Text Classification

Hossein Zinaghaji
300098309

Soroush Salehi
300140057

Sahand Malek
30031

Contents

1 Introduction:

In this project, the goal is to identify the author of several books by analyzing some words from the original books. Three different Feature selection methods for texts and five different classifying methods are being used.

The ultimate target is to classify, predict, and compare the methods.

2 Data Explanation and Preparation

Seven different books are selected from Gutenberg Digital Books Library. The books are listed below:

- Three Musketeers by Alexandre Dumas, Pere
- Adventures of Sherlock Holmes, by A. Conan Doyle
- Dorothy and the Wizard in Oz, by L. Frank Baum.
- The Mysterious Island, by Jules Verne
- Mechanical Drawing Self-Taught, by Joshua Rose
- The History and Practice of the Art of Photography, by Henry H. Snelling
- Christmas Carol, by Charles Dicken

Firstly, since each text is downloaded from Project Gutenberg contains a header with the name of the author, the names of people who scanned and corrected it, a license, and so on, we removed the beginning and the end, to be just the content and nothing else.

Then, we tokenized the books to preprocess the texts. We will filter the text by removing punctuations and stopwords and Lemmatization by using the NLTK library.

Then, we made a DataFrame contains 200 records; each includes n words from each book. (We made a loop for n from 10 to 200 with steps of 10).

The word clouds are shown as below:

