TEXT CLASSIFICATION

Assignment Group 1

Group Members:

- *Ahmed Nasser*
- *Abelrahman Ali*
- *Esraa Fayed*
- *Aya Metwally*

Group 5

6/4/2023

Contents

1. Introduction

In the project, we use five different samples of Gutenberg digital books with different authors. We train a machine learning model that can identify which author the document belongs to. The final target is to classify, and predict author of five mystery books for distinct authors.

2. Methodologies

2.1 Data Preparation and Preprocessing

We choose mystery genre for five distinctive authors. The books are:


The Man Who Was Thursday.txt


Dracula.txt


Crime and Punishment.txt


The murder of Roger Ackroyd.txt


The Moonstone.txt

The authors are:

1. Fyodor Dostoevsky
2. Wilkie Collins
3. G. K. Chesterton
4. Agatha Christie
5. Bram Stoker

We prepare the data by creating random samples of 200 documents of each book, representative of the source input. We prepare the records of 100 words records for each document and label them as ‘Fyodor Dostoevsky':'a', 'Wilkie Collins':'b', 'G. K. Chesterton':'c', 'Agatha Christie':'d', 'Bram Stoker':'e'. We then preprocess the data by removing stop-words and garbage characters. We also label authors as {0: 'Fyodor Dostoevsky', 1: 'Wilkie Collins', 2: 'G. K. Chesterton', 3: 'Agatha Christie', 4: 'Bram Stoker'}

	sentence	author	label
0	walked hundred pace beside workman called murd...	Fyodor Dostoevsky	a
1	disguised saw told clever hindoo people concea...	Wilkie Collins	b
2	lunch partook reflectively four small quaint c...	G. K. Chesterton	c
3	upon survive without widespread public support...	G. K. Chesterton	c
4	anything trapped would done woman find nothing...	Fyodor Dostoevsky	a
...	...	...	...
995	shop staring window man top hat loaded snow li...	G. K. Chesterton	c
996	silence secretary said curious strained know s...	G. K. Chesterton	c
997	delicately withdrew svidrigaïlov drew raskolni...	Fyodor Dostoevsky	a
998	major blunt raymond billiard think look see mi...	Agatha Christie	d
999	could trace drifted snow little professor sign...	Bram Stoker	e

