Final Project Report
Genre Prediction Based on Movie Plot

Prabhjot Kaur
Master of Applied Computer Science
Dalhousie University
B00843735
Email: pr285985@dal.ca

***Abstract*—In this project, I performed text classification task that is predicting movie genres based on movie plot summaries using two different machine learning methods. Automatic genre prediction systems can be useful for movie recommendation engines to suggest suitable movies to the user where genre is not specified. In this work, after reviewing the available approaches, I compared the two machine learning models for predicting movie genres based on plot summaries, namely, Support Vector Machine and MLP Classifier. My findings suggests that the two models have similar performance with acccuracy of 63% for SVM model and 58% for MLP classifier, but there are some unique characteristics that make one model preferable over the other. At the end of the paper, I made specific recommendations on selecting the appropriate model for anyone interested in utilizing them in practice.**

1. Introduction

Text classification or categorization is an age-old problem in the field of Natural Language Processing(NLP). The main aim of the text classification is to assign labels to the textual components which can be sentences, paragraphs or documents. Since text is an unstructured data, extracting insights from text can be very challenging [11]. Automatic labeling or automatic text classification can be done with the help of machine learning methods. In this project, I used supervised machine learning methods to perform automatic labeling of movie plot summaries and assign suitable genres. Once the models were trained with labeled examples, unseen plot summaries were fed to the models for prediction.

The class distribution is imbalanced in movie dataset and total number of genres in the dataset are 2265 which include multi-labels as well.The plot summaries were converted to word embedding using BERT embeddings [12]. The word embeddinging were used to form clusters using K-means clustering algorithm to reassign the genres to the movies data and choose top eleven most occurring genres. Once the data pre-processing was done, dataset was divided into training data and testing data. The two supervised models, Support Vector Machine and MLP Classifier, were trained using the training set where plot summaries were converted to word embeddings. Model comparison was done using F-1 score derived from five fold cross-validation.

2. Related Work

There are many proposed solutions for predicting movie genres from different features of movies dataset. Naive Bayes approach is used to predict movie genre from user ratings [10]. The idea behind this approach was that most of the users are consistent with their movie genre choice and they rate accordingly. I went with the choice of using plot summaries because my focus was on text classification based task.

Another solution in which logistic regression model was used to classify movie scripts based on NLP-related features such as as the ratio of descriptive words to nominals or the ratio of dialogues frames to non-dialogue frames [6].For each movie script,the probability is estimated that the movie belong to each genre based on extracted features and the k best score is taken to be its predicted genres, where k is a hyper-parameter. A small dataset with only 399 scripts is used for the experiment and the best subset of features achieves an F1 score of 0.56 [9].

Different machine learning methods like One-Vs-All approach with Support Vector Machines (SVM), Multi-label K-nearest neighbor(KNN), Parametric mixture model (PMM) and Neural network are examined to classify movie genres based on synopsis [13]. The features used for this experiment is term frequency–inverse document frequency(tf-idf) of the words of synopsis. The train and test sets contained 16,000 movie titles. This experiment predicts top 10 most famous movie genres. The selection of models for my project was inspired from this experiment.

Gated Recurrent Unit (GRU) which is a type of Reccurent Neural Networks(RNNs) is used to predict movie genre based on plot summaries [9]. GRU combines the forget and input gates into a single update gate and has simpler architecture as compared to LSTM (Long short-term memory).The Gated Recurrent Units (GRU) neural networks employed for the probabilistic classification with learned probability threshold approach achieves the best result in this experiment. The model attains a Jaccard Index of 50.0%, a F-score of 0.56, and a hit rate of 80.5%.

After analysing all the above mentioned solutions, I selected two machine learning methods for comparison Support Vector Machine as it performed best in all the analysed methods in the experiment [13] and neural network based Multilayer Perceptron(MLP) classifier as neural networks have also shown high success rate in the natural language processing tasks.

3. Data

The movie plots dataset used in this project was downloaded from Kaggle open datasets [1]. The movie plot summary data is scrapped from Wikipedia [4]. The dataset contains descriptions of 34,886 movies from around the world with the columns: Release Year - Year in which the movie was released, Title - Movie title, Origin/Ethnicity - Origin of movie (i.e. American, Bollywood, Tamil, etc.), Director, Cast - Main actor and actresses, Genre - Movie Genre(s), Wiki Page - URL of the Wikipedia page from which the plot description was scraped, Plot - Long form description of movie plot.