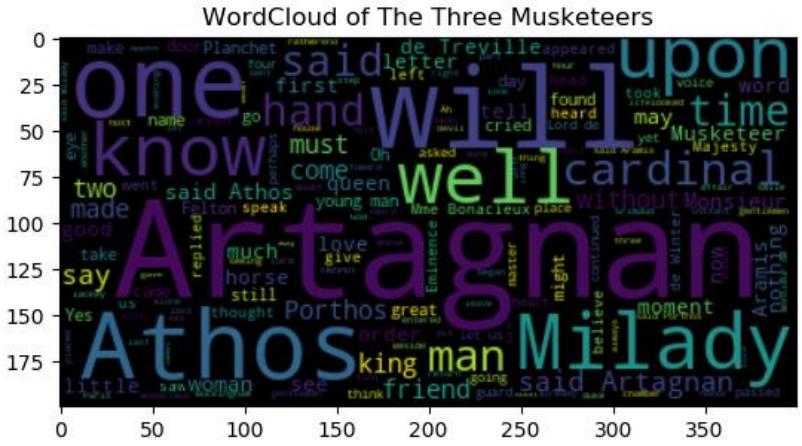


Figure 2-1 WordCloud of The Three Musketeers

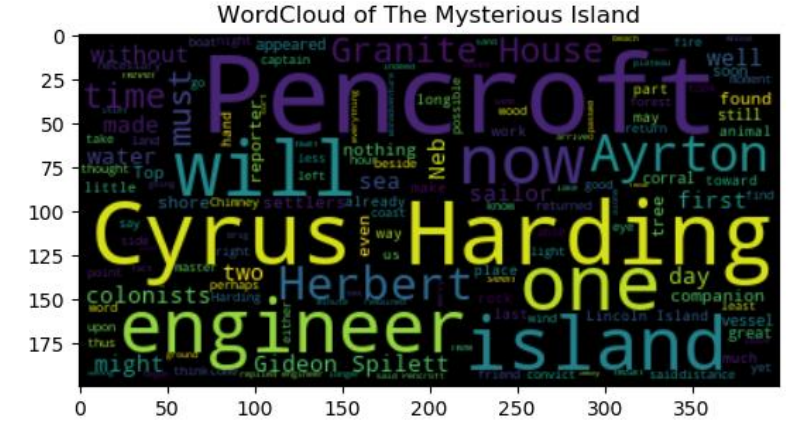


Figure 2-2 WordCloud of The Mysterious Island

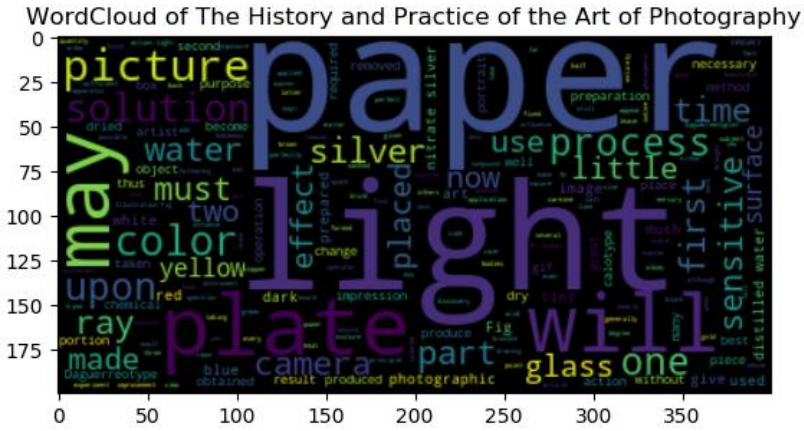


Figure 2-3 WordCloud of The Mysterious Island

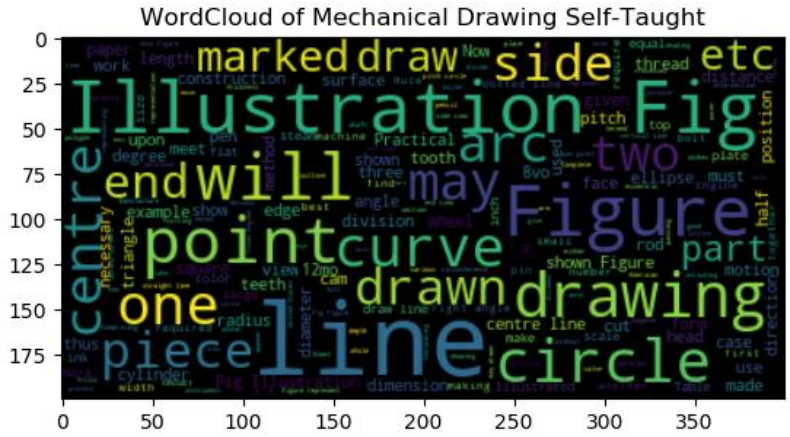


Figure 2-4 WordCloud of Mechanical Drawing Self-Taught

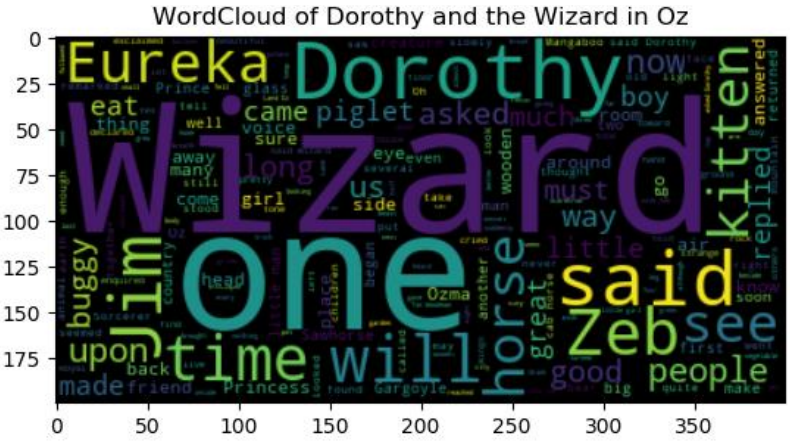


Figure 2-5 WordCloud of Dorothy and the Wizard in Oz

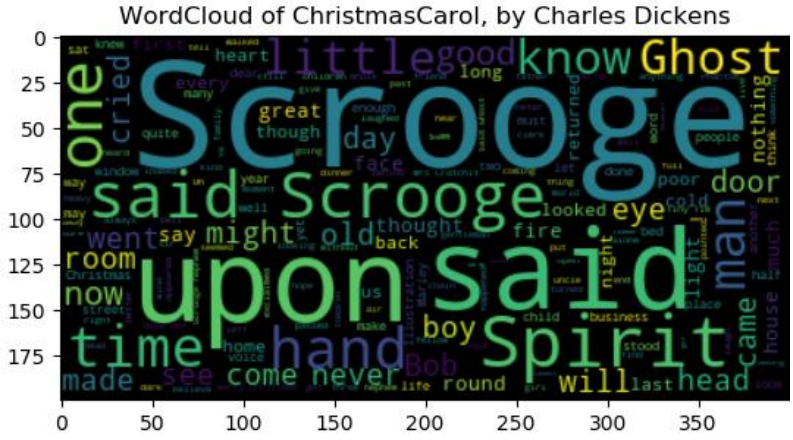


Figure 2-6 WordCloud of Christmas Carol

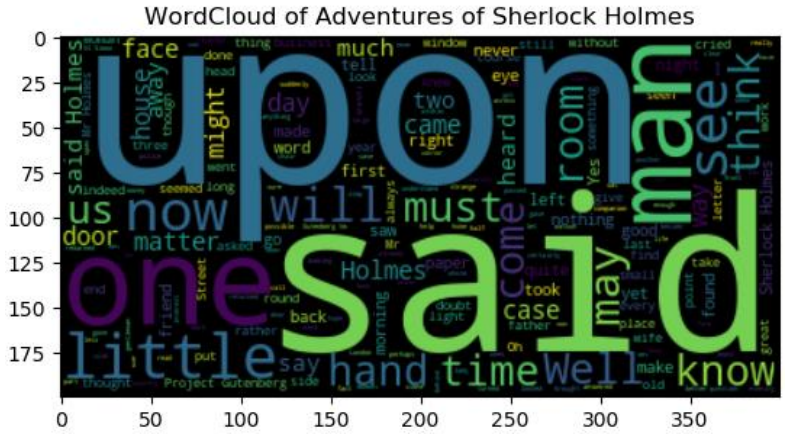


Figure 2-7 WordCloud Adventures of Sherlock Holmes

3 Feature Engineering

Text and text documents are unstructured and therefore need to be transformed into computer-structured and computable data by various operations (categorized as pre-processing). Three different methods of feature engineering of texts are used.

- Bag of Words
- TF-IDF
- N-Gram

3.1 Bag of Words

The Bag-of-words model is a simple representation used in natural language processing and information retrieval (IR). Also known as the vector space model. In this model, a text (such as a sentence or document) is displayed as a package of several sets of words, disregarding of its grammar and even word order. In practice, this model is mainly used as a tool for generating features. After converting the text into a bag of words, we can calculate different sizes to specify the text. The most common type of attribute derived from the model is the repetition of the phrase, that is, the number of times a phrase appears in the text.

3.2 TF-IDF

