[DTI5125 [EG] Data Science Applications
Group 4, Assignment 1, Text Classification]

[AbdelMageed Ahmed AbdelMageed Hassan]

[Mohamed Sayed AbdelWahab Hussien]

[Sarah Hossam AbdelHameed Elmowafy]

[Ahmed Shehata Mahmoud AboMooustafa]

Abstract

Text classification is one of the major tasks of AI in general, NLP in particular. Having five Gutenberg books, this report discusses the methodologies and models with different transformation techniques that have been applied to reach the best accuracy that the champion model achieves by correctly classifying unseen text to the corresponding book.

Introduction

Text can be a rich source of information, however extracting insights from it is not an easy task due to its unstructured nature. The overall objective is to produce classification predictions and compare them; analyze the pros and cons of algorithms and generate and communicate the insights so, the implementation is needed to check the accuracy of each model going to be used and select the champion model. the best language to be used in such problems is python with its vast libraries.

Dataset

The Gutenberg dataset represents a corpus of over 60,000 book texts, their authors and titles. The data has been scraped from the Project Gutenberg website using a custom script to parse all bookshelves. we have taken five different samples of Gutenberg digital books that are of five different authors, that we think are of the criminal same genre and are semantically the same. The books are Criminal Sociology by Enrico Ferri. Crime: Its Cause and Treatment by Clarence Darrow. The Pirates' Who's Who by Philip Gosse. Celebrated Crimes by Alexandre Dumas and Buccaneers and Pirates of Our Coasts by Frank Richard.

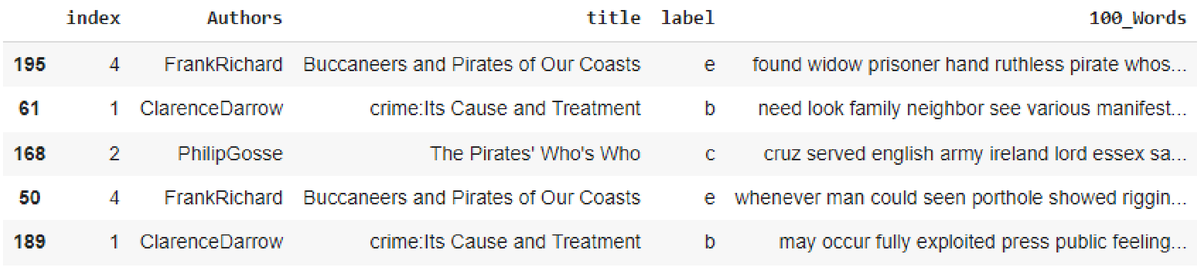
Books

The Pirates' Who's Who
Giving Particulars Of The Lives and Deaths Of The Pirates And Buccaneers
Philip Gosse
Project Gutenberg

Data Preparation:

- First step is reading the books from Gutenberg's library.
- Data preprocessing:
  - Removing stop-words and garbage characters using regular expression, also this pattern "[a-zA-Z]{3,}" ensures that words with less than 3 characters like "ye" are removed as they have no special meaning.
  - Converting all words to the lower case.
  - Lemmatization is the next step as it carries out the morphological analysis of the words
- Data Partitioning, partition each book into 200 documents, each document is a 100-word record.
- Data labeling as follows:
  1. Criminal Sociology → a
  2. Crime: Its Cause and Treatment → b
  3. The Pirates' Who's Who → c
  4. Celebrated Crimes by → d
  5. Buccaneers and Pirates of Our Coasts → e

Each author, Book Title, label and 100-word Records are combined in data frame as follows.