1000 rows × 3 columns

Fig1: labellig authors as a, b, c, etc

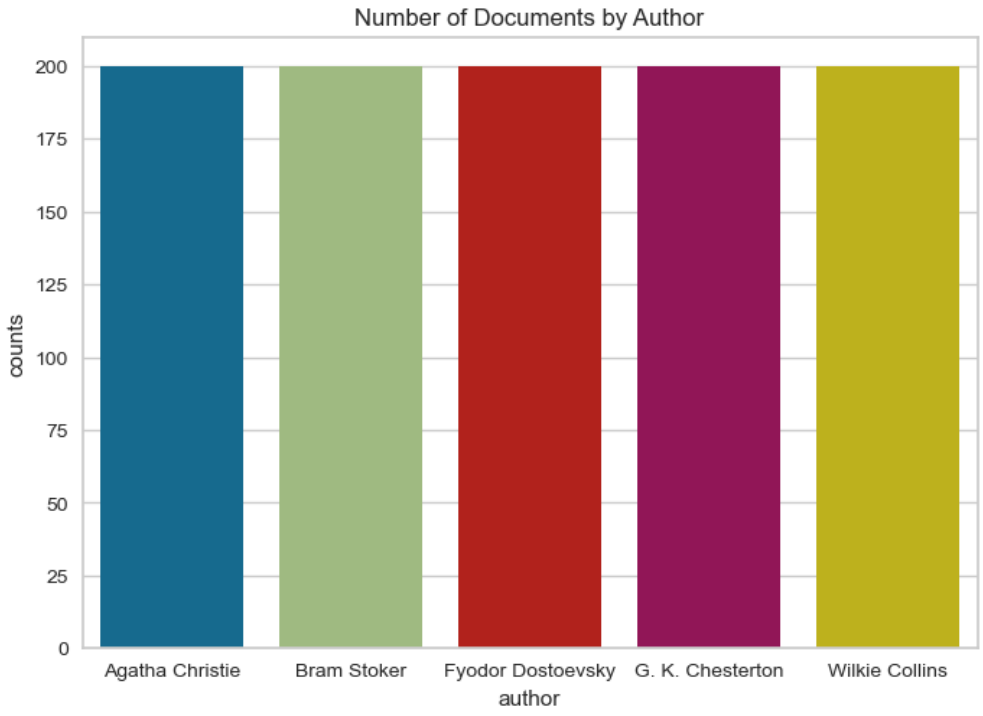


Fig2: Each author has 200 documents

2.2 Data Visualization

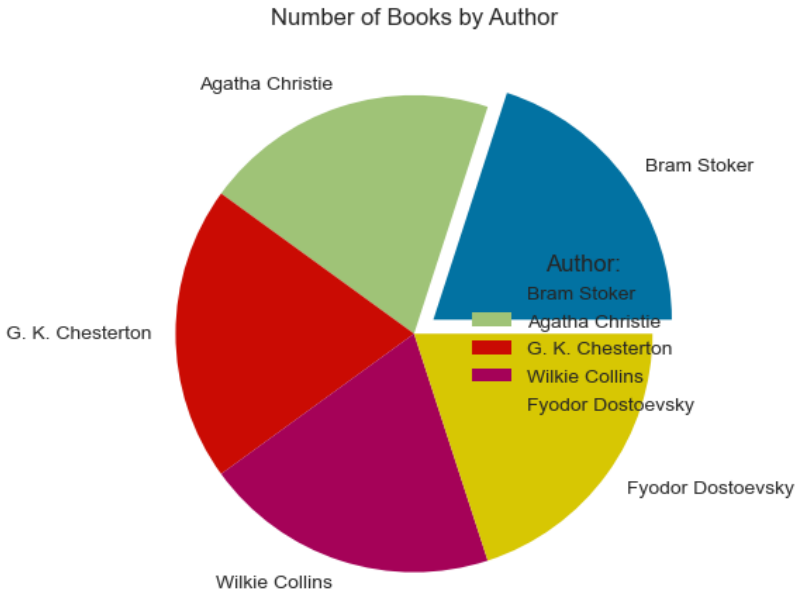


Fig3: Five Distincit Author

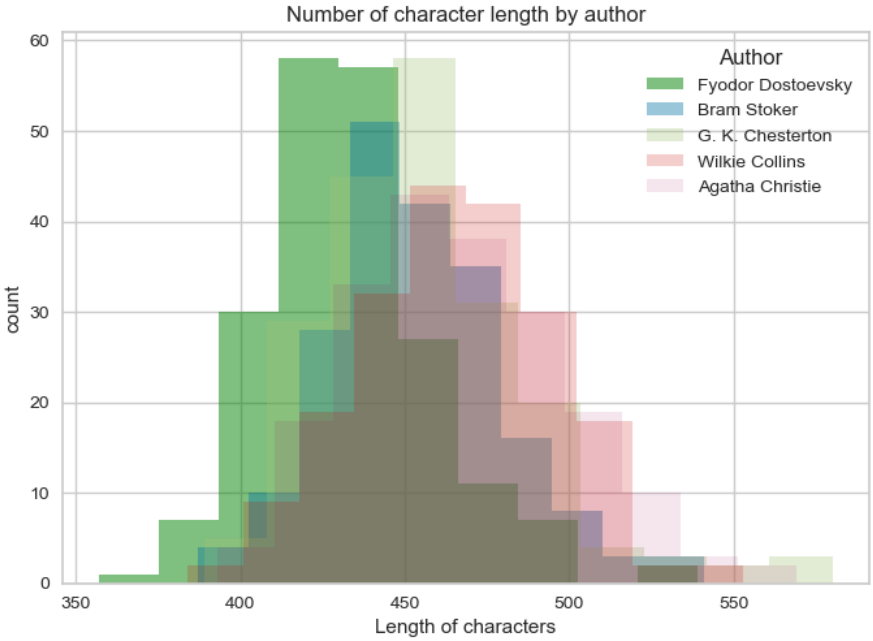


Fig4: Length of character by each author