The value of TF-IDF stands for two words: TF means Term Frequency, which means the number of repetitions of a word in a text, and IDF means Inverse Document Frequency, which can be translated to the number of repetitions in the text. TF-IDF is a way to measure the importance of a word in a document. TF measures that a term is repeated several times in a document and the IDF is used to measure the significance of a term.

3.3 N-Gram

the n-gram is a sequence of n letters in a given sequence of text or words. Depending on their use, items can be phonemes, syllables, letters, words, or pairs. N-grams are usually collected from a textual or oral body.

4 Modeling

Five classification methods have been used in our project.

- Multinomial Naive Bayes
- K-Nearest Neighbors (K-NN)
- Support Vector Machine (SVM)
- Decision Tree
- Random Forest

For comparing the models and choosing the champion model, we used cross-validation scores, and the mean of these ten scores is the final indicator.

Table 4-1 Bag of Words

	Multinomial Naive Bayes	K-Nearest Neighbors (K-NN)	Support Vector Machine (SVM)	Decision Tree	Random Forest
n = 10	50.46%	18.07%	44.82%	38.36%	45.61%
n = 20	73.61%	20.00%	60.75%	50.32%	57.82%
n = 30	84.36%	23.00%	75.71%	58.86%	67.68%
n = 40	88.93%	27.25%	83.18%	68.00%	78.36%
n = 50	93.68%	29.29%	88.39%	69.14%	82.82%
n = 60	97.54%	34.21%	92.54%	80.00%	88.43%
n = 70	98.43%	41.61%	94.04%	78.00%	91.00%
n = 80	99.07%	45.29%	96.54%	84.18%	93.57%
n = 90	99.71%	45.50%	97.64%	82.89%	95.39%
n = 100	99.82%	50.64%	97.71%	83.36%	96.61%
n = 110	99.68%	65.36%	99.14%	88.29%	98.11%
n = 120	99.71%	60.36%	99.11%	86.29%	97.82%
n = 130	100.00%	64.04%	98.82%	88.32%	98.71%
n = 140	99.89%	71.82%	99.64%	86.71%	99.14%
n = 150	100.00%	70.86%	99.61%	90.25%	99.11%
n = 160	100.00%	66.46%	99.36%	90.11%	99.21%
n = 170	100.00%	71.64%	99.64%	92.25%	99.54%
n = 180	100.00%	72.75%	99.68%	90.57%	99.14%
n = 190	100.00%	76.86%	99.54%	91.86%	99.71%
Average	93.94%	50.26%	90.83%	78.83%	88.83%

