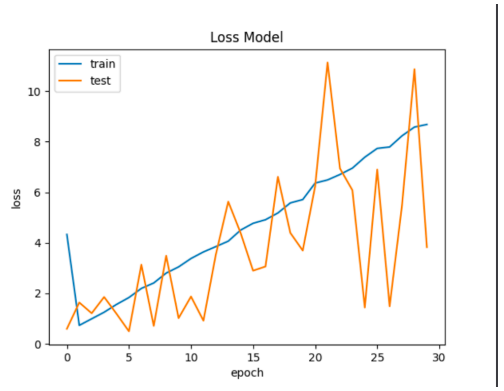

Figure 10: Loss model for experiment 2

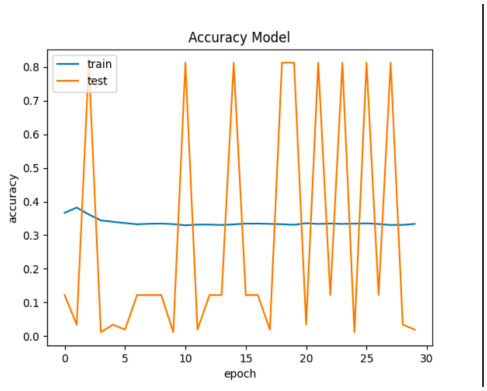

Figure 11: Accuracy model for experiment 3

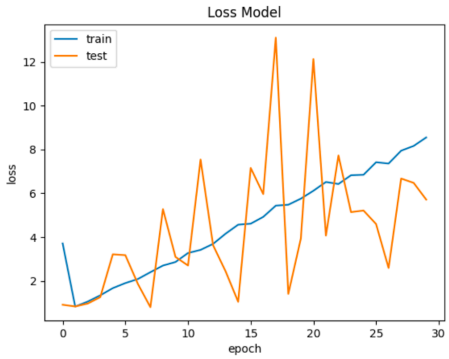

Figure 12: Loss model for experiment 3

Experiment Number	Accuracy
1	0.131600 ± 0.02607
2	0.363408 ± 0.089861
3	0.067983 ± 0.118257

Table 1: Average accuracy scores and their standard deviation obtained after running LSTM experiments.

Convolutional Neural Net Experiments

For training the Poster data set classification model we used a custom CNN architecture using dropoff layers and compared it with other modern models used in image classification (VGG16,VGG19, ResNet50 and InceptionV3). The custom model was trained from scratch while for the other standard models the last 4 layers were restrained while keeping the other layers the same. This is called Transfer learning.

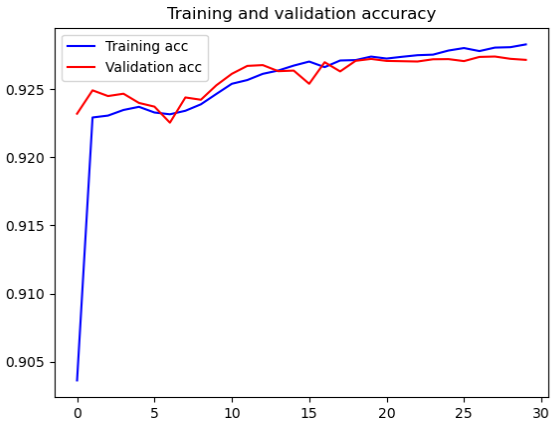

Figure 13: Accuracy for poster training for VGG16 model

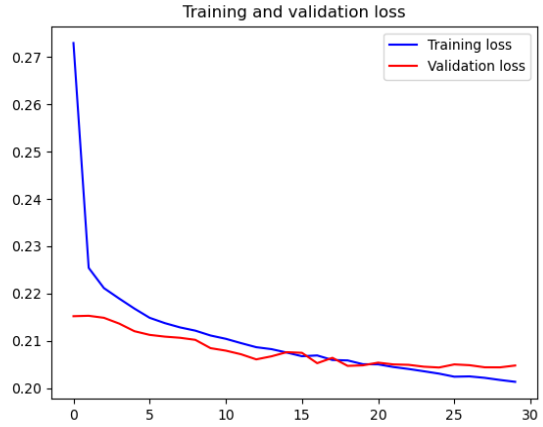

Figure 14: Loss for poster training for VGG model

Architecture	Accuracy (One Class)	Accuracy (Two Classes)	Accuracy (All classes)
Custom Model	84.67%	26.59%	3.6%
VGG16	82.36%	24.28%	2.6%
VGG19	79.77%	22.8%	3%
ResNet50	78.76%	12.65%	0.9%
InceptionV3	78.57%	25.94%	3.3%

Table 2: Comparing CNN architecture for image classification on the test dataset

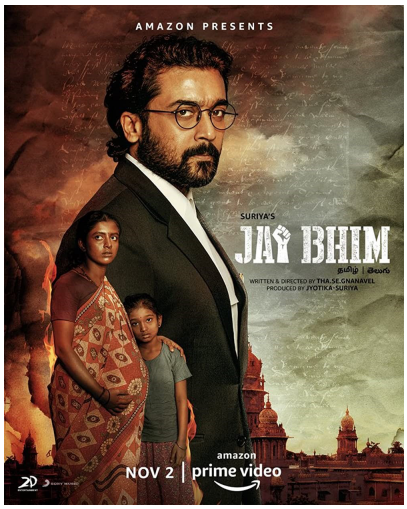

Figure 15: Movie poster with actual classification as Crime and Drama

Model	Prediction 1	Prediction 2	Prediction 3
Custom	Family	War	Music
VGG16	Family	Crime	Documentary
VGG19	Adult	Documentary	War
ResNet50	Family	Drama	Crime
InceptionV3	Family	Crime	Sci-Fi

Table 3: Model Predictions for the movie poster

Experiments with TFIDF

TFIDF not only considers the frequency of each word in the vocabulary but also the weight associated with each word in the vocabulary to further accurately determine how important a word is to a text.

$$tfidf(w_k, d_i) = w_{ki} * log \frac{m}{\sum_{d=1}^{m}\{w_{kd} > 0\}}$$

Here w_{ki} is the frequency of the k^{th} word in the i^{th} movie's description (d_i) and m is the number of training/test sets.

