A Text Classification Assignment for Collecting and Analyzing Big Data (B-KUL-S0K17A)  
FAO. Prof. Neal Caren  
  
  
  
Predicting Movie Genre Based on Plot Summary

MSc Statistics and Data Science, Faculty of Science, KU Leuven

Academic Year 2020/2021

Authors

Fazekas, Balazs r0827454

Vandermeersch, Lili r0691855

Predicting Movie Genre Based on Plot Summary

**(Pre)Processing**

We import the training set, the test set and our stopwords list. The stopwords list contains a list of frequent words, for which we have no use. We would rather continue our analysis without taking these words into consideration, so we remove them.

Next, we use the n-grams method to create bigrams and trigrams from the expressions in our dataset. Bigrams are two-word expressions which relate to a single piece of meaning. Example: "fast food", "United States" Trigrams are three-word expressions which relate to a single piece of meaning. Example: "against all odds", "based on results"

By creating n-grams, we can possibly improve our prediction score, as n-grams will try to predict how a certain number of expressions should end. Example: "United States" would have a large probability to end with "of America".

We create a function to preprocess and later process the words in our dataset. We get rid of undesired characters such as:

-roman numerals

-signs such as: ", @ \ ' ? . $ % \_

-single characters

-new line characters

In the processing part, we convert the ‘Plots’ into a list of lowercase tokens, build n-grams and implement lemmatization. With tokenization we create a vocabulary. With lemmatization we turn words with different endings (but still the same stem), into just the stem. Example: "uses, used, user" turned into "use". Now we can start our analysis.

**Methodology**

We create a pipeline where we use CountVectorizer, TfidfTransformer and SGDClassifier. We also run Gridsearch on the training set to autotune our hyperparameters.

CountVectorizer: Creates tokens from the words in our "Plots". We need these tokens for our classification. Tokens are needed to create objects from phrases, words or letters, in a way that our computer understands these tokens. It is also crucial in building a vocabulary, needed for our classification.

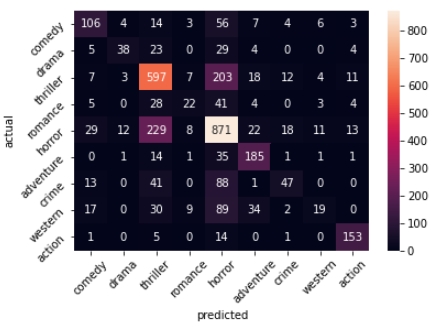
TfidfTransformer:"Tf" is short for term frequency. "idf" is short for inverse document frequency. TfidfTransformer assigns weights to tokens. It assigns more weight to more important tokens. Important tokens are usually those which do not occur frequently. This is because a rarely occurring word in a phrase is likely to have a large influence on the meaning of its phrase. This will possibly greatly increase our prediction scoring.

SGDClassifier: SGD stands for stochastic gradient descent, this is our classification model. It is based on a linear SVM classifier, with SGD training. The gradient loss is updated with each new sample, with a decreasing learning rate for each new sample. A learning rate that decreases over time is optimal, because the heavy weight changes are done at the start, leaving the classifier to only have to deal with smaller changes, fine-tuning, as the classifier progresses into later stages of classification.

GridsearchCV considers all possible combinations of hyperparameters throughout our dataset, and finds the optimal hyperparameters after having considered each combination. We autotune our hyperparameters this way.

**Prediction confusion matrix**

We showcase our prediction accuracy on the training set on a confusion matrix. A confusion matrix shows our predicted genre values, on the x scale, and the actual genre values on the y scale.

On the training set we can observe the correct classifications and the misclassifications relating to a given movie genre. Our accuracy for correctly predicting that a movie belongs to the comedy genre is 57.92%. Our accuracy for drama prediction is 65.51%, for thrillers it is 60.85%, for romance it is 44%, for horror it is 61.07%, for adventure it is 67.27%, for crime it is 55.29%, for westerns it is 42.18% and for action movies the accuracy is 80.95%.

From above we can see that the highest percentage of misclassifications occur for the western genre. Only 19 movies out of 48 western movies are correctly classified, while 11 movies are classified to the horror genre. This might be explained by the generally gory nature of many western movies.

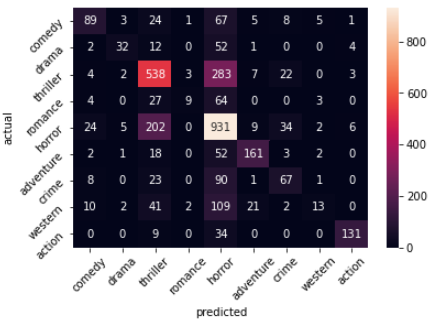
The second weakest prediction occurs for romance movies, where 22 movies out of 50 are correctly predicted. We would have expected that most misclassifications would belong to the comedy genre, but they in fact are westerns. This is not terribly surprising, as many western storylines contain extensive romantic narratives.

Next up is crime, where 47 out of 87 movies are correctly classified. We expected that most misclassifications would belong to the horror genre, due to the similar nature of the movies, and they do, at a number of 18. Then, comedy was classified correctly 106 times out of 183, most of the misclassifications belong to horror, which is surprising.

Next we have thrillers, where 597 out of 981 were correctly classified, and most misclassifications belong to horror, which is as expected. Then, we have horror, with 871 correct classifications out of 1426, where most misclassifications belong to thrillers. Then, we have drama with 38 out of 58 correct classifications, where most misclassifications belong to horror, which is surprising. Then, there is adventure, with 185 out of 275 correct predictions, most misclassifications belong to westerns, which makes sense. Finally, our model performed best on the action genre with 153 out of 189 correct classifications, where most misclassifications belong to horror, which might be due to the gory and or action packed sequences in both genres.

**Alternative approach**

For our alternative approach, we use the Naive Bayes classifier for Multinomial models. This model is supposedly ideal for text classification.

**Conclusion**

By comparing the two confusion matrices, we can see that the predictions were superior for our main model, the SGD classifier for the genres comedy, drama, thriller, adventure, western and action, and our alternative approach using the Naive Bayes classifier for multinomial models was superior for romance, horror and crime.

**References:** <https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html>

<https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html>

<https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html>

<https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html>

https://scikit-learn.org/stable/modules/generated/sklearn.model\_selection.GridSearchCV.html