Table 4-2 TF-IDF

	Multinomial Naive Bayes	K-Nearest Neighbors (K-NN)	Support Vector Machine (SVM)	Decision Tree	Random Forest
n = 10	49.54%	15.93%	48.11%	35.54%	42.93%
n = 20	70.64%	57.36%	66.43%	51.89%	58.61%
n = 30	80.86%	69.11%	78.46%	60.14%	66.86%
n = 40	89.71%	78.68%	86.71%	68.11%	76.36%
n = 50	92.86%	83.54%	92.43%	68.11%	82.86%
n = 60	95.11%	86.82%	94.50%	73.50%	86.11%
n = 70	98.32%	89.39%	97.00%	77.57%	90.64%
n = 80	98.04%	91.82%	98.00%	80.46%	92.96%
n = 90	98.50%	95.07%	98.75%	80.50%	93.43%
n = 100	98.61%	94.89%	99.18%	85.43%	95.93%
n = 110	99.57%	96.00%	99.50%	83.18%	97.14%
n = 120	99.43%	96.96%	99.21%	85.18%	97.82%
n = 130	99.79%	98.43%	99.39%	86.04%	97.82%
n = 140	99.89%	98.18%	99.57%	88.57%	98.39%
n = 150	99.89%	98.61%	99.86%	89.64%	98.93%
n = 160	99.75%	99.00%	99.75%	89.32%	99.14%
n = 170	100.00%	98.75%	100.00%	89.75%	99.57%
n = 180	100.00%	99.57%	99.89%	90.04%	99.46%
n = 190	100.00%	99.75%	99.93%	91.21%	99.50%
Average	93.18%	86.73%	92.46%	77.59%	88.13%

Table 4-3 N-gram

	Multinomial Naive Bayes	K-Nearest Neighbors (K-NN)	Support Vector Machine (SVM)	Decision Tree	Random Forest
n = 10	42.89%	25.25%	35.04%	28.64%	41.14%
n = 20	58.36%	30.21%	49.93%	35.89%	51.93%
n = 30	69.71%	37.50%	58.79%	46.18%	59.32%

n = 40	79.04%	44.36%	69.61%	50.64%	69.29%
n = 50	83.46%	47.71%	73.71%	52.04%	72.79%
n = 60	87.36%	53.11%	78.00%	56.21%	76.89%
n = 70	91.14%	59.46%	83.36%	63.39%	83.04%
n = 80	93.29%	62.18%	84.79%	63.18%	84.46%
n = 90	94.93%	64.32%	86.89%	64.46%	87.39%
n = 100	95.46%	70.07%	89.96%	72.39%	89.21%
n = 110	96.79%	69.32%	91.64%	70.96%	89.50%
n = 120	97.57%	77.64%	92.96%	71.18%	93.00%
n = 130	98.43%	79.21%	93.36%	75.32%	94.43%
n = 140	99.04%	82.36%	93.54%	73.61%	94.21%
n = 150	99.00%	81.71%	95.71%	76.57%	95.71%
n = 160	98.71%	85.04%	96.32%	78.32%	96.00%
n = 170	99.21%	84.50%	95.64%	81.71%	96.36%
n = 180	99.57%	85.79%	96.57%	77.50%	95.25%
n = 190	99.36%	88.75%	98.36%	80.29%	97.61%
Average	88.60%	64.66%	82.33%	64.13%	82.50%

Among used feature engineering methods, BOW with the average score of 80.54% is the champion. Moreover, TF-IDF and N-gram have an average of 87.62% and 76.44%, respectively.

Furthermore, Multinomial Naive Bayes, with the averages of 93.94%, 93.18%, 88.60% (respectively for BOW, TF-IDF, and N-gram), is the champion classifier in our models.

For having a better understanding, We have determined the most informative features. The results are shown below. As it was expected, many of the most informative words for each book are the same as the prominent words in the WordClouds.

```
Alexandre Dumas: know aramis milady porthos cardinal
monsieur de athos say artagnan
Charles Dickens: come old man upon ghost nt christmas
spirit say scrooge
Henry H. Snelling: process water color silver ray solution
picture plate light paper
Joshua Rose: mark show centre circle point fig illustration
figure draw line
Jules Verne: engineer neb would spilett sailor cyrus island
herbert harding pencroft
L. Frank Baum: buggy little people eureka nt say zeb jim
dorothy wizard
```

Figure 4-1 Most Informative Features for BoW - Multinomial Naive Bayes

```
The Author is ['L. Frank Baum']
A. Conan Doyle: re on in n  y  r  t  er  s e
Alexandre Dumas: re er  a r  s  t  n  y   s e
Charles Dickens: n  s  r   c d  er y  t   s e
Henry H. Snelling: ti  c  s y  r  re er t   p e
Joshua Rose: on  p  c en  s s  t  re in e
Jules Verne: n  in  c s  y  re  s er t  e
L. Frank Baum: s   c ar re er d  t   s y  e
```

Figure 4-2 Most Informative Features for N-Gram - Multinomial Naive Bayes

```
Alexandre Dumas: woman know yes madame aramis cardinal
milady athos say artagnan
Charles Dickens: spirit christmas make ghost go look hand
nt say scrooge
Henry H. Snelling: water use solution plate silver color
light ray picture paper
Joshua Rose: view centre show fig circle point illustration
figure draw line
Jules Verne: one would spilett granite neb reply island
herbert harding pencroft
L. Frank Baum: one eat horse little zeb nt say wizard jim
dorothy
```

Figure 4-3 Most Informative Features for TF-IDF - Multinomial Naive Bayes

5 Error Analysis

The performance of a text categorization system is measured by various parameters such as Accuracy, Recall, Precision. Understanding these metrics allows users to understand how well a developed classification model works in analyzing contextual data.

