# ASSIGNMENT 1

Group 4

## ABSTRACT

The overall objective of this assignment was to produce classification predictions and compare them; analyze pros and cons of algorithms and generate and communicate the insights.

## AUTHORS:

Ifiok Charles (0300041771)
Kalada Reagan Tuwotamuno (8489637)
Stevy Kuimi (8845324)
Gayatri (300095631)

## SUPERVISOR:

Prof. Arya Rahgozar

GNG 5125 X

# Contents

**Introduction** ........ 3

**Data preparation** ........ 3

**Data** ........ 3

**Data preparation, preprocessing and cleansing** ........ 3

**Feature Engineering** ........ 4

**Modelling** ........ 5

**Models:** Three models were used to perform the task: SVM, decision tree and K-NN ........ 5

**Performance:** ........ 5

**Error Analysis** ........ 11

Figure 1 Performance measurement Formula ..................... 6

Figure 2 Confusion matrix for SVM with BOW ..................... 6

Figure 3 Performance measurement for SVM with BOW ..................... 6

Figure 4 Confusion matrix for SVM with TF-IDF ..................... 7

Figure 5 performance measurement for SVM with TF-IDF ..................... 7

Figure 6 Confusion matrix for decision tree with BOW ..................... 7

Figure 7 Performance measurement for decision tree with BOW ..................... 8

Figure 8  Confusion matrix for decision tree with TF-IDF ..................... 8

Figure 9 Performance measurement for decision tree with TF-IDF ..................... 8

Figure 10 Confusion matrix for K-NN with BOW ..................... 9

Figure 11 Performance measurement for K-NN with BOW ..................... 9

Figure 12   Confusion matrix for K-NN with TF-IDF ..................... 9

Figure 13 performance measurement for K-NN with TF-IDF ..................... 10

Figure 14 Ten Fold cross validation ..................... 10

Figure 15 Champion without bi-grams ..................... 11

Figure 16 performance measurement of the champion with bi-grams ..................... 11

Figure 17 prediction -vs- actual with bi-grams on the champion ..................... 12

Figure 18 Performance measurement with bigram on the champion ..................... 12

Figure 19 Challenger without Bi-grams ..................... 13

Figure 20 Error location ..................... 13

Figure 21 Author 1 document 353 ..................... 14

Figure 22 word cloud for author 4 with multiple documents ..................... 14

# Introduction

The overall objective of this assignment was to produce classification predictions and compare them; analyze pros and cons of algorithms and generate and communicate the insights.

# Data preparation

### Data

The task of this assignment is to classify segments of books by authors. This was done by choosing five random books from different genre from the Gutenberg online libraries. They were chosen in the categories of mythology, music, engineering, physics and poetry. The table below shows the names of the authors of each book chosen.

| **Genre** | **Authors** |
|---|---|
| Mythology | Sir James George Frazer |
| Music | Magaret Blake-Alverson |
| Engineering | Charles M. Hortons |
| Physics | Amos Emerson Dolbear |
| Poetry | John Milton |

### Data preparation, preprocessing and cleansing

These are the steps we followed in writing our code:

- Upload the data to the data frame,
- Tokenize the data to make sure the characters are separated,
- Remove stop words to lower the dimensional space and help with the collocation
- Stemming: Basically, this convert words into their root form by reducing the difference between their inflected forms,
- create random samples of 200 documents of each book, each sample with a 150 words.
- Create a corpus, meaning putting all the document samples in the same data frame and proceed further cleaning, which involves only keeping letters of the alphabet from a-z.

Removing all numbers, white spaces and punctuations, and putting all the words into lower cases. This was necessary for the predictor to avoid confusion.

## Feature Engineering

- **BOW:** Bags of Words. This is a representation of text that describes the occurrence of words within a document. This is the most popular technique used to convert categorical features to numerical ones.
- **TF-IDF:** Term Frequency-Inverse Document Frequency: this is another way or technique used for occurrence of words. TF-IDF measures relevance, not frequency. The TF part divides the number of occurrences of each word in the document by the total number of words in the document, while the IDF does the downscaling of weight for words that occur in many documents in the corpus. For example, if the words like ‘the’, ’and’ appears in all documents, those will be systematically discounted. Each word’s TF-IDF relevance is a normalized data format that also adds up to one.

| | **BAG OF WORDS** | **TF-IDF** |
|---|---|---|
| Advantages | • Easy to understand<br>• Easy to implement | • suitable in comparing two documents<br>• suitable for long document |
| Dis-advantages | • Not suitable for long documents<br>• Do not consider the semantic relation between words<br>• Curse of dimensionality | • Less informative for assessing occurrence in long document<br>• The dependence on BOW is a liability |

# Modelling

**Models:** Three models were used to perform the task: SVM, decision tree and K-NN

- **SVM:** The Support Vector Machine is a classification technique that produces very high prediction accuracy. It distinctly classifies the data points to find hyperplane for n-dimensional space.
- **Decision tree:** Decision tree is one of the predictive modeling approaches used in statistics, data mining and machine learning. The goal is to predicts the value of target variable based on several input variables.
- **K-NN:** standing for K-Nearest Neighbors is a widely used classification technique in the industry mainly because it is easy to implement, easy to interpret results and has a low computational time. In this assignment, the optimal value of K was taken from the grid search after several trials and errors.

**Performance:**

to be able to compare the model's performance, the confusion matrices and the precision graphs showing the scores values were extracted. The performance measurements data are calculated automatically using the confusing matrices from the code, but for better explanation and understanding, it can be done using the formula shown on Figure 1 Performance measurement Formula.