The models being used for classification are LinearSVC, Multinomial Naive Bayes and Logistic Regression. The accuracy and F1 score are calculated using sci-kit learn library. The method used to calculate the score was cross validation.

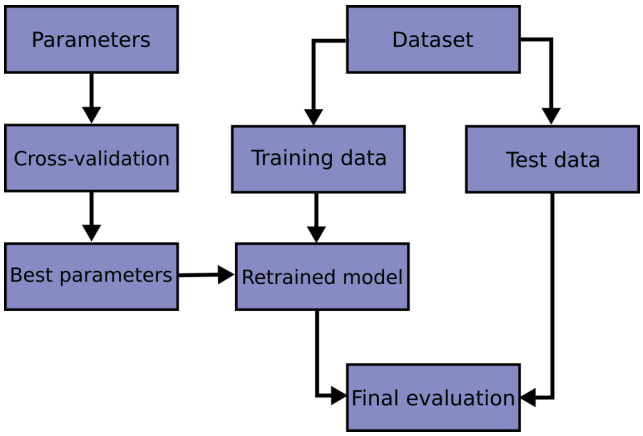

Figure 16: Cross validation architecture

Model	Accuracy	F1 Score
Linear SVC	0.334451 ± 0.014253	0.334430 ± 0.014253
Logistic Regression	0.360900 ± 0.009441	0.360900 ± 0.009441
Multinomial Naïve Bayes	0.344895 ± 0.004571	0.344895 ± 0.004571

Table 4: Average scores obtained after cross validating 5 folds with the standard deviation

Conclusion

We have implemented models for poster genre classification and compared the results. Multilabel classification is obtained by taking the top 3 probabilities of the different classes. As observed from comparing the different models, it can be said that larger models trained using transfer learning performed worse than the simpler CNN model trained from scratch.

Also, it can be observed that correctly predicting all three classes of the movie correctly is a challenging task for all models. It can be argued that predicting all three classes of a movie is impossible even for Humans just by looking at the poster.

We implemented a LSTM architecture for classifying movies based on their plot summaries. As for further work, we can incorporate critic reviews for word based genre classification. We can explore newer architectures such as Bi LSTM for natural language processing and better preprocessing of the data.

Using TFIDF to process data and finding the accuracy and F1 score for LinearSVC, multinomial Naïve Bayes and Logistic Regression to compare the metrics. Logistic regression performed best overall giving a still not very good result of around 36% accuracy – which might be because of the heavily skewed dataset.

We plan on improving the performance by utilizing a larger and a more balanced data set.

References

Saxena, S. (2021, March 18). *Introduction to Long Short Term Memory (LSTM)*. Analytics Vidhya. Retrieved December 15, 2021, from https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/

A. M. Ertugrul and P. Karagoz, "Movie Genre Classification from Plot Summaries Using Bidirectional LSTM," *2018 IEEE 12th International Conference on Semantic Computing (ICSC)*, 2018, pp. 248-251, doi: 10.1109/ICSC.2018.00043.

Hrnjica, B. (2019, August 22). *In depth LSTM implementation using CNTK on .net platform*. CodeProject. Retrieved December 15, 2021, from https://www.codeproject.com/Articles/5165357/In-Depth-LSTM-Implementation-using-CNTK-on-NET-Pla

Kleinbaum, D. G.; Dietz, K.; Gail, M.; Klein, M.; and Klein, M. 2002. *Logistic regression*. Springer.

Barney, G., & Kaya, K. (2019). Predicting Genre from Movie Posters. Stanford CS 229: Machine Learning

Chu, W. T., & Guo, H. J. (2017). Movie genre classification based on poster images with deep neural networks. In Proceedings of the Workshop on Multimodal Understanding of Social, Affective and Subjective Attributes (pp. 39-45).

Kim, SW., Gil, JM. Research paper classification systems based on TF-IDF and LDA schemes. *Hum. Cent. Comput. Inf. Sci.* 9, 30 (2019). https://doi.org/10.1186/s13673-019-0192-7

Allamy, S., & Lameiras Koerich, A. (2021). 1D CNN Architectures for Music Genre Classification. *ArXiv, abs/2105.07302*.

S, Deepak and B. G. Prasad. "Music Classification based on Genre using LSTM." *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)* (2020): 985-991.

Kumar, A., Rajpal, A., & Rathore, D. (2018). Genre Classification using Feature Extraction and Deep Learning Techniques. *2018*

10th International Conference on Knowledge and Systems Engineering (KSE), 175-180.

Rupawala, A., Pujara, D., Shikalgar, M., & Ukey, E. (2020). Movie Genre Prediction from Plot Summaries by Comparing Various Classification Algorithms.

Austin, A., Moore, E., Gupta, U., & Chordia, P. (2010). Characterization of movie genre based on music score. *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 421-424.