To Evaluate the performance of a textual data classification system, one can use a fixed test dataset (a set of predefined textual data whose class (label) of each sample is specified), or use a method called "cross-validation." Such a process, at the evaluation stage, divides the training data into two subsets; the first subset is used to train machine learning models, and the second subset is used to test system performance.

In this section, various criteria for evaluating the performance of the text categorization model are introduced, and the cross-validation method is also described.

5.1 Accuracy

The Accuracy denotes the number of correct predictions made by the category, divided by the number of total predictions made by the same category. The accuracy of the classification in our model was shown in table 4-1, 4-2, and 4-3. However, this criterion alone is not appropriate for evaluating the performance of a category. When classes in data are imbalanced (that is, the

number of data in a particular label is much higher than in other classes), the system may face a phenomenon called Accuracy Paradox.

As a result of this inconsistency, the classification model is likely to perform very well in predicting the label because most of the data belong to only one class. If such a phenomenon occurs, it is advisable to consider other criteria such as Recall and Precision, to evaluate the performance of the system.

5.2 Confusion Matrix

A confusion matrix is a matrix in which the performance of the relevant algorithms is shown. A confusion matrix summarizes prediction results on a classification problem.

The summary shows the number of correct and incorrect predictions by counting values and breaking down by each class. The confusion matrix indicates how your model is confused when it makes predictions.

5.3 Recall

The Recall represents the ratio of "the number of correctly categorized textual data" in a particular class to the total number of data that must be categorized in the same class. The high value for the recall criterion indicates the low number of data that was not mistakenly categorized in that particular class. Using this criterion alone is not correct for evaluating system performance and should be used alongside the Precision criterion. Because it is easy to design textual classification models that have a high recall, and this does not necessarily mean high precision.

5.4 Precision

The Precision criterion evaluates the ratio of the number of correct predictions made for samples of a particular class to the number of total predictions for samples of the same class (this includes the sum of all the accurate predictions and false predictions).

The high value for the accuracy criterion indicates the low number of data that is mistakenly categorized in a particular class. It is noteworthy that the accuracy criterion is evaluated only for those cases in which the classification model predicts that a sample belongs to a particular class.

5.5 F1-Score

The F1-measure is a type of mean between the parameter P (Precision) and the Parameter R (Recall).

$$F_1 = 2.\frac{precision.recall}{precision + recall}$$

By analyzing the accuracy that it was shown in chapter 4, We decided to use n = 50 to perform our analysis.

Outcomes of the confusion matrix, recall, precision are shown below:

$$\begin{pmatrix} 33 & 5 & 7 & 0 & 0 & 4 & 1 \\ 2 & 37 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 31 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 39 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 47 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 32 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 37 \end{pmatrix}$$

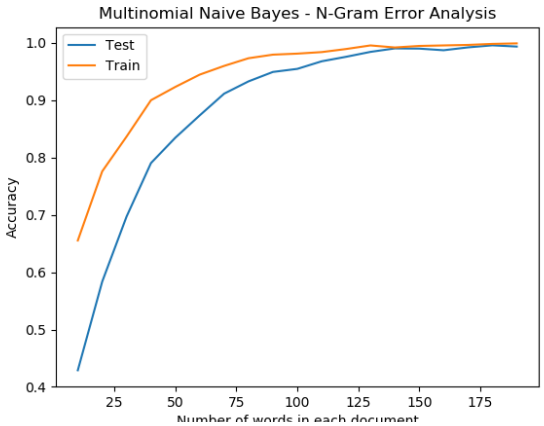
The table shown below is for the BoW Random Forest classifier. We also have analyzed classification report for all other classifiers.