Fig5: most common 200 words by author: G. K. Chesterton

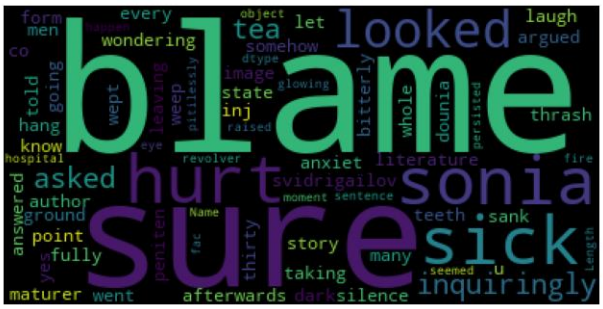


Fig6: most common 200 words by author: Fyodor Dostoevsky

Fig7: most common 200 words by author: Bram Stoker

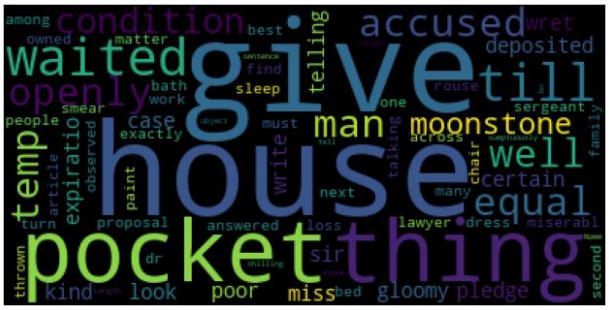
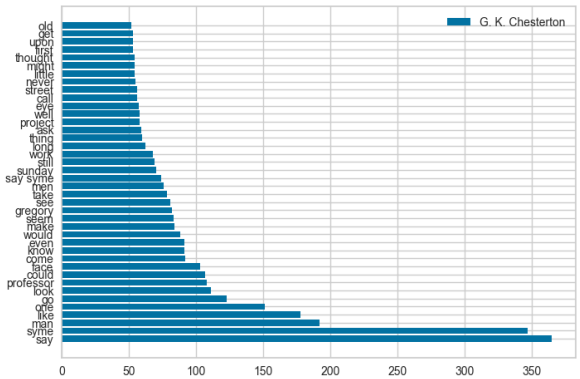
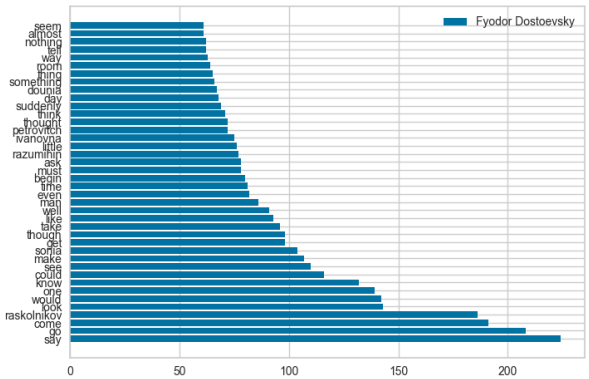