After importing the dataset, various features were analysed for the data pre-processing task. The dataset contains 6,083 movie plots which do not have assigned genres. Since, I was focusing on supervised learning the data without labels was not very useful, so these movie descriptions were removed from the dataset.The origin of the movies was also reviewed and as can be seen from plot(Figure 1), more than 16,000 movies are of American origin. All the other features were removed from the dataset except: Plot and Genre, as I used only plot summaries to predict the movie genre.

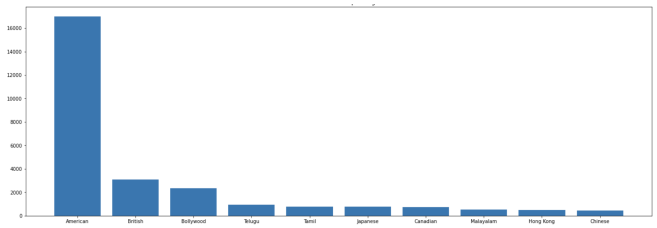

Figure 1. Number of Movies per Origin

In the movie dataset, there are 2,265 unique movie genre sets which also include multi-label genres. In this project, I have focused on predicting single label, so top 11 movie genres were chosen. Clustering algorithm K-means was used to reassign the labels in the dataset.The top 11 occurring genres are:'drama', 'comedy', 'horror','action','thriller', romance','western','crime','adventure','musical','science fiction'. As can be seen from the plot(Figure 2), 'drama' and 'comedy' genre occur for almost 50% of the dataset and hence dominate the genre prediction.

After the basic data analysis and pre-processing, the next task was to make classes uniformly distributed and reassign the movie genres to remove multi-label classes. These tasks were performed with the help of K-means clustering algorithm after converting movie plots to word embeddings.

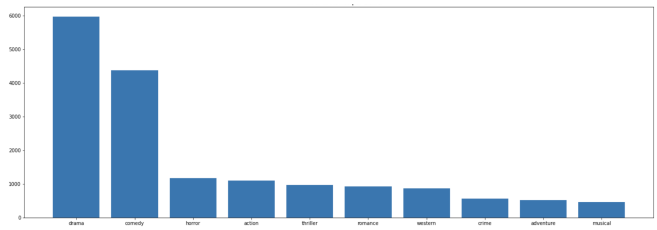

Figure 2. Number of Movies per Genre

4. Overview of the Solution

The major components of the approach used are shown as rectangular blocks in the flow diagram(Figure 3). Role of each component is as following:

Word Embeddings-Once the data pre-processing is done, plot summaries are converted to word embedding using BERT base uncased pre-trained model [8]. These word embedding are the most important step for the solution as both the tasks, clustering and classification, rely on the quality of these word embeddings.

K-means Clustering- As discussed earlier, that dataset contains labels with more that one genres also known as multi-label. Word embedding are used to form ten relatable clusters, based on the most occurring genre in each cluster, multi-label are resolved. This step helps with removing multi-labels from dataset and top 11 genres are assigned to the dataset by end of this step.

Splitting the dataset- This step is used to split the dataset into training and testing data which is stratified by genre. The divided datasets use BERT model to get embedding for corresponding plot summaries. This step is important as train,test datasets are used to train,evaluate and compare both the models.

Train the models- Both the models, Support Vector Machine and MLP classifier are trained with the word embeddings derived from BERT model for train dataset. These models are fed with word embeddings as input and encoded genres as output.

Test the models- Both the models, Support Vector Machine and MLP classifier are tested with the word embeddings derived from BERT model for test dataset. These models are fed with word embeddings as input and predicted genre is compared with true genres. Various performance metrics are derived for each model like precison, recall, f1-score and accuracy.

Cross-validation- Since both the models, Support Vector Machine and MLP classifier, contain some random components like weight initialization in case of MLP, so these could not be simply compared based on accuracy. Five fold cross-validation was used to derive f1-macro score and accuracy for each model. P-value is also derived based on cross-validation scores to establish the statistical significance of an observed effect.