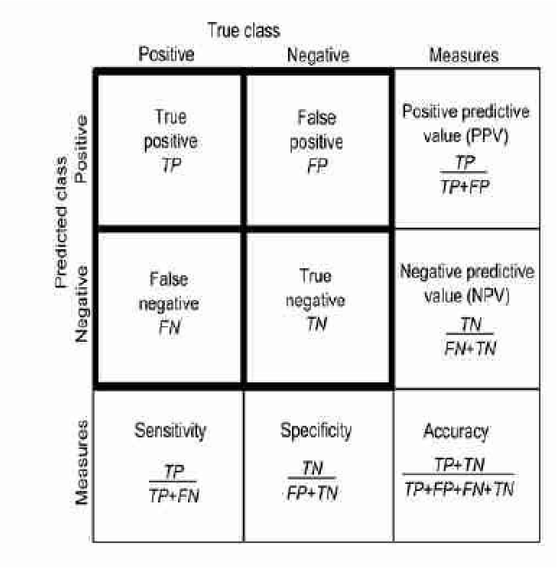
The figures below show the confusion matrices and performance measurements for the different models used with each feature, SVM ( Figure 2 Confusion matrix for SVM with BOW, Figure 3 Performance measurement for SVM with BOW, Figure 4 Confusion matrix for SVM with TF-IDF, Figure 5 performance measurement for SVM with TF-IDF), decision Tree (Figure 6 Confusion matrix for decision tree with BOW, Figure 7 Performance measurement for decision tree with BOW, Figure 8 Confusion matrix for decision tree with TF-IDF, Figure 9 Performance measurement for decision tree with TF-ID) and K-NN (Figure 10 Confusion matrix for K-NN with BOW Figure 11 Performance measurement for K-NN with BOW Figure 12 Confusion matrix for K-NN with TF-IDF Figure 13 performance measurement for K-NN with TF-IDF)

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

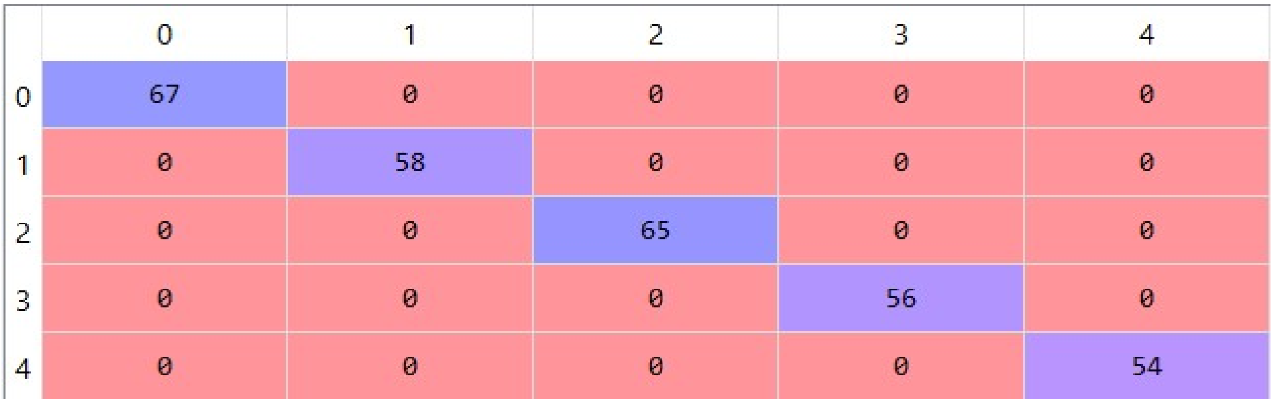
$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$



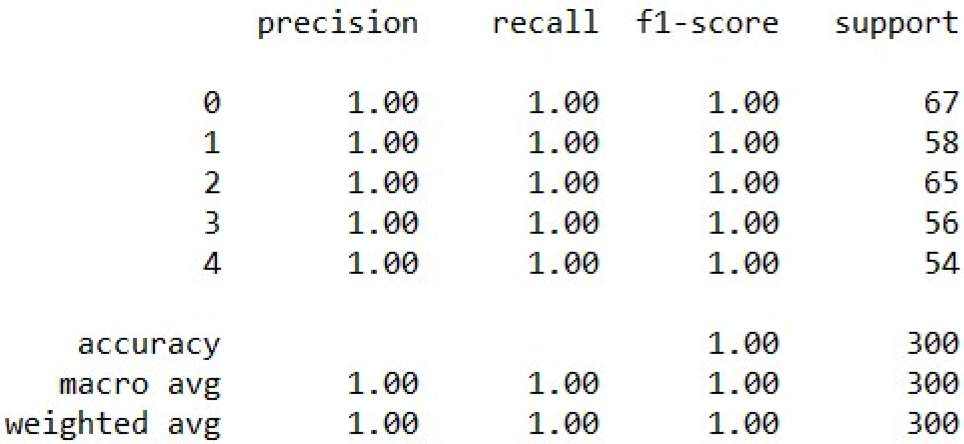
| | | True class | | |
|---|---|---|---|---|
| | | Positive | Negative | Measures |
| Predicted class | Positive | True positive<br>TP | False positive<br>FP | Positive predictive value (PPV)<br>$\frac{TP}{TP+FP}$ |
| | Negative | False negative<br>FN | True negative<br>TN | Negative predictive value (NPV)<br>$\frac{TN}{FN+TN}$ |
| Measures | | Sensitivity<br>$\frac{TP}{TP+FN}$ | Specificity<br>$\frac{TN}{FP+TN}$ | Accuracy<br>$\frac{TP+TN}{TP+FP+FN+TN}$ |