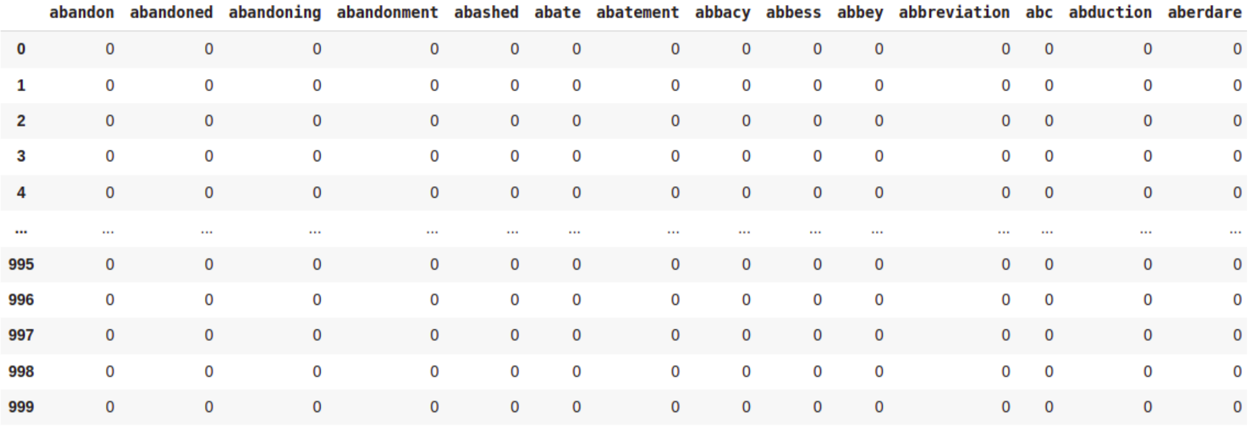
	index	Authors	title	label	100_Words
195	4	FrankRichard	Buccaneers and Pirates of Our Coasts	e	found widow prisoner hand ruthless pirate whos...
61	1	ClarenceDarrow	crime:Its Cause and Treatment	b	need look family neighbor see various manifest...
168	2	PhilipGosse	The Pirates' Who's Who	c	cruz served english army ireland lord essex sa...
50	4	FrankRichard	Buccaneers and Pirates of Our Coasts	e	whenever man could seen porthole showed riggin...
189	1	ClarenceDarrow	crime:Its Cause and Treatment	b	may occur fully exploited press public feeling...

Feature Engineering and Methodology

Text transformation using BOW, TF-IDF, N-gram, and word Embedding.

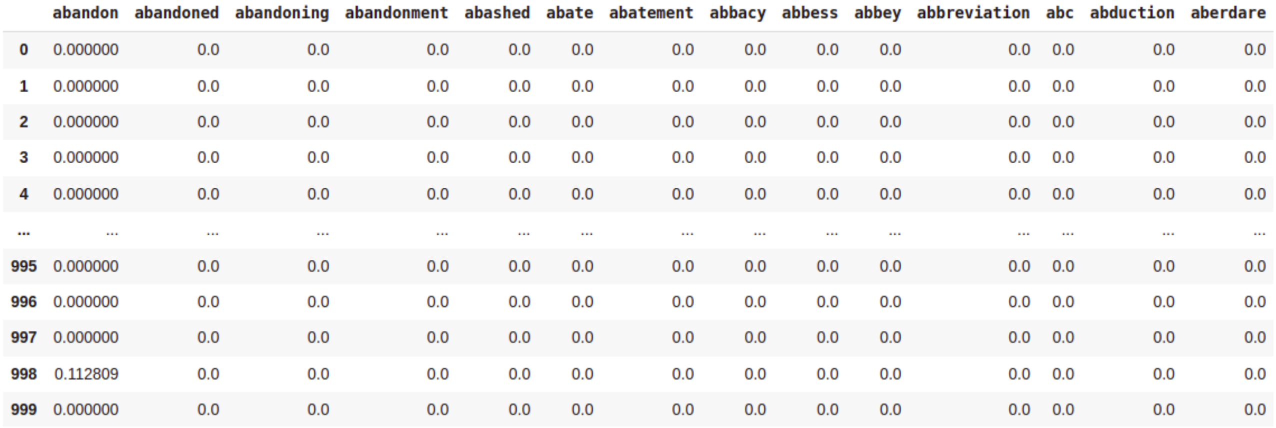
- **BOW**: It is a representation of text that describes the occurrence of words within a document, it involves two things:

    1. A vocabulary of known words.
    2. A measure of the presence of known words.



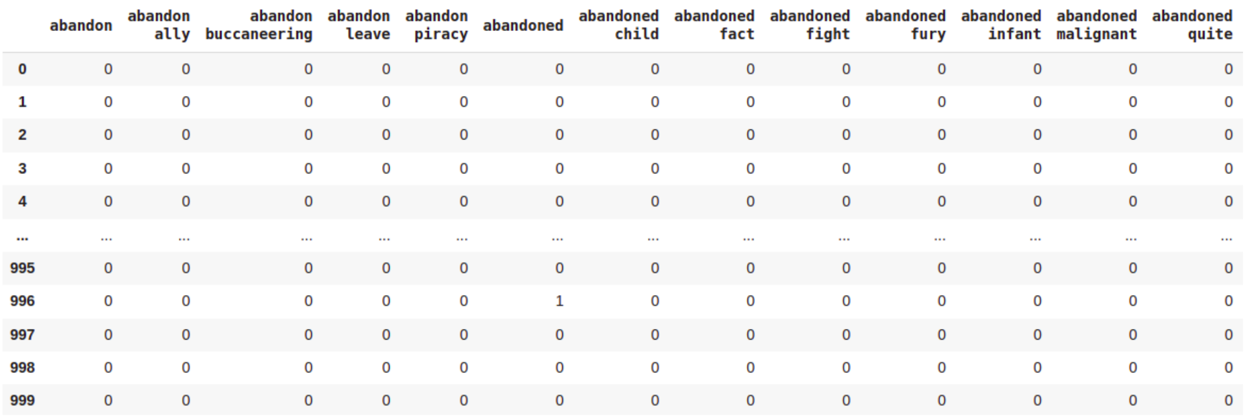
	abandon	abandoned	abandoning	abandonment	abashed	abate	abatement	abbacy	abbess	abbey	abbreviation	abc	abduction	aberdare
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	0	0	0	0	0	0	0	0	0	0	0	0	0	0
996	0	0	0	0	0	0	0	0	0	0	0	0	0	0
997	0	0	0	0	0	0	0	0	0	0	0	0	0	0
998	0	0	0	0	0	0	0	0	0	0	0	0	0	0
999	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- **TF-IDF:** a technique to quantify words in a set of documents. We compute a score for each word to signify its importance in the document and corpus.



	abandon	abandoned	abandoning	abandonment	abashed	abate	abatement	abbacy	abbess	abbey	abbreviation	abc	abduction	aberdare
0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
996	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
997	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
998	0.112809	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
999	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

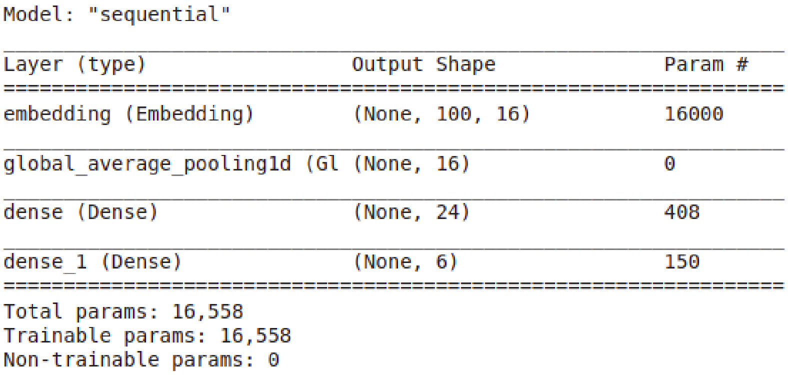
- **N-Grams**
  We use it because it preserves the meaning of words and sentences
  In our model, we used unigram and bigram.



	abandon	abandon ally	abandon buccaneering	abandon leave	abandon piracy	abandoned	abandoned child	abandoned fact	abandoned fight	abandoned fury	abandoned infant	abandoned malignant	abandoned quite
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	0	0	0	0	0	0	0	0	0	0	0	0	0
996	0	0	0	0	0	1	0	0	0	0	0	0	0
997	0	0	0	0	0	0	0	0	0	0	0	0	0
998	0	0	0	0	0	0	0	0	0	0	0	0	0
999	0	0	0	0	0	0	0	0	0	0	0	0	0

Word Embedding:

Is a type of word representation that allows words with similar meaning to have a similar representation.