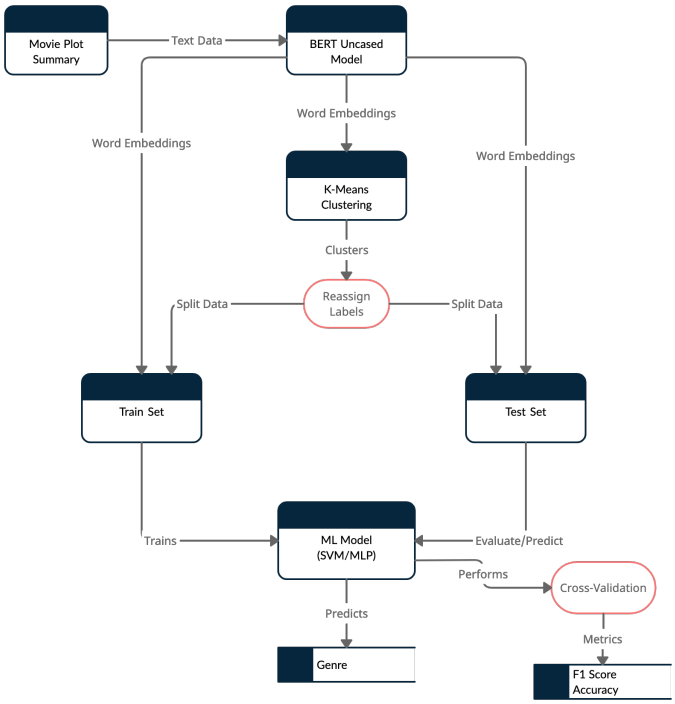

Figure 3. Data Flow Diagram of Solution

5. Word Embeddings

Word embeddings are the vector representations of the word. These word embeddings are used as input for both clustering and classification tasks in my project. Plot summaries were converted to word embedding using pre-trained BERT base uncased model [8]. This is a transformer model that has 12-layer, 768-hidden, 12-heads, 110M parameters and is trained on lower-cased English text. The main reason behind using BERT embeddings is that each word does not have a fixed representation like Word2Vec, rather each word's representation depends on words around it and is based on the context of the text. The pre-trained transformer models in BERT use language comprehension feature extractors to give accurate word representations and give better model performance.

After importing BERT base uncased model, I instantiated new pipeline. BERT provides pipelines which can be defined for various tasks, including Named Entity Recognition, Masked Language Modeling, Sentiment Analysis, Feature Extraction and Question Answering. I went for feature extraction, as it was the requirement for this project.

6. K-means Clustering

The next step was to use clustering to remove multi-labels from the genre field and assign the most influential genre according to the plot summary. I used K-means clustering algorithm for this task. The K-means algorithm clusters data points around K centroids by alternating between two steps: (i) assign each point to its closest centroid, (ii) update the centroids from the newly assigned points [5].I used sklearn.cluster.KMeans [7] for clustering the reduced features data based on similarity. The K-Means algorithm clusters data by trying to separate samples in N groups of equal variance, minimizing a criterion known as the inertia or within-cluster sum-of-squares. It scales well to large number of samples and has been used across a large range of application areas in many different fields.

K-means has three steps. The first step chooses the initial centroids, with the most basic method being to choose samples from the dataset . After initialization, K-means consists of looping between the two other steps. The first step assigns each sample to its nearest centroid. The second step creates new centroids by taking the mean value of all of the samples assigned to each previous centroid. The difference between the old and the new centroids are computed and the algorithm repeats these last two steps until this value is less than a threshold. In other words, it repeats until the centroids do not move significantly.

The number of clusters K needs to be chosen in order to run K-means which was passed as K equals 10 in this case. After, getting the assigned clusters for the entire dataset, each cluster was grouped by genres and if the genre does not belong to top eleven occuring it was reassigned the most occuring genre in that cluster. This decision was based on the assumption that since BERT word embeddings were given as input to the model, all related plot summaries should be in each cluster. After reassigning of the labels is done, genres are encoded using label encoder.Additionaly, embeddings are visualised to see if they form any clusters. Here I used t-SNE to project the plot embedding on to a plane(Figure 4).