*Figure 1 Performance measurement Formula*



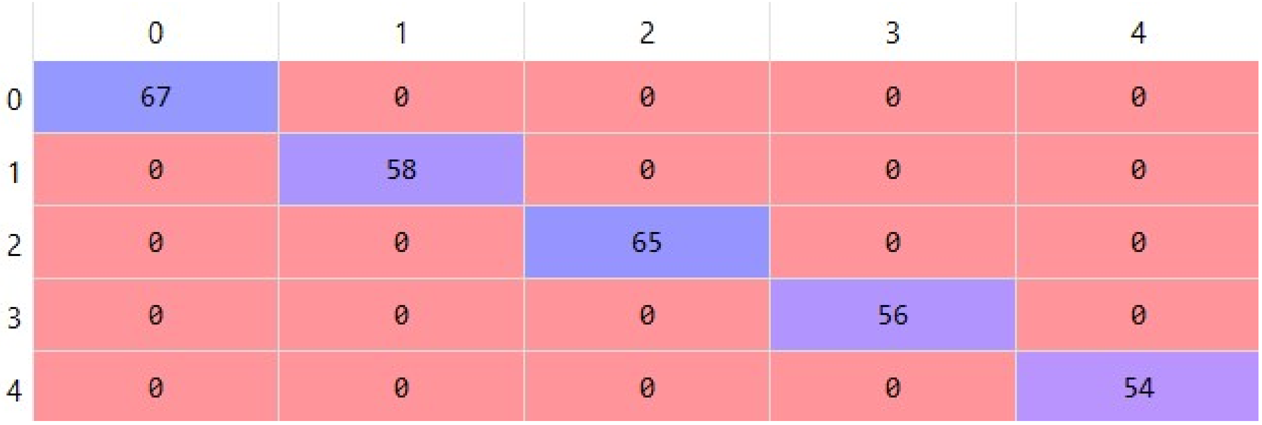
| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 67 | 0 | 0 | 0 | 0 |
| 1 | 0 | 58 | 0 | 0 | 0 |
| 2 | 0 | 0 | 65 | 0 | 0 |
| 3 | 0 | 0 | 0 | 56 | 0 |
| 4 | 0 | 0 | 0 | 0 | 54 |

*Figure 2 Confusion matrix for SVM with BOW*



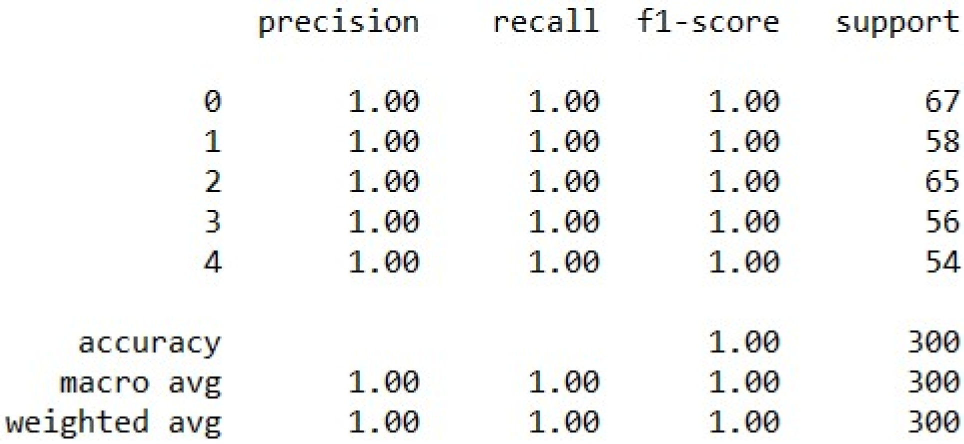
```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        67
           1       1.00      1.00      1.00        58
           2       1.00      1.00      1.00        65
           3       1.00      1.00      1.00        56
           4       1.00      1.00      1.00        54

    accuracy                           1.00       300
   macro avg       1.00      1.00      1.00       300
weighted avg       1.00      1.00      1.00       300
```

*Figure 3 Performance measurement for SVM with BOW*



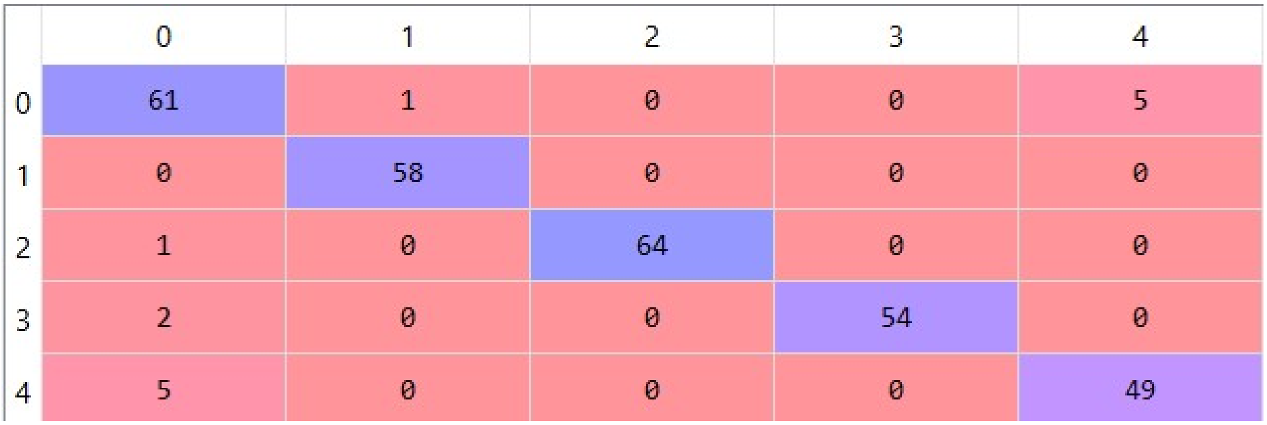
| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 67 | 0 | 0 | 0 | 0 |
| 1 | 0 | 58 | 0 | 0 | 0 |
| 2 | 0 | 0 | 65 | 0 | 0 |
| 3 | 0 | 0 | 0 | 56 | 0 |
| 4 | 0 | 0 | 0 | 0 | 54 |

*Figure 4 Confusion matrix for SVM with TF-IDF*



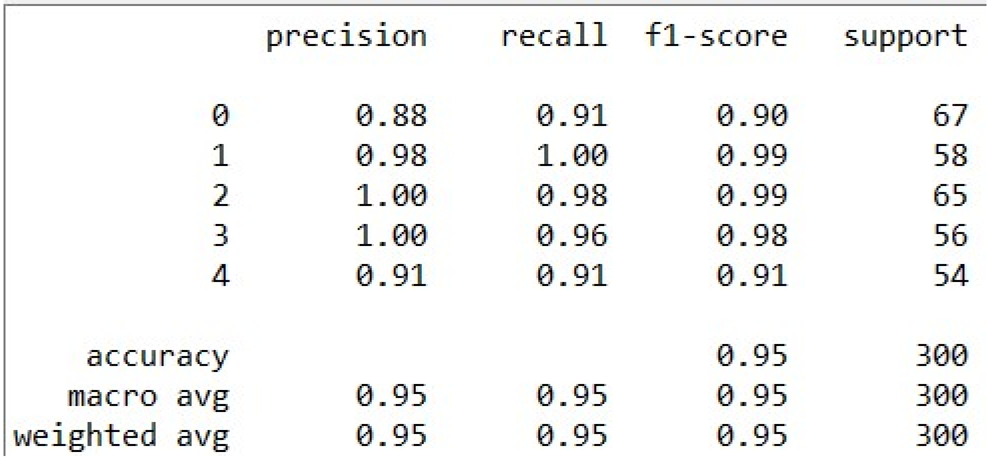
```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        67
           1       1.00      1.00      1.00        58
           2       1.00      1.00      1.00        65
           3       1.00      1.00      1.00        56
           4       1.00      1.00      1.00        54

    accuracy                           1.00       300
   macro avg       1.00      1.00      1.00       300
weighted avg       1.00      1.00      1.00       300
```