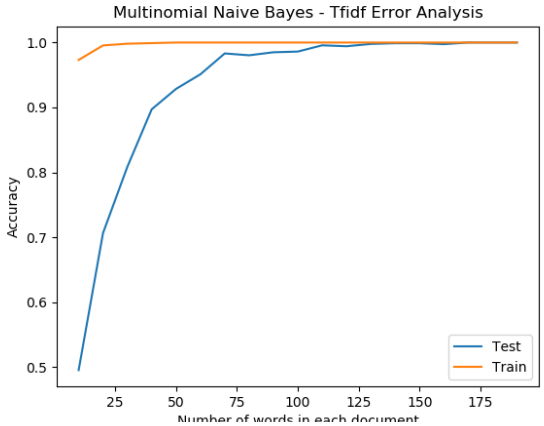
Table 5-1 An example of Classification Report (BoW Random Forest)

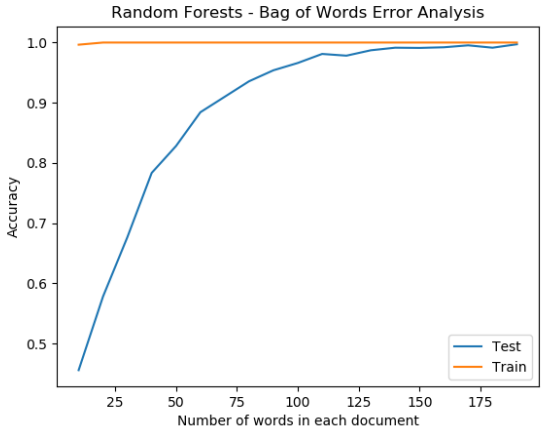
	Precision	Recall	F1-Score	Support
0	0.66	0.74	0.7	50
1	0.84	0.95	0.89	39
2	0.82	0.85	0.84	33
3	0.83	0.97	0.89	39
4	0.95	0.89	0.92	47
5	1	0.74	0.85	34
6	1	0.82	0.9	38
micro avg	0.85	0.85	0.85	280
macro avg	0.87	0.85	0.86	280
weighted avg	0.86	0.85	0.85	280

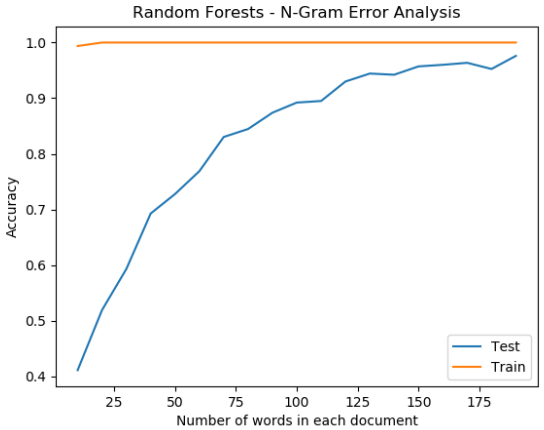
To conclude, we have reached to this point that the Confusion Matrix is accurate enough, and Precision value and Recall value are reasonably high.

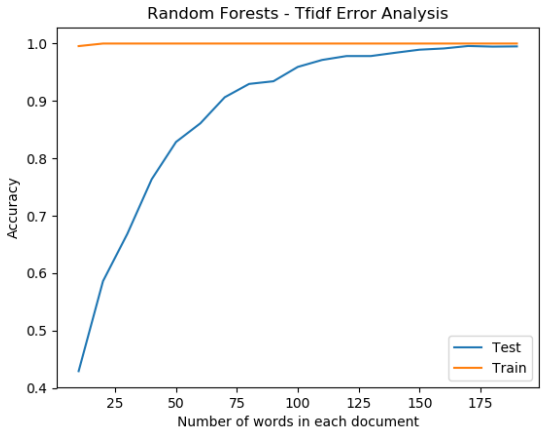
By another analysis that we have done so far, we decided to evaluate the impact of changing n (number of words for each record) on our accuracy. In this regard, we calculate the accuracy (the average of ten-fold cross-validation scores), then we plotted the mentioned score versus score for the training data set.

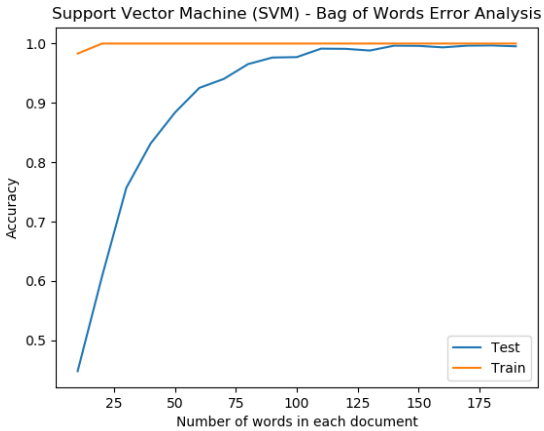

Multinomial Naive Bayes - N-Gram Error Analysis
Test
Train
Accuracy
Number of words in each document