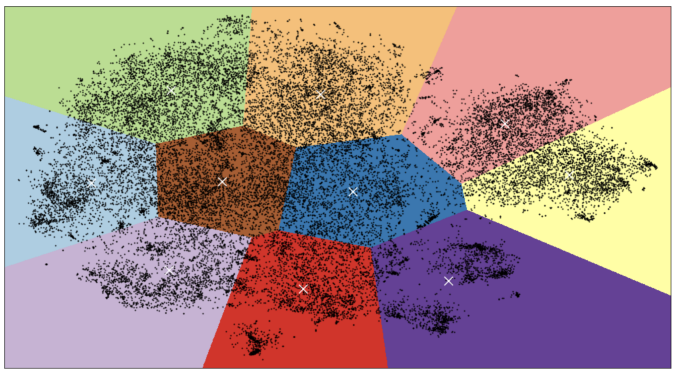

Figure 4. K-means clustering on the plot embeddings (t-SNE-reduced data) Centroids are marked with white cross

7. Classification Models

For the text classification task, two different models were chosen Support vector machine and Multilayer perceptron which is a feed-forward neural network. Both the models are structured differently and have different supervised learning technique. Support vector machines work by identifying the

hyperplane that corresponds to the best possible separations among the closest observations belonging to distinct classes whereas Neural networks follow universal approximation theorem that if a decision boundary of a classification problem can be defined as a continuous function, which is always the case, then it can also be defined as a continuous mapping of the feature space [3]. Both the models were trained and tested on the same dataset and their performance was compared.

7.1. Preparing dataset

The movie dataset was splitted using train_test_split() method [7] which was stratified by genre field to have balance of classes in both test and train data. After splitting the data set, train data had 23,042 samples and test data had 5,761 samples. These sample were converted to word embedding using BERT model. After conversion, train data had shape 23042 X 768 and test data had shape 5761 X 768. Similarly, train labels had shape 23042 X 1 and test data had shape 5761 X 1. These train and test data set were used in both the models, Support vector machine and Multilayer perceptron.

7.2. Support Vector Machine

Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection [2]. SVM is efficient in high dimensional spaces and uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.Support vector machines work by identifying the hyperplane that corresponds to the best possible separations among the closest observations belonging to distinct classes [3]. SVMs aren't restricted to the feature space in which the input is defined. Instead, they can increase the dimensionality of the problem up to a space in which a solution exists.

I used Sklearn's SVC(Support Vector Classification) model and passed the training data to train the model. Once the model was trained, test data was passed to predict the genre values. Later, classification report was generated which gives performance metrics like precision, recall, f1-score for each class shown in Table 1 and also accuracy, f1-macro and f1-weighted of the model shown in Table 2.

7.3. MLP Classifier

Multi-layer Perceptron is a class of feed-forward artificial neural network. Any neural network with input layer, hidden layer(s) and output layer where all neurons of previous layer are connected to next layer is called feed-forward neural network. This model optimizes the log-loss function using LBFGS, learning rate of 1e-5 is used. It has three hidden layers of 20,10,2 neurons and runs for 1500 iterations.

I used Sklearn's MLP Classifier model and passed the training data to train the model. Once the model was trained, test data was passed to predict the genre values. Later, classification report was generated which gives performance metrics like precision, recall, f1-score for each class shown in Table 3 and also accuracy, f1-macro and f1-weighted of the model shown in Table 4.

Class	Precision	Recall	f1-score
action	0.25	0.00	0.01
adventure	0.25	0.02	0.04
comedy	0.67	0.43	0.53
crime	0.00	0.00	0.00
drama	0.62	0.92	0.74
horror	0.59	0.27	0.37
musical	0.00	0.00	0.00
romance	0.00	0.00	0.00
science fiction	0.76	0.79	0.77
thriller	0.00	0.00	0.00
western	0.76	0.73	0.75

TABLE 1. PERFORMANCE METRICS PER CLASS FOR SVM

Metric	Value
Accuracy	63%
f1-macro	29%
f1-weighted	56%

TABLE 2. ACCURACY AND F1-SCORE SVM

Class	Precision	Recall	f1-score
action	0.15	0.05	0.08
adventure	0.00	0.00	0.00
comedy	0.60	0.41	0.48
crime	0.00	0.00	0.00
drama	0.62	0.85	0.72
horror	0.37	0.43	0.40
musical	0.40	0.06	0.11
romance	0.00	0.00	0.00
science fiction	0.62	0.72	0.67
thriller	0.00	0.00	0.00
western	0.60	0.62	0.61