Fig8: most common 200 words by author: Wilkie Collins

Fig9: most common 200 words by author: Agatha Christie





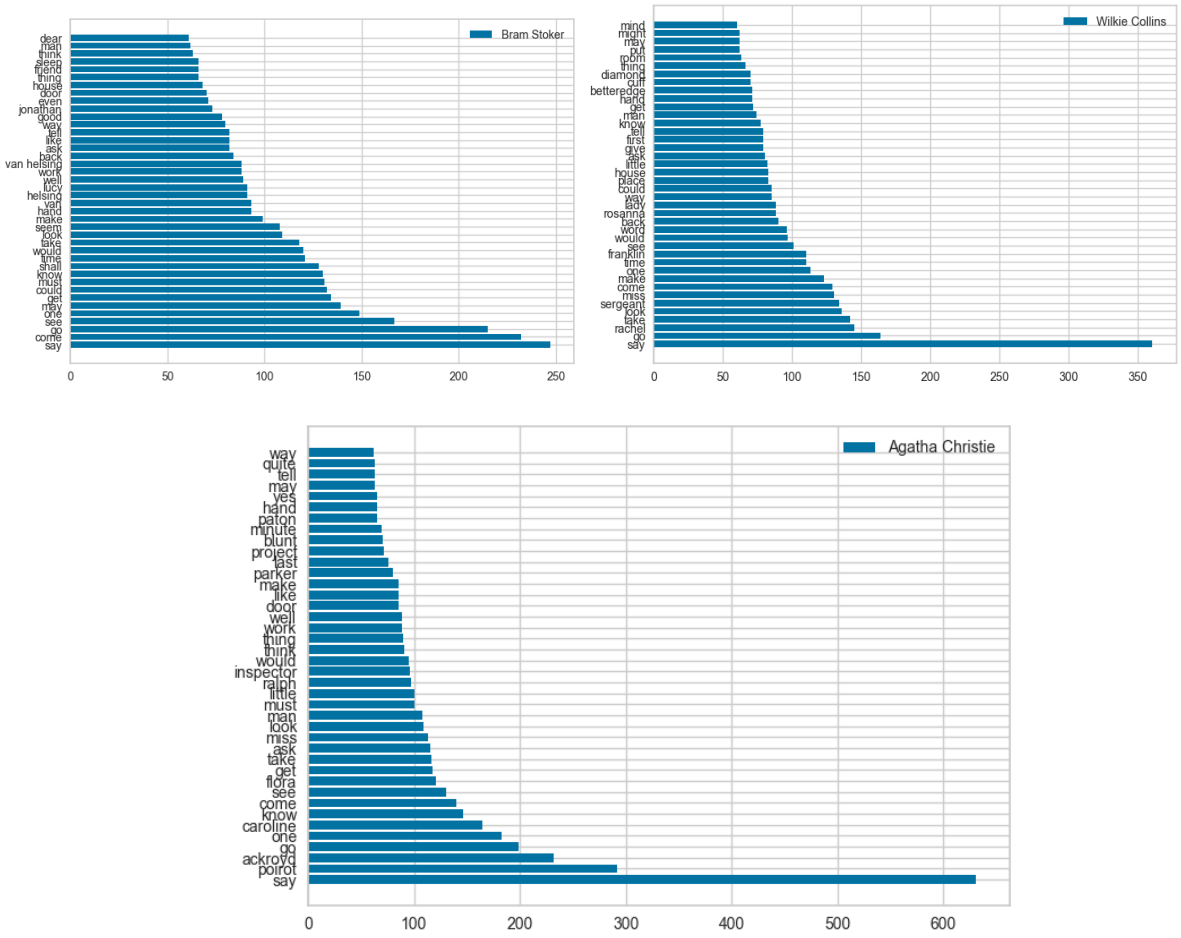


Fig10: top 40 word occurs in each book by author

2.3 Feature Engineering

We transform the data to BOW, TF-IDF, and n-gram. Each of these techniques has its own strengths and weaknesses, we apply each algorithm with every technique and compare results to choose the champion model.