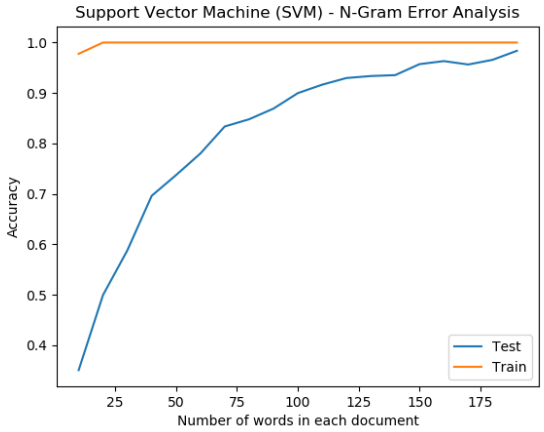

Multinomial Naive Bayes - Tfidf Error Analysis
Test
Train
Accuracy
Number of words in each document

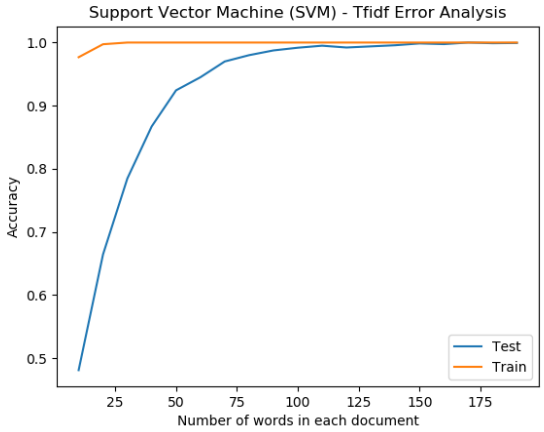

Random Forests - Bag of Words Error Analysis
Test
Train
Accuracy
Number of words in each document

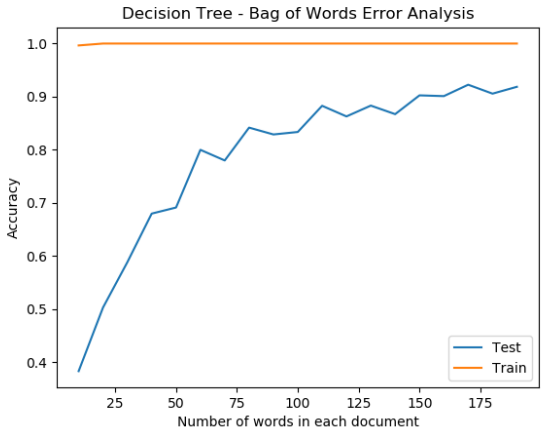

Random Forests - N-Gram Error Analysis
Test
Train
Accuracy
Number of words in each document

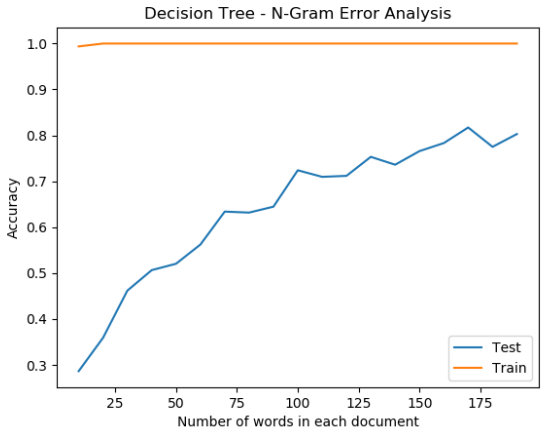

Random Forests - Tfidf Error Analysis
Test
Train
Accuracy
Number of words in each document


Support Vector Machine (SVM) - Bag of Words Error Analysis
Test
Train
Accuracy
Number of words in each document

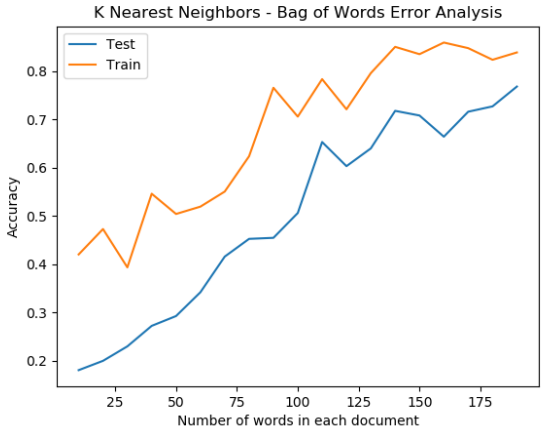

Support Vector Machine (SVM) - N-Gram Error Analysis
Accuracy
Number of words in each document
Test
Train


