Selina Wong Jinen A0142657B

BT2101 ASSIGNMENT 2 TEXT CLASSIFICATION

Task

Exploring the viability of using text classification techniques to classify a review into its App category.

Datasets
R Script: classification_training.R

All columns except review: X, author, date, rating and title columns are removed as these columns will not be useful in adding any information to the classification.

Education (first 10,000)
Finance (first 10,000)
Game (first 10,000)
Social (first 10,000)
Weather (first 10,000)
Education (next 2000)
Finance (next 2000)
Game (next 2000)
Social (next 2000)
Weather (next 2000)

As the entire dataset is very large, a smaller set of training and testing is obtained from the dataset given.

For the purpose of this experiment, training data will be sampled using the first 50,000 rows of the entire data. The **training data (blue)** is formed using the first 10,000 rows of each of the data frames namely education, finance, game, social and weather to ensure consistency and balance of different categories across the entire training data. The **testing data (white)** will be sampled using the next 10,000 rows of data. The testing data is formed using the next 2000 rows of each of the data frames from education, finance, game, social and weather. Both training data and testing data would be combined into a single data set of 60,000 rows. Reviews which are empty or contain "NA" values are removed from the dataset as they will not be useful in adding any information to the classification.

Classifier

Naive Bayes learning algorithm will be used as the main classifier as it is generally faster to run Naïve Bayes on the same set of data compared to Support Vector Machine.

Approach

<u>Individual Text Processing</u>
R Script: text_process.R

Using the set of training and testing data, individual text processing will be carried out in a series of steps: Firstly, a duplicate of the dataset is obtained. From this set of data, non-words will be removed, followed by normalizing text to lower case. After which, stop words provided by the text mining package will be removed. Punctuations from the dataset is then removed, followed by numbers before carrying out porter stemming. A term document matrix is then obtained in order to further remove sparse terms and weight by term frequency. These set of training data is used to train the Naïve Bayes classifier before producing predictions, which is used to generate the following confusion matrix.

Predicted					
	Education	**Finance**	**Game**	**Social**	**Weather**
Education	116	20	1579	23	262
Finance	103	137	1224	68	468
Game	27	8	1812	13	140
Social	54	26	1555	89	276
Weather	48	20	1341	40	551

Weighted Precision = (116/348 + 137/211 + 1812/7511 + 89/233 + 551/1697)/5 = 0.3861
Weighted Recall = (116/2000 + 137/2000 + 1812/2000 + 89/2000 + 551/2000)/5 = 0.2705

Evaluation
The weighted precision and recall obtained are not particularly high. In an attempt to improve weighted precision and recall, a new list of stop words will be introduced in the next approach to investigate if the text classification performance would be better.

Individual Text Processing with Additional Stop Words
R Script: text_process_stopwords.R

Similar to the above approach, text processing is carried out first by removing non-words and converting to lower case. With a new list of stop words[1] in addition to the stop words provided by the text mining package, stop words contained in this vector are removed from the dataset. The text is further processed by similar to the above approach. The Naïve Bayes classifier is trained using the model to produce predictions which generates the following confusion matrix.

Predicted					
	Education	**Finance**	**Game**	**Social**	**Weather**
Education	86	14	1624	14	262
Finance	86	86	1242	67	519
Game	24	8	1859	11	98
Social	28	22	1631	72	247
Weather	47	5	1450	33	465

Weighted Precision = (86/271 + 86/135 + 1859/7806 + 72/197 + 465/1591)/5 = 0.3700
Weighted Recall = (86/2000 + 86/2000 + 1859/2000 + 72/2000 + 465/2000)/5 = 0.2568

Evaluation
Surprisingly, both weighted precision and recall have decreased. Precision reduced by 4% and recall reduced by 5.06%. It is expected that common words usually make the feature vector very large and adds “noise” to the training data, which will penalize accuracy. Hence, removing these common words might improve accuracy. However, this indicates that the increase in the amount of stop words used did not necessarily improve the performance of the Naïve Bayes classifier. In fact, the performance of the classifier became poorer as the text classifier is unable to classify individual reviews accurately into its corresponding categories. This could be because certain stop words could add information to text classification. While removing these stop words may sometimes mean increased accuracy and classification time, it is also essential to select stop words that should be removed from the dataset wisely as they include the most informative features in running a text classification. One way is to grow the stop words list recursively in order to match the right stop words to the classification by looking at the existing stop words within the feature vector columns.

Length of Reviews
R Script: length_reviews.R

In this approach, reviews with a character length of less than 10 are removed by first converting to NA before omitting these entries. After which, equal number of rows of reviews from each category (10,000 each for training and 2000 each for testing) are combined to form the dataset to ensure balance and representative of data from each category. Similar to the first approach, further text processing is carried out. The Naïve Bayes classifier is trained using the model to produce predictions which generates the following confusion matrix.