2.3.1 BOW (Bag of Words)

A simple and popular technique for representing text data. It involves creating a dictionary of all the unique words in the text corpus and then representing each document as a vector of word frequencies. The order of words is not considered in this technique, only the presence or absence of words in a document is used to represent the text.

	aback	abandon	abandoned	abandoning	abandonment	abbey	abbot	abide	ability	able	...	zane	zaraïsk	zigzagged	zoo	zoological	zossimov	zoöp
0	0	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	0	0	
2	0	0	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	0	0	
4	0	0	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	0	0	
996	0	0	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	0	0	
998	0	0	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	0	0	

1000 rows × 8755 columns

Fig11: The transformation of BOW

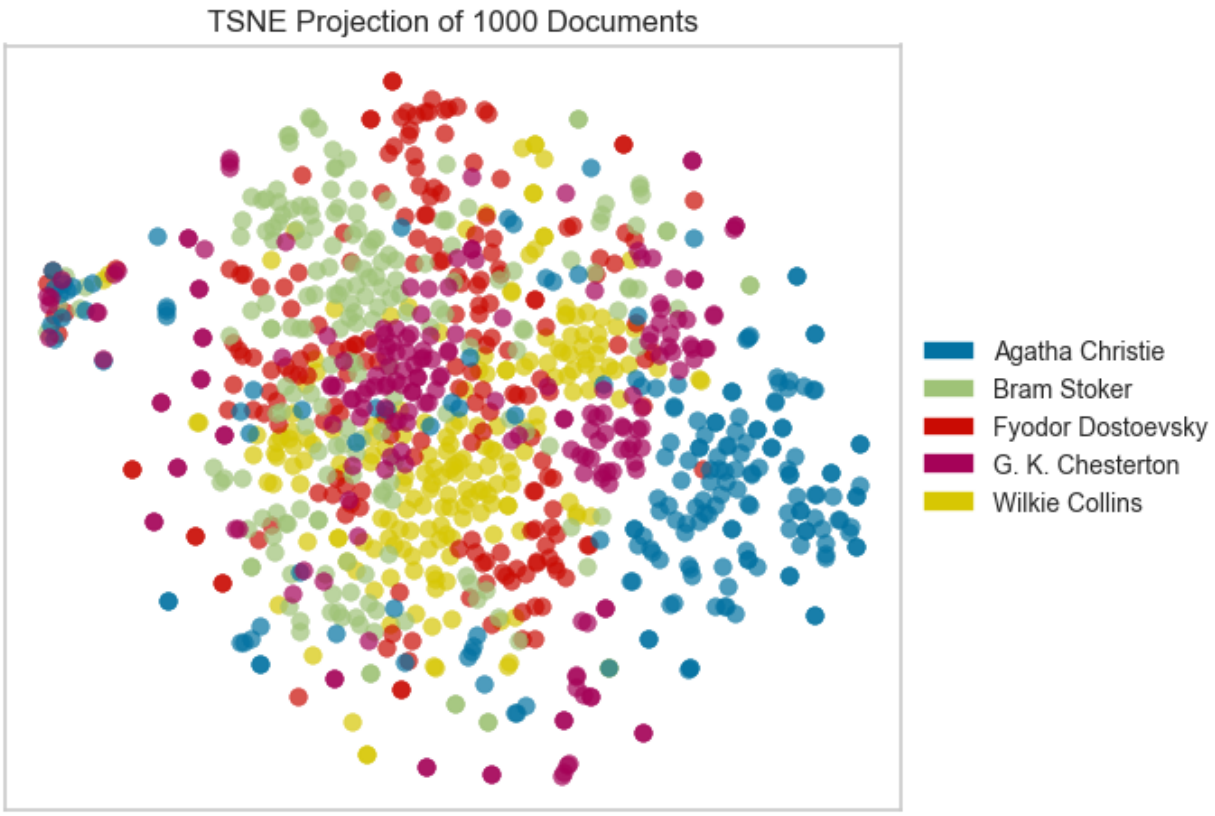


Fig12: The TSNE of BOW

2.3.2 TF-IDF (Term Frequency-Inverse Document Frequency)

A more advanced technique that takes into account the frequency of words in a document as well as their importance in the context of the entire corpus. It assigns a weight to each term in a document based on its frequency in the document and its rarity in the entire corpus. This technique helps to identify the most important words in a document.