*Figure 5 performance measurement for SVM with TF-IDF*



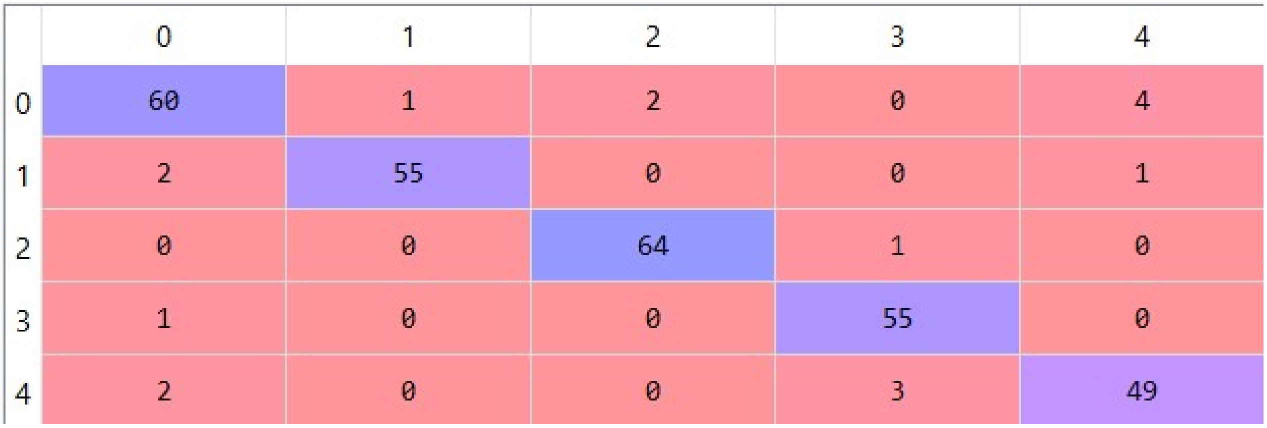
| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 61 | 1 | 0 | 0 | 5 |
| 1 | 0 | 58 | 0 | 0 | 0 |
| 2 | 1 | 0 | 64 | 0 | 0 |
| 3 | 2 | 0 | 0 | 54 | 0 |
| 4 | 5 | 0 | 0 | 0 | 49 |

*Figure 6 Confusion matrix for decision tree with BOW*



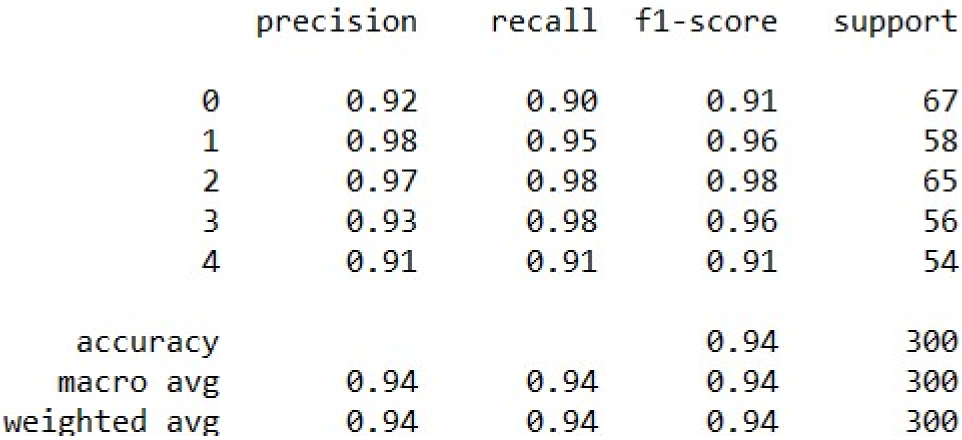
```
              precision    recall  f1-score   support

           0       0.88      0.91      0.90        67
           1       0.98      1.00      0.99        58
           2       1.00      0.98      0.99        65
           3       1.00      0.96      0.98        56
           4       0.91      0.91      0.91        54

    accuracy                           0.95       300
   macro avg       0.95      0.95      0.95       300
weighted avg       0.95      0.95      0.95       300
```

*Figure 7 Performance measurement for decision tree with BOW*



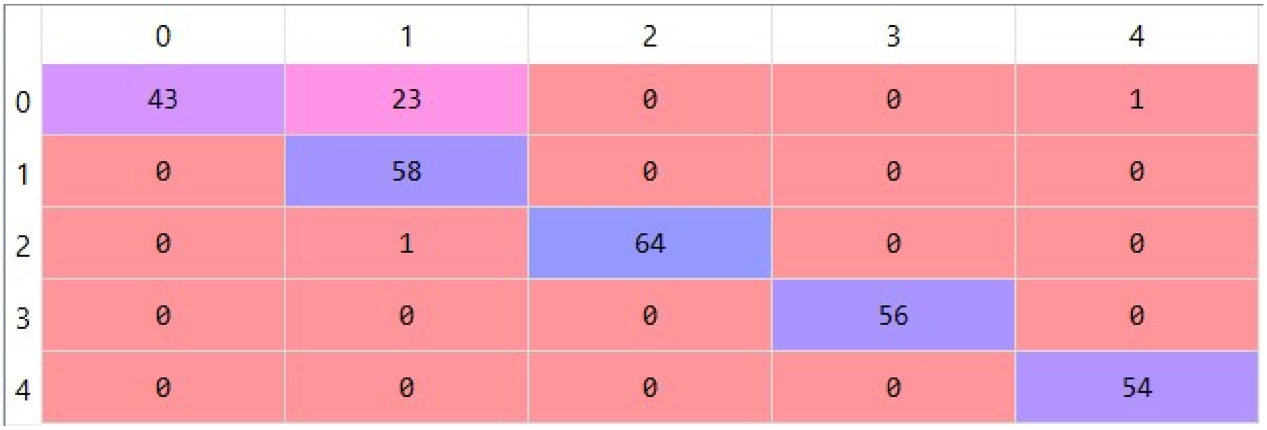
| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 60 | 1 | 2 | 0 | 4 |
| 1 | 2 | 55 | 0 | 0 | 1 |
| 2 | 0 | 0 | 64 | 1 | 0 |
| 3 | 1 | 0 | 0 | 55 | 0 |
| 4 | 2 | 0 | 0 | 3 | 49 |