[1] http://jmlr.csail.mit.edu/papers/volume5/lewis04a/a11-smart-stop-list/english.stop

	Predicted				
	Education	**Finance**	**Game**	**Social**	**Weather**
Education	308	54	1155	19	464
Finance	144	248	1015	50	543
Game	105	42	1536	24	293
Social	102	95	1349	118	336
Weather	72	33	1275	19	601

Weighted Precision = (308/731 + 248/472 + 1536/6330 + 118/230 + 601/2237)/5 = 0.3942
Weighted Recall = (308/2000 + 248/2000 + 1536/2000 + 118/2000 + 601/2000)/5 = 0.2811

Evaluation
Both weighted precision and recall have increased. It appears that the length of reviews contributes as a factor in the performance of the text classifier as it has visibly improved the accuracy of classification. This could be due to longer reviews containing more information about the app category, whereas shorter reviews could contain descriptive words such as “awesome”, which do not add any information to the categories. Eliminating these low quality features do lead to an increase in overall accuracy.

<u>Using Smaller Subset of Training and Testing Data</u>
R Script: training_testing_subset.R

Education (first 1000)
Finance (first 1000)
Game (first 1000)
Social (first 1000)
Weather (first 1000)
Education (next 200)
Finance (next 200)
Game (next 200)
Social (next 200)
Weather (next 200)

In this approach, the amount of training(blue) and testing(white) data are reduced to 5000 and 1000 respectively.

Text Processing is carried out according to the approach above using the new vector of stop words to remove stop words from the data.

The Naïve Bayes classifier is then used to train the model, from which predictions are used to generate the confusion matrix.

	Predicted				
	Education	**Finance**	**Game**	**Social**	**Weather**
Education	4	0	177	13	6
Finance	20	3	151	14	12
Game	8	0	176	7	9
Social	16	0	155	17	12
Weather	3	0	173	14	10

Weighted Precision = (4/51 + 3/3 + 176/832 + 17/65 + 10/49)/5 = 0.3511
Weighted Recall = (4/200 + 3/200 + 176/200 + 17/200 + 10/200)/5 = 0.21

Evaluation

Using a smaller subset of training and testing data has led to a much lower weighted precision and recall compared to using larger training and testing data. This could possibly be due to the absence of important information in smaller subset of training data required in order to classify the reviews into their subsequent categories accurately. Texts in the testing data may not contain features that the classifier has been trained on using the training data. Such words do not exist enough in the training data set, resulting in reviews being classified inaccurately into categories. The classifier may not know which category to classify these reviews from the testing data set.

Using Support Vector Machine (SVM) Classifier

R Script: training_testing_svm.R

In this approach, the SVM classifier is used to train the model with smaller subset of training and testing data (used in the above approach) as SVM generally takes longer to train data. With a smaller dataset, it is possible to reduce excess time required to wait for SVM classifier to train the data. Text Processing is carried out similarly as above before training the model with SVM classifier. Similarly, a confusion matrix is generated after carrying out predictions.

	Predicted				
	Education	**Finance**	**Game**	**Social**	**Weather**
Education	80	18	41	44	17
Finance	7	163	7	6	17
Game	16	18	96	57	13
Social	23	19	31	113	14
Weather	29	12	5	60	94

Weighted Precision = (80/155 + 163/230 + 96/180 + 113/280 + 94/155)/5 = 0.5536
Weighted Recall = (80/200 + 163/200 + 96/200 + 113/200 + 94/200)/5 = 0.546

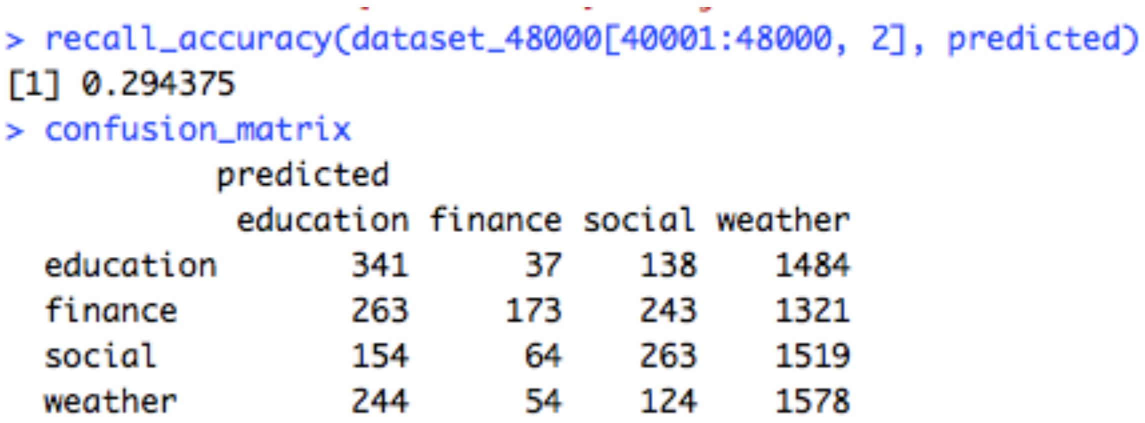
Evaluation

Using the SVM classifier on this dataset has resulted in much higher weighted precision and recall with the Naïve Bayes classifier. The SVM algorithm appears to work better and have higher overall accuracy in text classification on this set of data.

