Name: Morgan Gere

Course: IST 736- Text Mining

Date: 5/27/2022

**Compare MNB and SVMs for Kaggle Sentiment Classification**

**Introduction**

To compare Multinomial Naïve Bayes (MNB) and Linear Support Vector Machines (SVM) the Kaggle move review data set was obtained (<https://www.kaggle.com/c/sentiment-analysis-on-movie-reviews>). The data provided is a training set of data consisting of 156,060 examples of movie reviews and a sentiment label. The labels are 0-4. 0 is “very negative”, 1 is “somewhat negative”, 2 is “neutral”, 3 is “somewhat positive”, and 4 is “very positive”. The provided data also has a test set which only consists of 66,292 movie reviews without labels. The goal is to vectorize that data and create models from the training set choosing the best model and predict the testing set and submit the best model to Kaggle and receive a score. The models will be evaluated based on their accuracy, precision, recall, and a look at the f1-score.

**Method**

Input and Preprocessing

Both the training data and test data was imported into separate pandas data frame and manually inspected for required preprocessing. It was determined that none was needed. The value counts of the labels were also generated to see a baseline for the final model. The baseline was determined to be 50.99%. This was found to compare the final model before Kaggle submission with its cross validation generated accuracy.



Graphical user interface, text, application

Description automatically generated Text, letter

Description automatically generated\

For the traing data the text containing the actual reviews were placed into a list and the sentiment labels were placed into another separate list. For the testing data the text containing the reviews were placed into a list and the id’s were obtained for kaggle submission.

Text, letter

Description automatically generated

Data splitting

For the purposes of comparison of the different models, the hold out method was implemented. The movie reviews and their labels were split into 60% (training data) to train a model and 40% (validation data) for evaluation of its predictions.

Text

Description automatically generated

To determine the baseline for the hold out method and to see if the potion of the data obtained for training had similar distribution of each sentiment category the number of each label was generated. This number shows the distribution compared to the full data. The largest portion was then divided by the sum of the split portion to find the baseline percent. This was found to be 50.96% which is quite like the baseline of the whole data together.

Graphical user interface, text, application, email

Description automatically generated

Unigram SVM and MNB

The training data was vectorized using term frequency inverse document frequency (tfidf) vectorizer from sklearn selecting to remove stop words and set the minimum document frequency to 5. The stop words were removed because they are ubiquitous in the text and would have only created issues with overfitting the training data. The vectorizer was used on the validation data to transform it for prediction.

Text

Description automatically generated

Text

Description automatically generated with medium confidence

Using the vectorized training data and their corresponding labels models were created using LinearSCV from sklearn and MultinomialNB from sklearn.

Text

Description automatically generated Text

Description automatically generated

To view what tokens had been selected at the extreme values the 10 best indicators of very negative sentiment and very positive sentiment were obtained from each model.

Graphical user interface, text, application, email

Description automatically generatedText

Description automatically generated

Graphical user interface, text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

The model was used to predict the sentiment of the validation data and a confusion matrix was created. The scores obtained from the confusion matrices were placed into a dictionary and then into a bar graph for easy visual comparison later on.

A picture containing text

Description automatically generatedA picture containing text

Description automatically generated

Text, letter

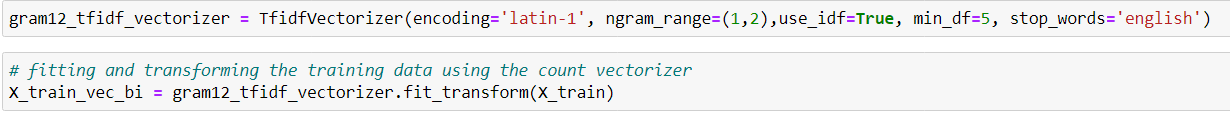
Description automatically generated with medium confidence

Text

Description automatically generated

Bigram and Unigram SVM and MNB

The tfidf vectorizer was used again the same parameters were kept, except that n-gram range was selected as 1,2. This allowed the model to use bigrams in addition to unigrams during vectorization. Again, the validation data was transformed for prediction.