*Figure 8  Confusion matrix for decision tree with TF-IDF*



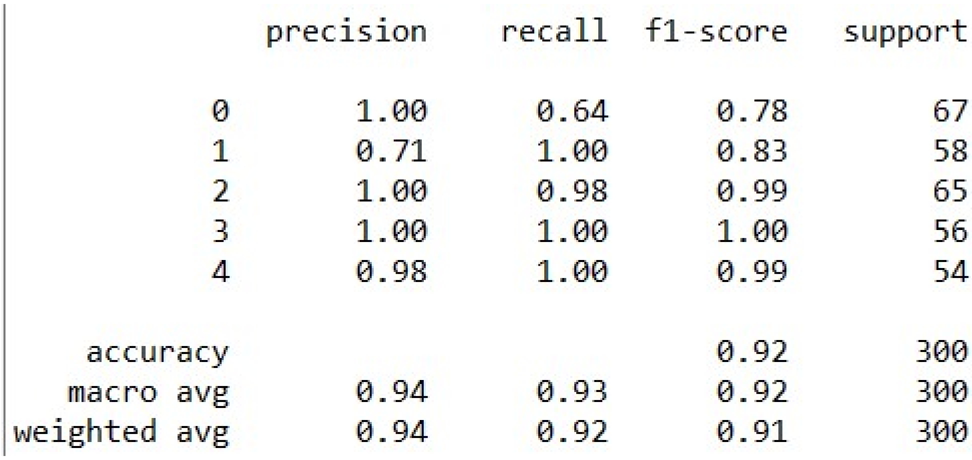
```
              precision    recall  f1-score   support

           0       0.92      0.90      0.91        67
           1       0.98      0.95      0.96        58
           2       0.97      0.98      0.98        65
           3       0.93      0.98      0.96        56
           4       0.91      0.91      0.91        54

    accuracy                           0.94       300
   macro avg       0.94      0.94      0.94       300
weighted avg       0.94      0.94      0.94       300
```

*Figure 9 Performance measurement for decision tree with TF-IDF*



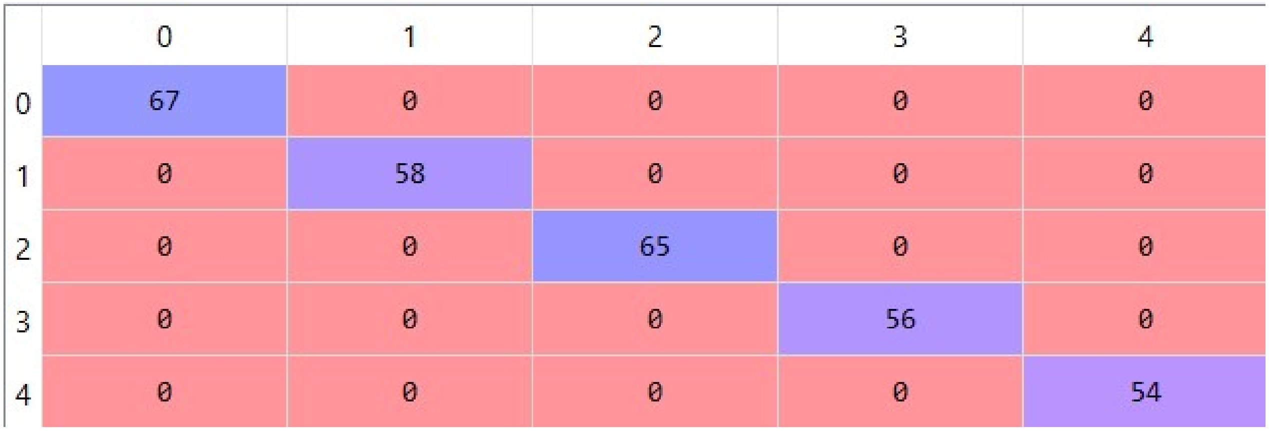
| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 43 | 23 | 0 | 0 | 1 |
| 1 | 0 | 58 | 0 | 0 | 0 |
| 2 | 0 | 1 | 64 | 0 | 0 |
| 3 | 0 | 0 | 0 | 56 | 0 |
| 4 | 0 | 0 | 0 | 0 | 54 |