	aback	abandon	abandoned	abandoning	abandonment	abbey	abbot	abide	ability	able	...	zane	zaraïsk	zigzagged	zoo	zoological	zossimov	zoöp
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	0.0	0.0	0.0	0.0	
1	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	0.0	0.0	0.0	
2	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	0.0	0.0	0.0	
3	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	0.0	0.0	0.0	
4	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	0.0	0.0	0.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
995	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	0.0	0.0	0.0	
996	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	0.0	0.0	0.0	
997	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	0.0	0.0	0.0	
998	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	0.0	0.0	0.0	
999	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	0.0	0.0	0.0	

1000 rows × 8755 columns

Fig13: The transformation of TF-IDF

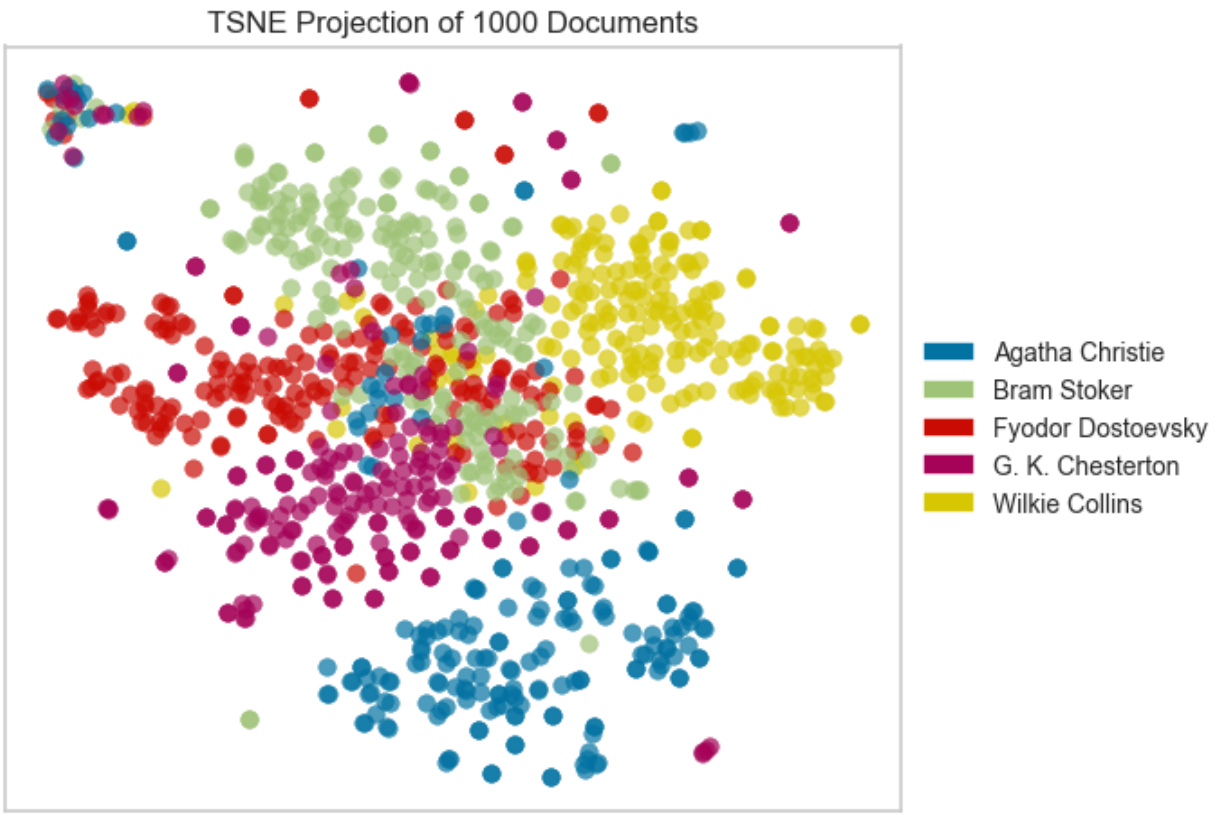


Fig14: The TSNE of TF-IDF

2.3.3 N-gram

A contiguous sequence of n items from a given sample of text or speech. N-grams are often used to represent text data as a vector of n consecutive words.

	aback	aback bowed	aback bowed understand	aback case	aback case seem	aback made	aback made best	aback minute	aback minute know	aback seriously	...	zoöphagous	zoöphagous patient	zoöphagous patient report	zumpt	æsthetics	æstheti schil
0	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	0	
2	0	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	0	
4	0	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	0	
996	0	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	0	
998	0	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	0	

1000 rows × 122073 columns

Fig15: The transformation of N-gram (n=1:3)