A picture containing logo

Description automatically generated

The new vectors that contained bigrams and unigrams were used to create a SVM model and MNB model.

Text

Description automatically generated Text

Description automatically generated

The best indicators for very negative sentiment were obtained to check for the bigram incorporation into the models. Note that for MNB model the number was increased from 10 to 165 in order to find a bigram being used.

Text

Description automatically generated

Text

Description automatically generated

The models were used to predict the validation data and confusion matrices were produced. Again, this data was placed into a dictionary and graphed for visual comparison.

A picture containing text

Description automatically generated A picture containing table

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Cross Validation SVM

The two SVM model (unigram vs bigram and unigram) confusion matrices results were placed into a dictionary together and graphed. This was done to select the better version for the cross validation and Kaggle submission.

Using the tfidf vectorizer with the same selected parameters (min. document frequency set to 5, and stop words removed) the entire original training data was vectorized to have a vector of unigrams and bigrams. A 10-fold cross validation was used to obtain accuracy scores. These scores were used to generate an average accuracy for the final model.

Graphical user interface, text, application, email

Description automatically generated

Kaggle Submission

The testing data with no labels was transformed using the vectorizer created in the cross-validation model and the bigram and unigram model was used to predict the sentiment of each movie review. This was exported to a csv file along with the Kaggle ID’s and a header of PhraseID and Sentiment. The CSV file was submitted to Kaggle, and a screen shot of the results are in the Kaggle Submission section.

**Graphical user interface, text, application, email

Description automatically generated**

**Kaggle Submission**

Graphical user interface, text, email

Description automatically generated

**Results**

Unigram SVM vs MNB

*Chart, histogram

Description automatically generated*

Unigram and Bigram SVM vs MNB

Chart, histogram

Description automatically generated

Unigram vs Bigram and Unigram SVM

Chart, bar chart, histogram

Description automatically generated

Cross Validation SVM

The Average cross validation score when the full data was run was found to be ~51.26%

Kaggle Submission Results

The obtained score from Kaggle was 0.61090.

**Conclusion**

Unigram SVM vs MNB

The SVM does a better job of classifying the extremes. To illustrate when looking at “Very Negative” the MNB has a very low recall and f1-score. This shows it is not correctly predicting many of these only getting 4% of the possible prediction. The Precision is comparable between the models for “Very Negative” showing that if either model does predict a score of “Very Negative” about half of the predictions are correct. The MNB is having a very high score in the “Neutral” recall but the precision of “neutral” is low and this shows that the MNB is placing a lot of prediction into “Neutral” even when the actual sentiment is something else. It captures a high accuracy because this data is skewed towards neutral sentiment. Overall, the SVM is doing a better job.

Unigram and Bigram SVM vs MNB

These models when compared to one another are having the same results as the unigram model. Since the indicative words needed to be expanded for MNB to 165 the bigrams are not changing this model in meaningful ways. With the SVM the bigrams appear in the top 10 indicative words showing this model is using them. A direct comparison of the SVM unigram vs SVM bigram and unigram will reveal more information.

Unigram vs Bigram and Unigram SVM

When looking at the two SVM models next to one another for each level of sentiment towards the middle (neutral) the rates increase. For the bigram model it consistently is having a equal or better recall score. This shows its predicting more of each category correctly. On the extreme ends (very negative and very positive) the precision score is slightly lower stating it has a harder time with these and more of its predictions are wrong. Yet it does a better job with this in the other 3 categories.

If the goal is to get a better model overall the bigram and unigram SVM wins out.

Cross Validation SVM

The overall cross validation accuracy is found to be lower than the hold out. This isn’t necessarily a bad thing as it could be showing that the model is not overfitting the data.

Kaggle Submission

The score is good for a first attempt. Inspecting that indicating words and removing more stop words could greatly increase the score. Another possibility would be to and more layers of n-grams or increase the preprocessing steps such as removing punctuation or lemmatization.