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 100, 16)           16000
_________________________________________________________________
global_average_pooling1d (Gl (None, 16)                0
_________________________________________________________________
dense (Dense)                (None, 24)                408
_________________________________________________________________
dense_1 (Dense)              (None, 6)                 150
=================================================================
Total params: 16,558
Trainable params: 16,558
Non-trainable params: 0
```

Classification

For each technique of the above, these following models are trained and tested.

1. SVM
2. KNN
3. Decision Tree (DT)
4. For word Embedding, we built our deep neural with network and trained with our text data using embedding layer.

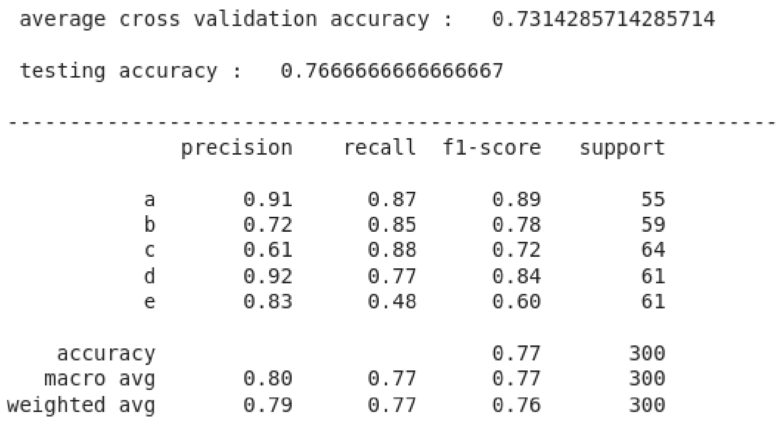
This table shows the accuracy resulting from combining different classifiers with different transformations.