*Figure 10 Confusion matrix for K-NN with BOW*



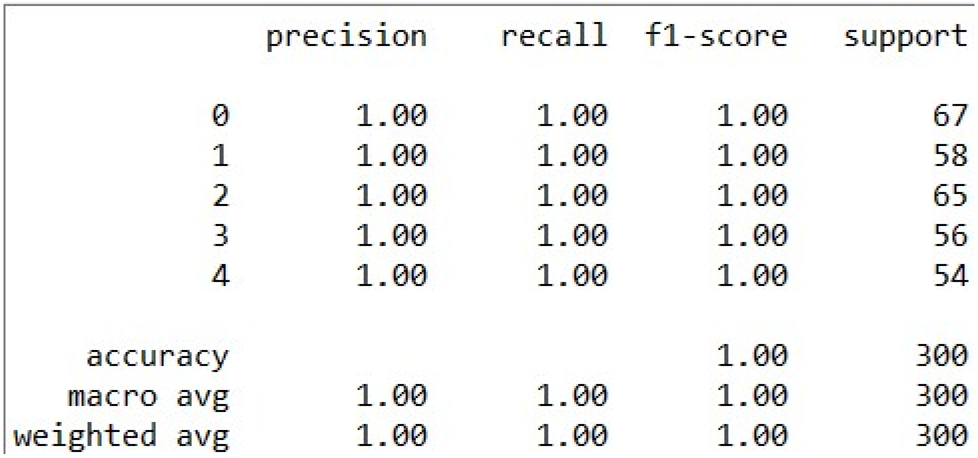
```
              precision    recall  f1-score   support

           0       1.00      0.64      0.78        67
           1       0.71      1.00      0.83        58
           2       1.00      0.98      0.99        65
           3       1.00      1.00      1.00        56
           4       0.98      1.00      0.99        54

    accuracy                           0.92       300
   macro avg       0.94      0.93      0.92       300
weighted avg       0.94      0.92      0.91       300
```

*Figure 11 Performance measurement for K-NN with BOW*



| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 67 | 0 | 0 | 0 | 0 |
| 1 | 0 | 58 | 0 | 0 | 0 |
| 2 | 0 | 0 | 65 | 0 | 0 |
| 3 | 0 | 0 | 0 | 56 | 0 |
| 4 | 0 | 0 | 0 | 0 | 54 |

*Figure 12   Confusion matrix for K-NN with TF-IDF*



```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        67
           1       1.00      1.00      1.00        58
           2       1.00      1.00      1.00        65
           3       1.00      1.00      1.00        56
           4       1.00      1.00      1.00        54

    accuracy                           1.00       300
   macro avg       1.00      1.00      1.00       300
weighted avg       1.00      1.00      1.00       300
```

*Figure 13 performance measurement for K-NN with TF-IDF*

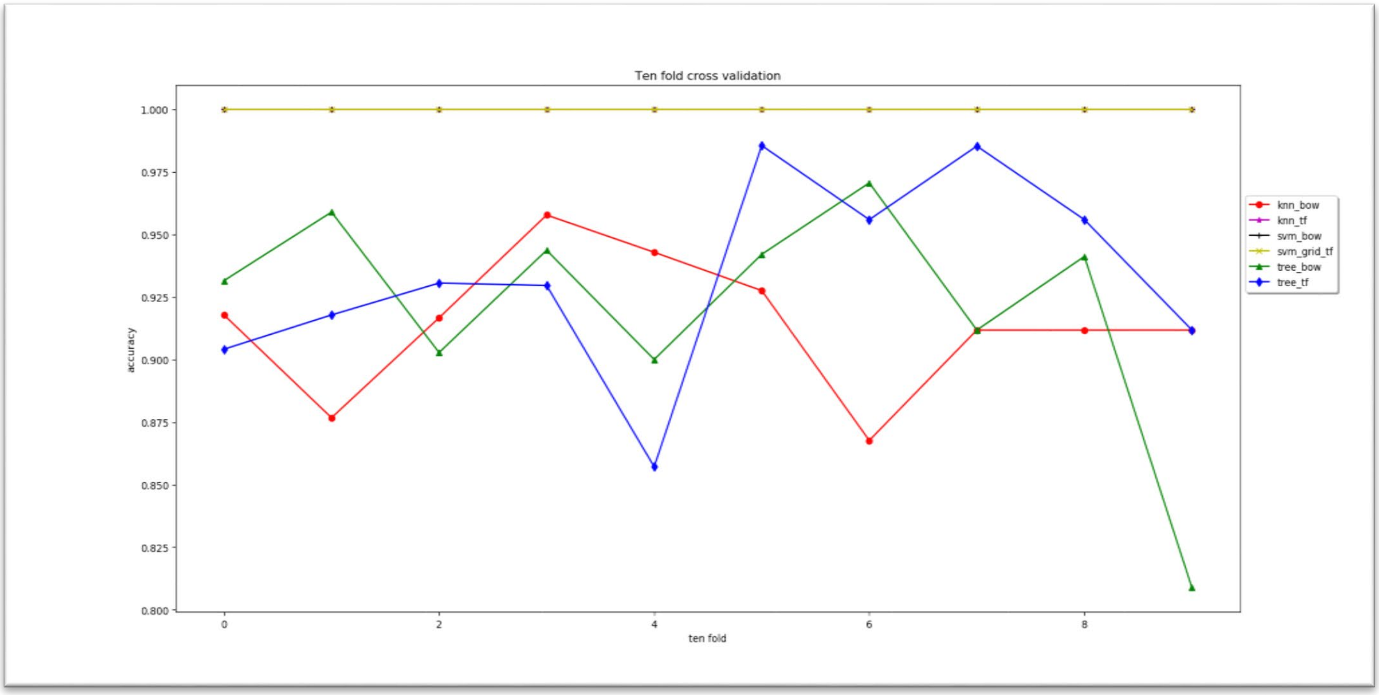


*Figure 14 Ten Fold cross validation*

Based on the performance measurements data, it can be noted that SVM with both BOW and TF-IDF (Figure 3 Performance measurement for SVM with BOW, Figure 5 performance measurement for SVM with TF-IDF), and K-NN with TF-IDF (Figure 13 performance measurement for K-NN with TF-IDF) have a 100% accuracy compared to other models used. The tenfold graph above displays that even better. It can be observed that SVM and K-NN with TF-IDF have the highest accuracy compared to the decision tree and K-NN with Bow. Given the fact that SVM with BOW was easier to implement and did not require an extra work to find the optimal parameter as needed to perform the SVM with TF-IDF, it can be confidently concluded that the champion model is SVM with BOW. The challenger will be the decision tree with TF-IDF.

## Error Analysis

Initially, the champion had a very high accuracy without bi-grams as shown on Figure 15 Champion without bi-grams and Figure 16 performance measurement of the champion with bi-grams, giving no room for error analysis.