Excluding Game & Weather Categories

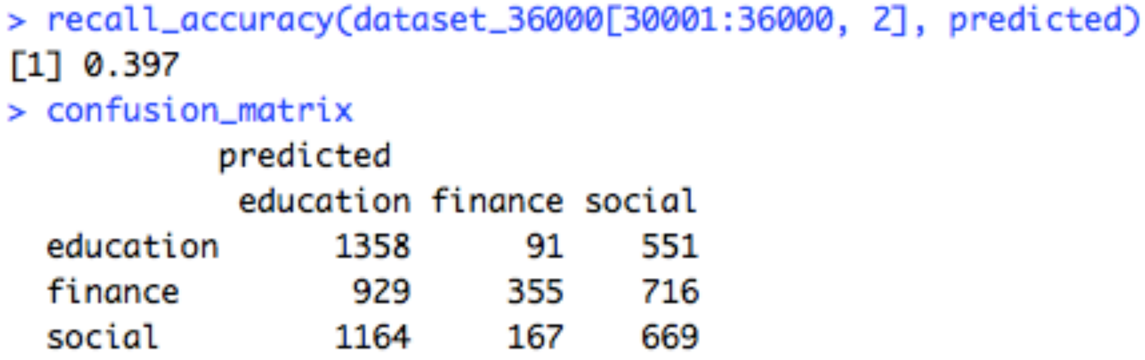
R Script: exclude_game_weather.R

The game category is excluded from this text classification to determine if the text classification results would be more accurate since most of the reviews have been incorrectly classified into the game category. As before, text processing is carried out before producing the confusion matrix.



```
> recall_accuracy(dataset_48000[40001:48000, 2], predicted)
[1] 0.294375
> confusion_matrix
           predicted
            education finance social weather
  education       341      37    138    1484
  finance         263     173    243    1321
  social          154      64    263    1519
  weather         244      54    124    1578
```

It appears that most of the reviews are now inaccurately classified into the weather category. The weather category is further excluded from the training and testing data to determine if the classifier's performance would improve. Notice that the weighted recall has increased from 0.2705 in the first approach to 0.294375.



```
> recall_accuracy(dataset_36000[30001:36000, 2], predicted)
[1] 0.397
> confusion_matrix
           predicted
            education finance social
  education      1358      91    551
  finance         929     355    716
  social         1164     167    669
```

The performance of the classifier on the three categories: education, finance and social appears to be more accurate here, with an improvement of the overall weighted recall to 0.397 from 0.294375.

Evaluation

After narrowing down the categories to education, finance and social, the text classification results are more accurate. It might be due to presence of texts in the feature vector which could distinguish between the distinct categories of education, finance and social.

Conclusion

Text Classification on this set of data revealed that the SVM algorithm is able to classify the reviews more accurately than Naïve Bayes. Data pre-processing is crucial: not only in the sequence in pre-processing, but also in the selection of stop words to be removed from the dataset. As stop words are generally presented in lower case, data pre-processing in this task is first carried out by removing non-words and normalizing data to lower case before removing stop words to ensure that the stop words in the dataset matches the stop words provided in the tm package in R.

The length of reviews also plays a role in determining the accuracy of the classifier. Generally, longer reviews contain more information about the app category, hence results in higher accuracy. Although using a smaller subset of data improves the text classification run time, there is a trade off between run time and accuracy of the text classifier. It makes sense that using a greater amount of dataset is better as there are more information available that can be used to identify the app review category.

It is also observed that there is a large number of app reviews that are are incorrectly classified into the game category. Upon examining the dataset given, this could be due to the fact that reviews from other categories apart from game do not contain distinct and useful words to identify them accurately into their respective categories. Instead, these reviews contain an overlapping bag of general words such as "cool", which are also present in the game app reviews. A smaller subset of categories which may work better would be to exclude the game and weather category from the text classification. As indicated by the text classification results in the last approach, the text classifier is able to classify the reviews into their respective categories more accurately after narrowing down to the three distinct categories: education, finance and social.

To take this text classification further, it is possible to add in bigram features and explore the effect of having bigram features on the accuracy of the text classification.