Support Vector Machine (SVM) - Tfidf Error Analysis
Accuracy
Number of words in each document
Test
Train


Decision Tree - Bag of Words Error Analysis
Accuracy
Number of words in each document
Test
Train


Decision Tree - N-Gram Error Analysis
Accuracy
Number of words in each document
Test
Train


Decision Tree - Tfidf Error Analysis
Accuracy
Number of words in each document
Test
Train


K Nearest Neighbors - Bag of Words Error Analysis
Accuracy
Number of words in each document
Test
Train

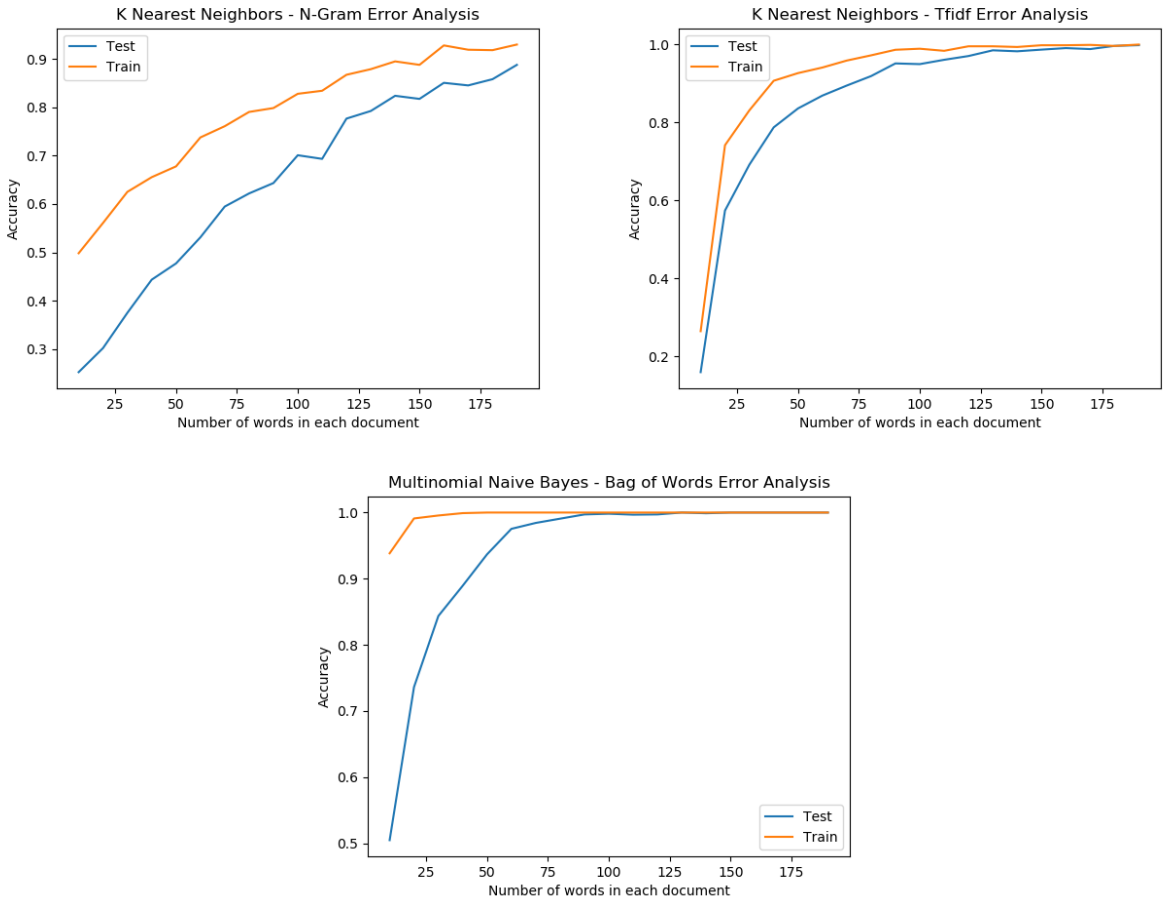


Figure 5-1 Learning Curves

Considering the above plots, It is evident that there is no overfitting, and after n of 50. The precision for the testing database and training database are close enough.

6 Conclusion

Finally, to conclude, our champion model is Bag of Words Multinomial Naive Bayes. We have given users the ability to enter a part of the book to see the related author. It is shown below:

```
Enter the text from the book: thinking the dangers above less dreadful than
   ...: those below, did not hesitate to throw overboard even their most useful
The Author is ['Jules Verne']
```

Figure 6-1 Result of Author Identification by a part of the book entered by a user