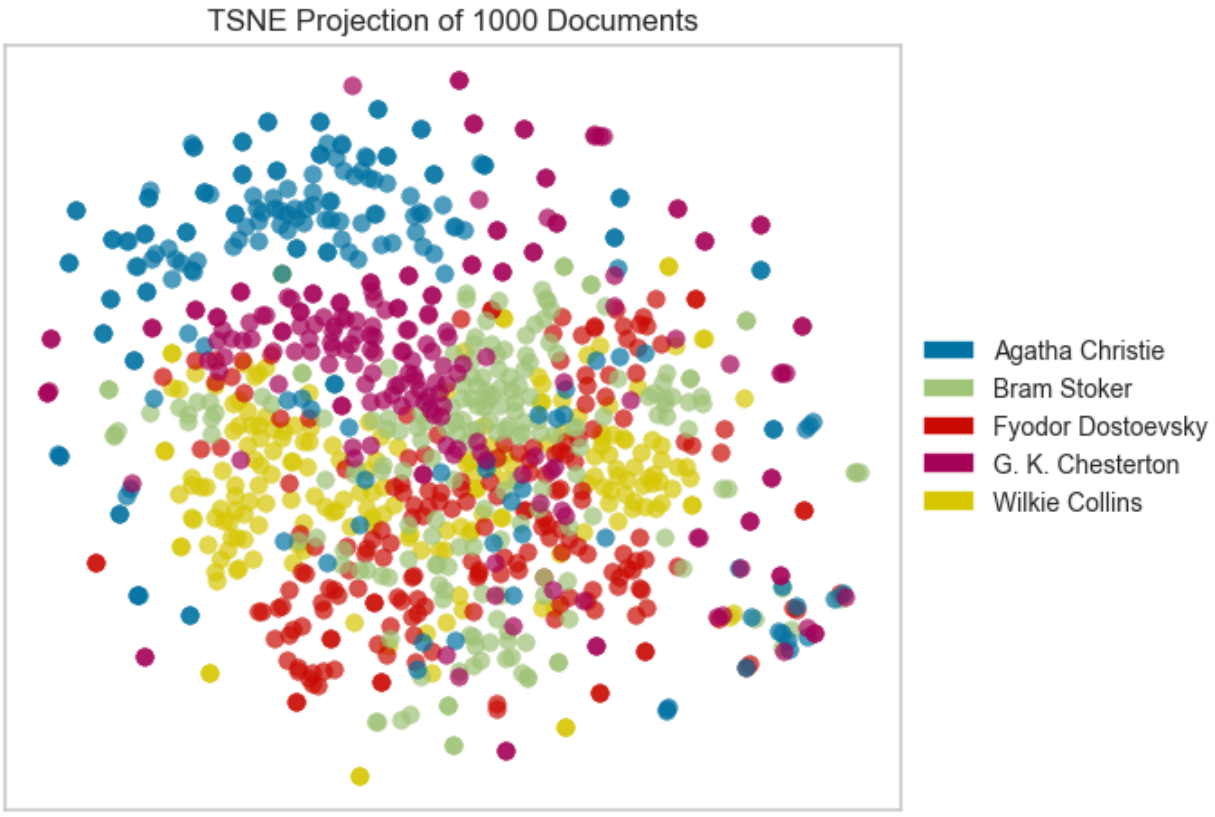


Fig16: TheTSNE of N-gram

2.4 Modeling

We train different models (SVM, Random Forest, Naïve Bayes, k-Nearest Neighbor, XG-Boost, SGD) to classify text into five different authors: Fyodor