Accuracy of each model			
	SVM	**KNN**	**DT**
BOW	0.96	0.953	0.80
TF-IDF	0.983	0.93	0.786
N-gram	0.956	0.92	0.793
Deep Learning (Word Embedding)			0.94
Cross Validation			
	SVM	**KNN**	**DT**
BOW	0.978	0.958	0.761
TF-IDF	0.978	0.954	0.771
N-gram	0.971	0.909	0.775

Champion Model

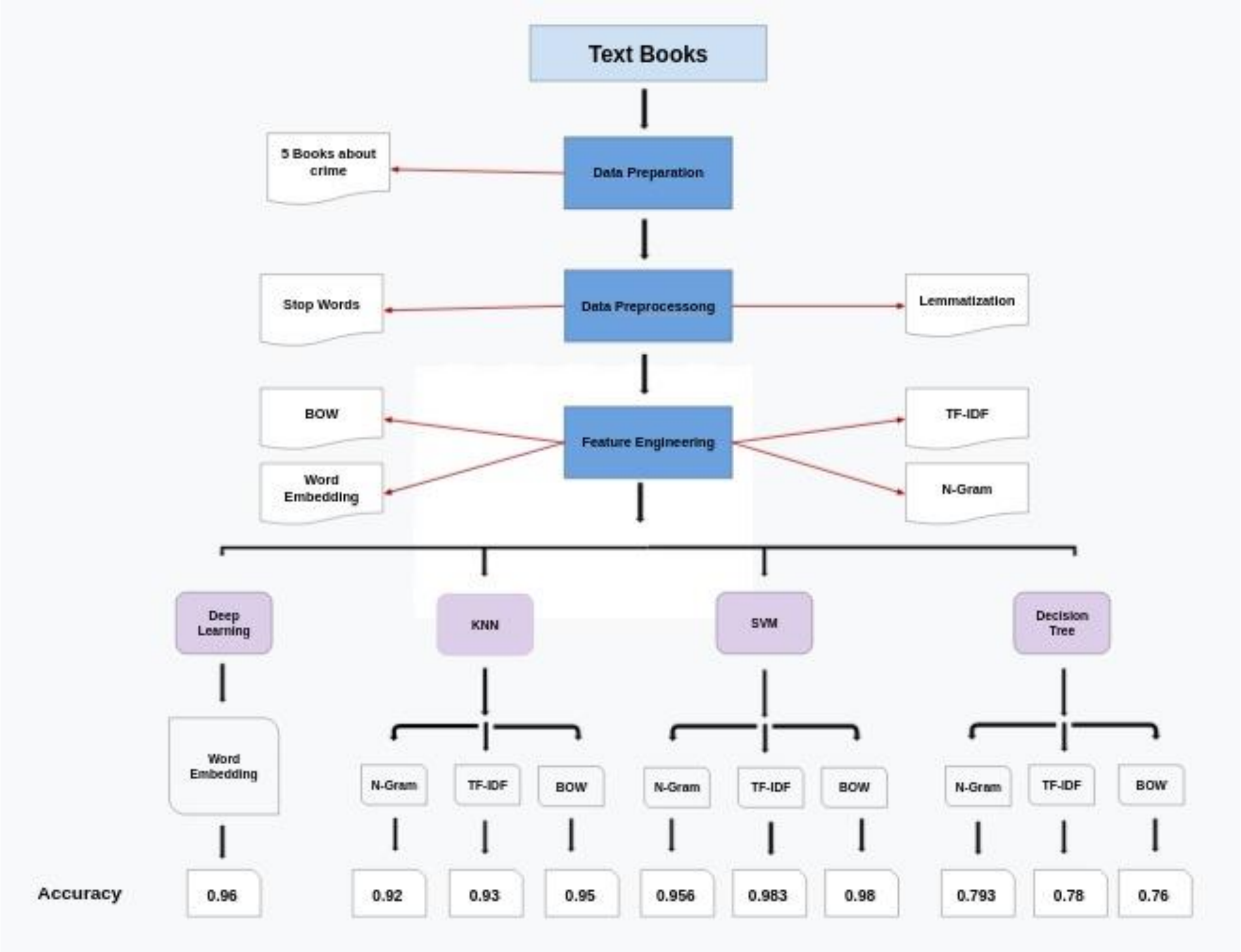
- Based on our models, the best model here is SVM based on TF-IDF with accuracy **0.983**, so it's our champion model.
  - Default parameters are used, just change kernel to **'sigmoid'**
  - Testing accuracy is **0.986**
  - cross validation accuracy is **0.98**

- When we change the kernel to poly and the max_iter from -1 to 3 the model accuracy decreased to 0.76
  - Kernel Poly we use poly instead of sigmoid, which is a more generalized representation of the linear kernel. It is less efficient and accurate.
  - Max_iter we make it 3 to limit the number of model iteration.



