Movie Genre Prediction

Introduction

The analysis is mainly divided into three parts. The first part is data processing, in which the data will be explored, preprocessed and transformed using Tf-Idf Vectorizer. The second part is text classification using two different methods. Finally, a conclusion is drawn with comparison of the classification accuracy of the two methods.

Data Processing

Data Exploration

Before building the model, it is a good practice to have a look at the distribution of the origin data and get some insights. Figure 1 shows the distribution of genres in the training set, where many of the movies belong to the drama or comedy genre.

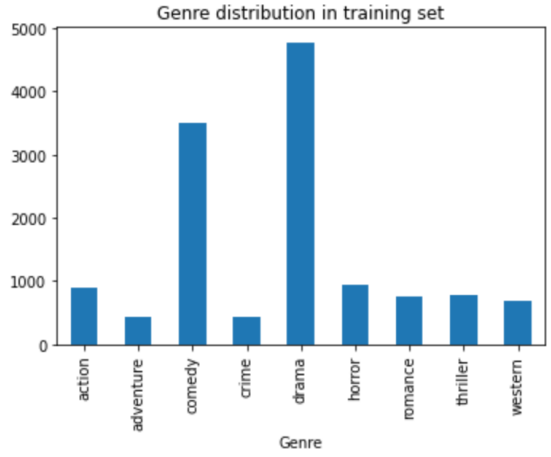

Figure 1. Genre distribution in training set

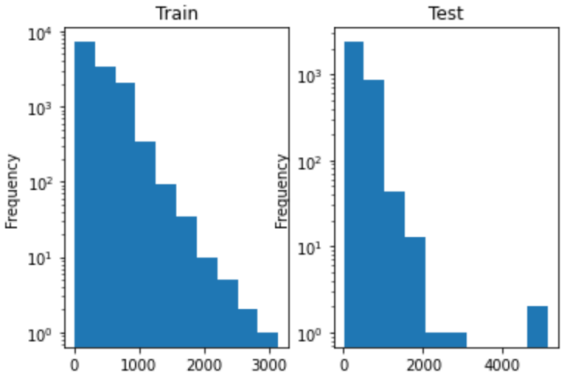

Figure 2. The distribution of sentence lengths in training set and test set

The sentence length of the text variable Plot in the training set and the test set are shown in Figure 2. We can see that the test set has a few texts with sentence length over 4000 words, while the longest sentence in the training set is around 3000 words.

Data Cleaning

To clean the text data, we first removed backslashes, kept only english alphabets, striped extra white-spaces and normalized text. Then we removed the stopwords in English and applied stemming to the text to get the cleaned data.

Tf-Idf Vectorizer

In the feature extracting process, Tf-Idf Vectorizer is used to convert text to a matrix of TF-IDF features. For the parameters in the Tf-Idf Vectorizer, we used english stopwords, set ngram_range to (1,2) and set max features to 150000.

Model Building

After processing the text data, we can feed them into models to see the accuracy and get prediction of genres for the movies in the test set. To find the most appropriate parameter set, the grid search method is used to optimize hyperparameters. Cross validation is also used with f1 score to validate and evaluate model performance.

Method 1: Logistic Regression

In the logistic regression model, different parameter sets (Figure 3) were tried and the grid search result suggested that a logistic model with C=20, solver='liblinear', multi_class='auto' and penalty='l2' produced a best f1 score (0.614).

```
parameters = {'C' : [1,20,50],
              'solver' : ['liblinear', 'newton-cg'],
              'multi_class': ['multinomial','auto'],
              'penalty': ['l1','l2']}
```

Figure 3. Parameter sets for logistic regression

Method 2: Stochastic Gradient Descent Classifier

SGDClassifier is a generalized linear classifier that will use Stochastic Gradient Descent as a solver. The parameters we want to tune are shown in Figure 4. The grid search result suggested that a SGDClassifier with alpha=0.0001, max_iter=100, random_state=42 and early_stopping=False produced a best f1 score (0.616).

```
parameters = {'alpha': [ 0.01, 0.001, 0.0001],
              'max_iter': [100,1000],
              'random_state': ['None',42],
              'early_stopping': [True,False]}
```

Figure 4. Parameter sets for SGDClassifier

Conclusion

By comparing the f1 score of the two models, we can see that the f1 score of the SGDClassifier is slightly higher. To predict the movie genre for the test set, the SGDClassifier is used and resulted in a 0.60206 score on kaggle.