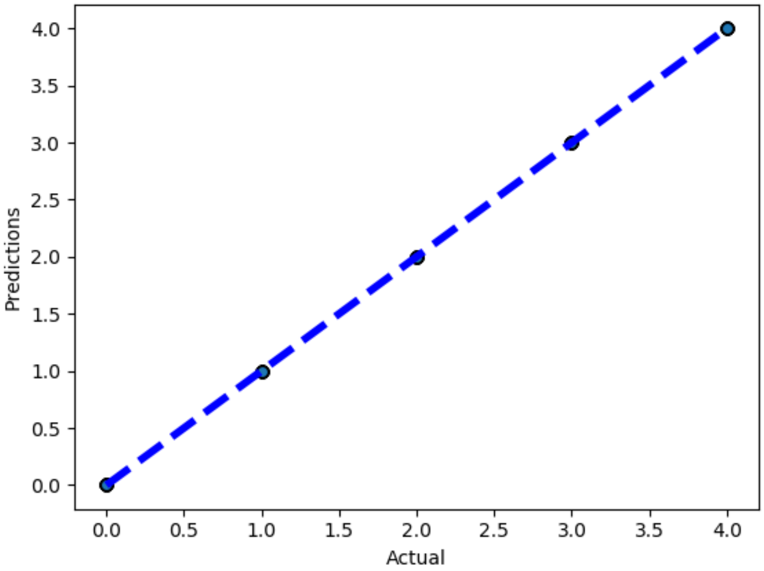
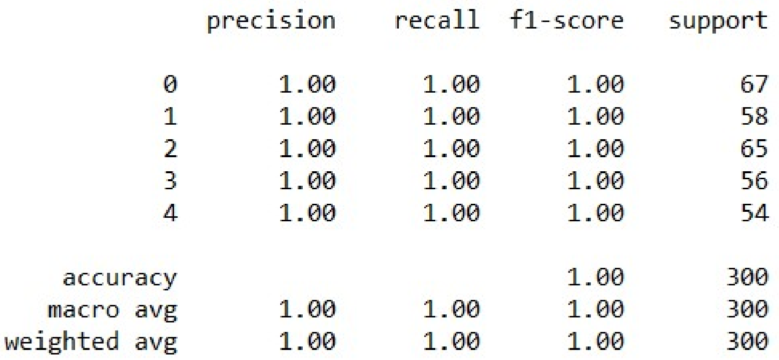


*Figure 15 Champion without bi-grams*



```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        67
           1       1.00      1.00      1.00        58
           2       1.00      1.00      1.00        65
           3       1.00      1.00      1.00        56
           4       1.00      1.00      1.00        54

    accuracy                           1.00       300
   macro avg       1.00      1.00      1.00       300
weighted avg       1.00      1.00      1.00       300
```

*Figure 16 performance measurement of the champion with bi-grams*

Then bi-grams were added and it reduced the accuracy drastically as shown on Figure 17 prediction -vs- actual with bi-grams on the championand Figure 18 Performance measurement with bigram on the champion. Therefore, it was concluded that the program was better without bigrams because the accuracy was low with all three models.

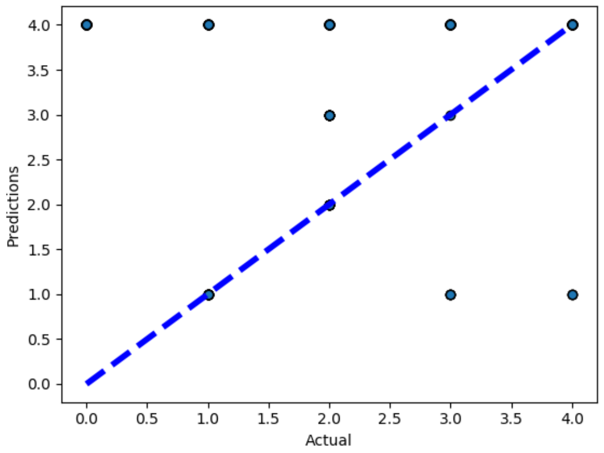
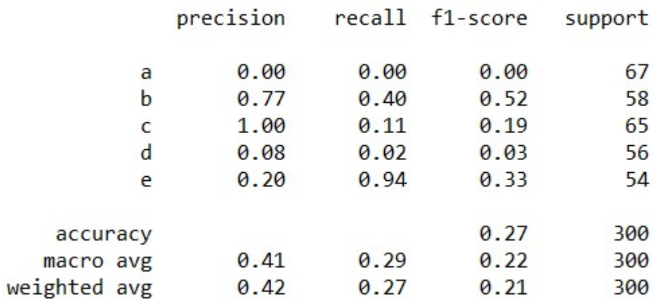


*Figure 17 prediction -vs- actual with bi-grams on the champion*



```
              precision    recall  f1-score   support

           a       0.00      0.00      0.00        67
           b       0.77      0.40      0.52        58
           c       1.00      0.11      0.19        65
           d       0.08      0.02      0.03        56
           e       0.20      0.94      0.33        54

    accuracy                           0.27       300
   macro avg       0.41      0.29      0.22       300
weighted avg       0.42      0.27      0.21       300
```

*Figure 18 Performance measurement with bigram on the champion*

Error analysis was then performed without bi-grams on the challenger model to locate the possible collocations which threw off the predictor (Figure 20 Error location, Figure 21 Author 1 document 353, Figure 22 word cloud for author 4 with multiple documents).