TABLE 3. PERFORMANCE METRICS PER CLASS FOR MLP

Metric	Value
Accuracy	60%
f1-macro	28%
f1-weighted	54%

TABLE 4. ACCURACY AND F1-SCORE MLP

8. Model Comparison

Cross-validation is used to evaluate the performance of both the models. As the MLP classifier contains random component that is weight initialization, so performance cannot be compared based on single session. Thus, cross-validation is used to resolve this issue, test set is held out and five fold cross validation is performed on that which gives the f1-score as well as accuracy for each run. The training set is split into k(equals to 5 in this case) smaller sets. The performance measure reported by k-fold

cross-validation is then the average of the values computed in the loop. The following procedure is followed for each of the k "folds":
1. A model is trained using of the K-1 folds as training data.
2. The resulting model is validated on the remaining part of the data.

Cross-validation F1-scores are plotted for both models in Figure- 5. P-value is also derived based on cross-validation scores to establish the statistical significance of an observed effect. P Value is a probability score that is used in statistical tests. If the p-value is smaller than the threshold, e.g. 1%, 5% or 10%, then we reject the null hypothesis of equal averages In our case, the cross-validation accuracy of SVM model is 63% and and for MLP classifier is 58%, but P-value for their F1-score is 0.014 which is around 1% which means SVM has better statistical performance as compared to MLP.

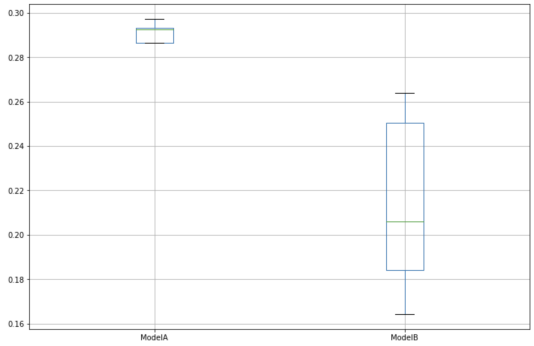

Figure 5. Cross-val F1-score plot for SVM and MLP

9. Conclusion

Based on the observations in this project, both the models have performed almost equally. The five cross validation scores for SVM and MLP are almost same i.e. SVM has accuracy of 63% and MLP has accuracy of 58% which does not show huge difference. But, when statistical test that is p-value test was performed on cross-val f-1 scores of both models it resulted in 0.014 which is around 1%. This result signifies that null hypothesis of averages should be rejected and SVM model has performed better than MLP. It also signifies that SVM is more reliable and its performance would be equal to or more extreme than its observed value. Thus, based on this observation, I recommend the use of SVM for similar text classification problems.

Apart from this, further research could be carried out by using advanced neural networks like LSTM and GRUs. The problem could be further explored by predicting multi-label genres rather than single label. So, there are lot of different directions in which this experiment could be further carried out based on the requirements.

References

[1] Datasets. https://www.kaggle.com/datasets. Accessed: 2020-12-1.

[2] Support vector machines. https://scikit-learn.org/stable/modules/svm.html. Accessed: 2020-12-6.

[3] Svm vs neural network. https://www.baeldung.com/cs/svm-vs-neural-network. Accessed: 2020-12-5.

[4] Wikipedia movie plots. https://www.kaggle.com/jrobischon/wikipedia-movie-plots. Accessed: 2020-12-1.

[5] A. Bietti. Online learning for audio clustering and segmentation. 2014.

[6] A. Blackstock and M. Spitz. Classifying movie scripts by genre with a memm using nlp-based features. 2008.

[7] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[9] Q. Hoang. Predicting movie genres based on plot summaries, 2018.

[10] E. Makita and A. Lenskiy. A multinomial probabilistic model for movie genre predictions, 2016.

[11] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao. Deep learning based text classification: A comprehensive review, 2020.

[12] M. Müller and P. E. Kummervold. Covid-twitter-bert repository. https://github.com/digitalepidemiologylab/covid-twitter-bert. Accessed: 2020-10-14.

[13] K. wing Ho. Movies ' genres classification by synopsis. 2011.