```
 average cross validation accuracy :   0.7314285714285714

 testing accuracy :   0.7666666666666667

----------------------------------------------------------------
              precision    recall  f1-score   support

           a       0.91      0.87      0.89        55
           b       0.72      0.85      0.78        59
           c       0.61      0.88      0.72        64
           d       0.92      0.77      0.84        61
           e       0.83      0.48      0.60        61

    accuracy                           0.77       300
   macro avg       0.80      0.77      0.77       300
weighted avg       0.79      0.77      0.76       300
```

System workflow:



Model Evaluation

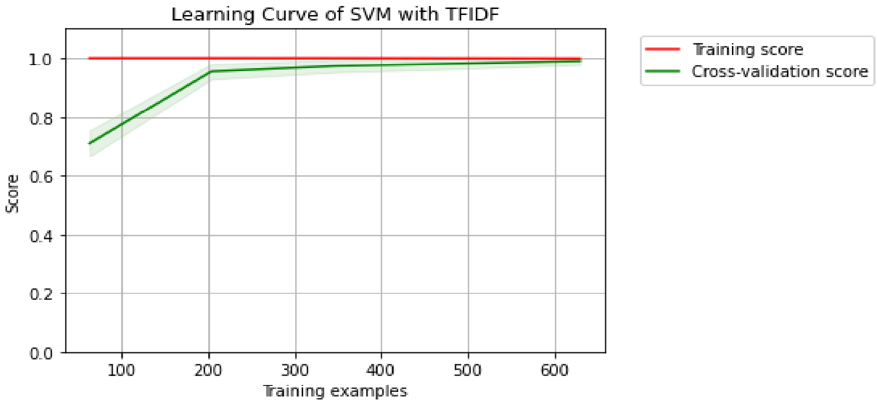
Cross-Validation

Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample to estimate the skill of a machine learning model on unseen data.

Applying ten-fold cross validation to the champion model (SVM based on TF_IDF) gives the following accuracies [1, 0.98 ,0.98, 0.98, 1, 0.95 ,1 ,0.98 ,1 ,1] with average cross validation accuracy 0.982 and 100% accuracy in the training set and testing accuracy of 0.976. These small differences indicate that the model fits the data correctly without Overfitting nor Underfitting

Learning curve

Another way to get an estimate of model's generalization performance is Learning Curves, it shows the model performance on training and validation sets as a function of training set size (training iterations), Ideal Learning Curve Model generalizes on testing and training data. the smaller the gap between the training and cross-validation scores, the better the model in generalization.



If the training and cross-validation scores converge together as more data is being added, then the model will probably not benefit from increasing the data. If the training score is much greater than the validation score, then the model probably requires more training examples in order to generalize more effectively [1].

Bias and Variability

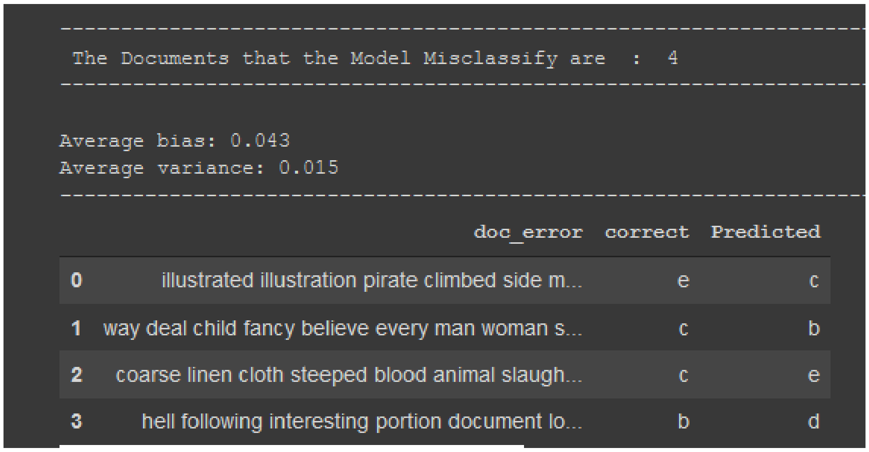
The curves are plotted with the mean scores, however variability during cross-validation is shown with the shaded light green areas that represent a standard deviation above and below the mean for all cross-validations.

If the model suffers from error due to bias, then there will be more likely to variability around the training score curve. If it suffers from errors due to variance, then there will be more variability around the cross validated score [1].

SVM Based on TF_IDF model have low average bias and variability as shown in the previous figure, by using mlxtend.evaluate.bias_variance_decomp module we get average bias 0.043 and average variance: 0.015. this guarantee avoiding underfitting and overfitting as the model perform well on the training set it also generalizes well according to the cross-validation metrics

Error Analysis of Champion Model

SVM based on TF_IDF misclassifies only 4 documents out of 300 documents in the testing set. By giving this document a close look, we found that the model misclassifies Two documents of label C as E and B, One document of label E as C, and one document of label B as D.