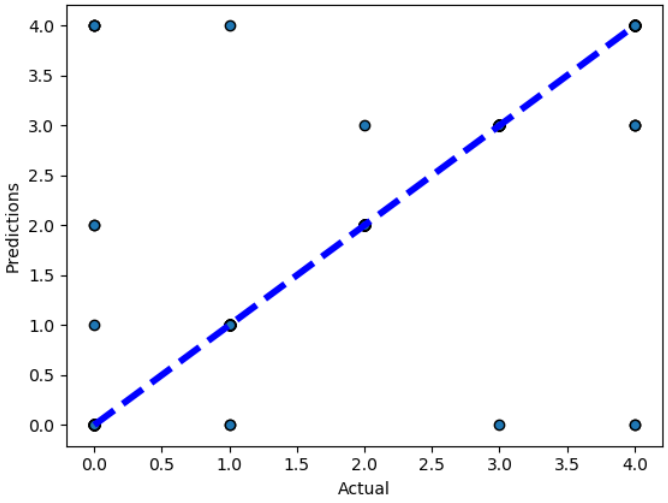
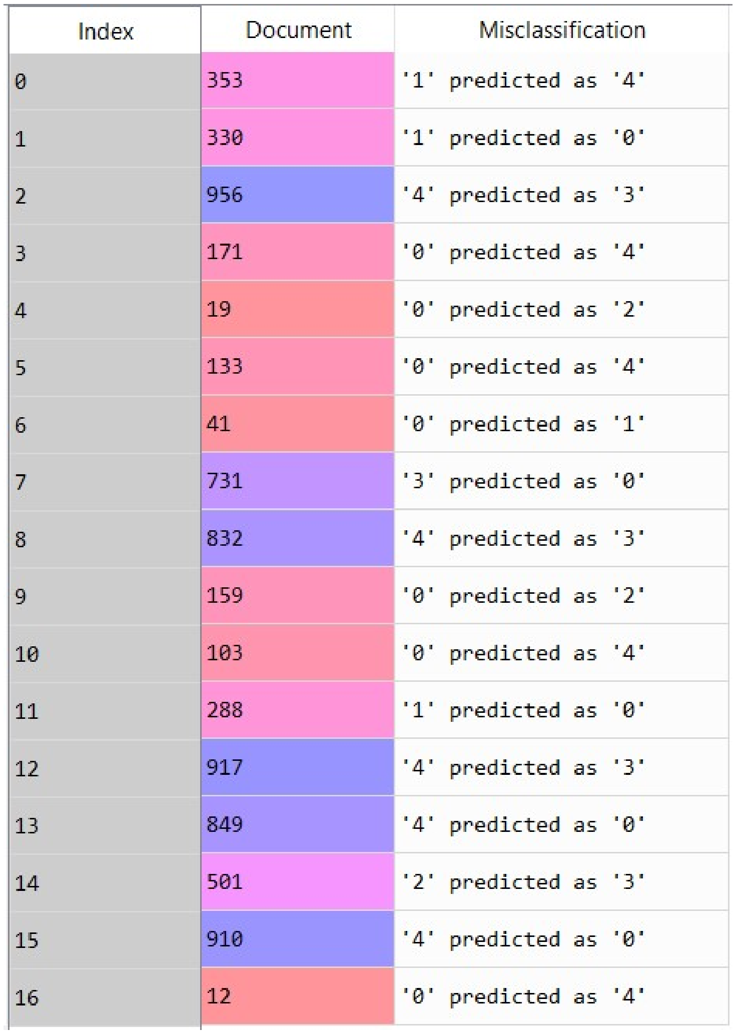


*Figure 19 Challenger without Bi-grams*



| Index | Document | Misclassification |
|---|---|---|
| 0 | 353 | '1' predicted as '4' |
| 1 | 330 | '1' predicted as '0' |
| 2 | 956 | '4' predicted as '3' |
| 3 | 171 | '0' predicted as '4' |
| 4 | 19 | '0' predicted as '2' |
| 5 | 133 | '0' predicted as '4' |
| 6 | 41 | '0' predicted as '1' |
| 7 | 731 | '3' predicted as '0' |
| 8 | 832 | '4' predicted as '3' |
| 9 | 159 | '0' predicted as '2' |
| 10 | 103 | '0' predicted as '4' |
| 11 | 288 | '1' predicted as '0' |
| 12 | 917 | '4' predicted as '3' |
| 13 | 849 | '4' predicted as '0' |
| 14 | 501 | '2' predicted as '3' |
| 15 | 910 | '4' predicted as '0' |
| 16 | 12 | '0' predicted as '4' |

*Figure 20 Error location*

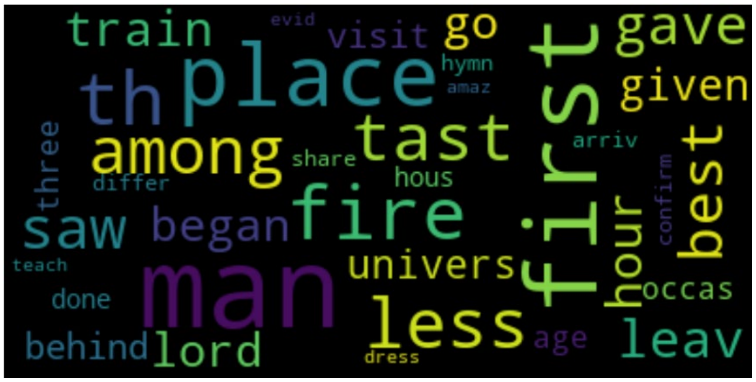


*Figure 21 Author 1 document 353*

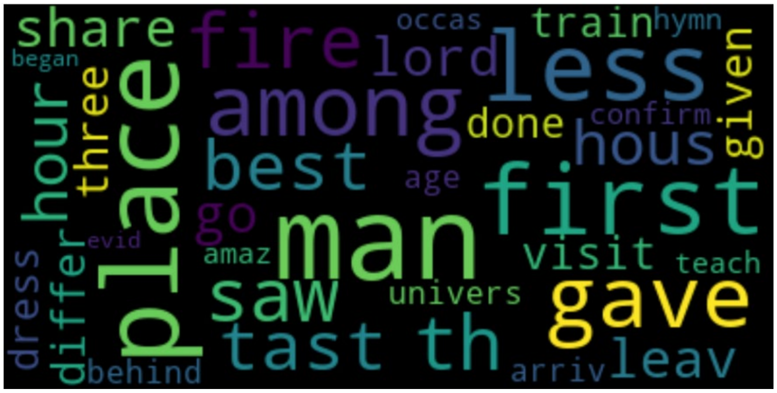


*Figure 22 word cloud for author 4 with multiple documents*

Figure 20 shows the document that was misclassified and also shows the wrong predictions. Figure 21 and 22 shows the collocations in the misclassified document which also occurs in multiple documents of the wrongly predicted author. For example, words like ‘man’, ‘first’ and ‘less’ occurred in the misclassified document and the multiple documents of the wrongly predicted author.