```
------------------------------------------------------------------
 The Documents that the Model Misclassify are  :  4
------------------------------------------------------------------

Average bias: 0.043
Average variance: 0.015
------------------------------------------------------------------
```

	doc_error	correct	Predicted
0	illustrated illustration pirate climbed side m...	e	c
1	way deal child fancy believe every man woman s...	c	b
2	coarse linen cloth steeped blood animal slaugh...	c	e
3	hell following interesting portion document lo...	b	d

These documents have common words in the actual and the predicted books , So the model classifies them to the class that has a higher number of words in common with that document ,

as the model had trained on TF_IDF, it doesn't understand that the sequence of some words or sentences is related to a specific class , So if a class C document has one word with high TF_IDF value, like Marker Word which indicates that this word belongs to Class C and also has several words with TF_IDF values a little bit lower than the Marker Word and these words refers class B. then it assigns this document to Class B.

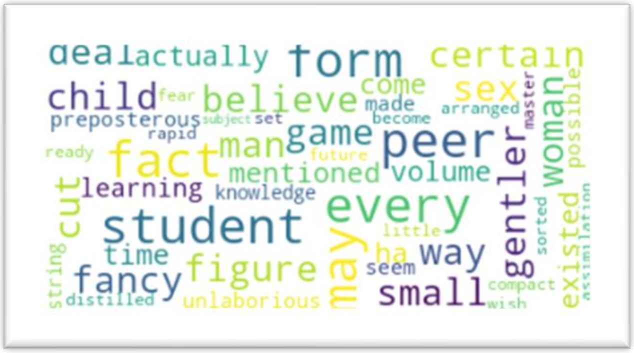
Another case that if specific words have high TF value in the document and TF value in any book rather than the actual class, the model will label it to the other class.

For example, the next document is incorrectly predicted as B while its actual label is C.

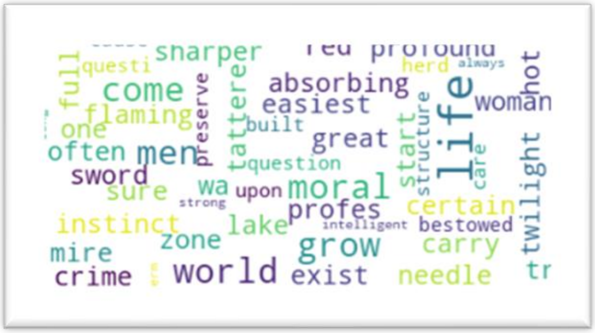
" way deal child fancy believe every man woman since certain gentler sex cut small figure game mentioned volume actually existed time come every form learning however preposterous may seem made unlaborious possible would student knowledge string fact arranged sorted distilled set compact form ready rapid assimilation little fear student may wish future become master subject delve original source search fact date surely pirate taking broadest sense much entitled biographical dictionary clergyman race horse artist ferro concrete assured medical men directory lawyer list peer peerage book record name particular musician schoolmaster stockbroker saint bookmaker dare say average adjuster almanac peer "

By plotting the word cloud for this document and the two Labels it indicates that two words (Man and Woman) have high TF value in the document and Label B and low TF in Label C, while it has the same IDF value for Both B and C so the model will label it as B while its actual class is C.

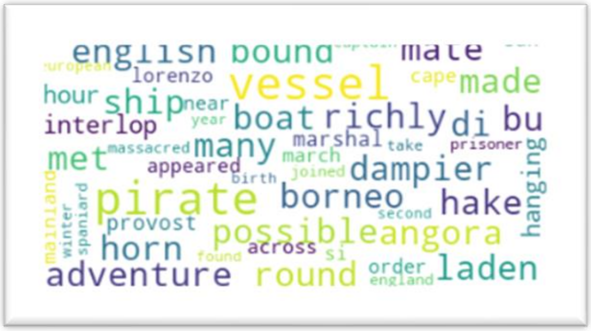
Document that misclassified



predicted class (B)



Actual Class (C)



In the end, text classification problems depend not on your model design but the quality of your data and how you process it as "input garbage output garbage", without all the preprocessing steps we would have obtained worse accuracy but with applying the best model with the clean data we obtained a more than satisfying accuracy.

References

https://www.scikit-yb.org/en/latest/api/model_selection/learning_curve.htm