Dostoevsky, Wilkie Collins, G. K. Chesterton, Agatha Christie and Bram Stoker.

Results of testing accuracy of each model with each feature:

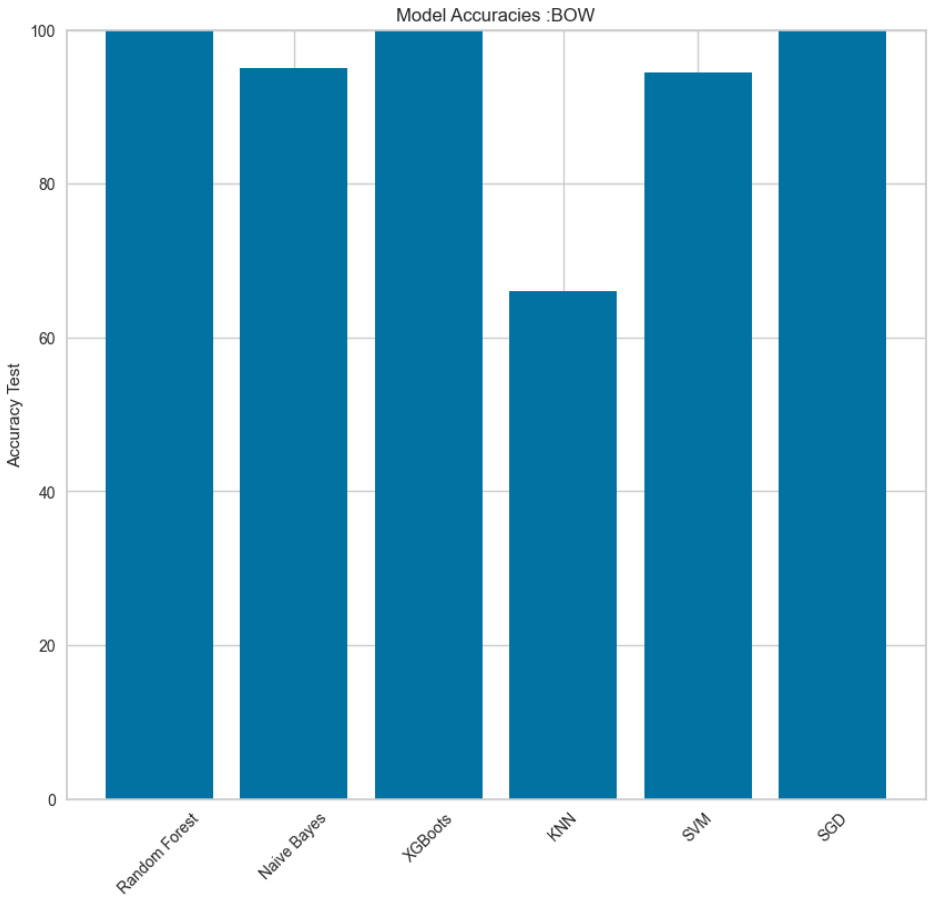


Fig17: Testing Accuracy Of Each Model With BOW

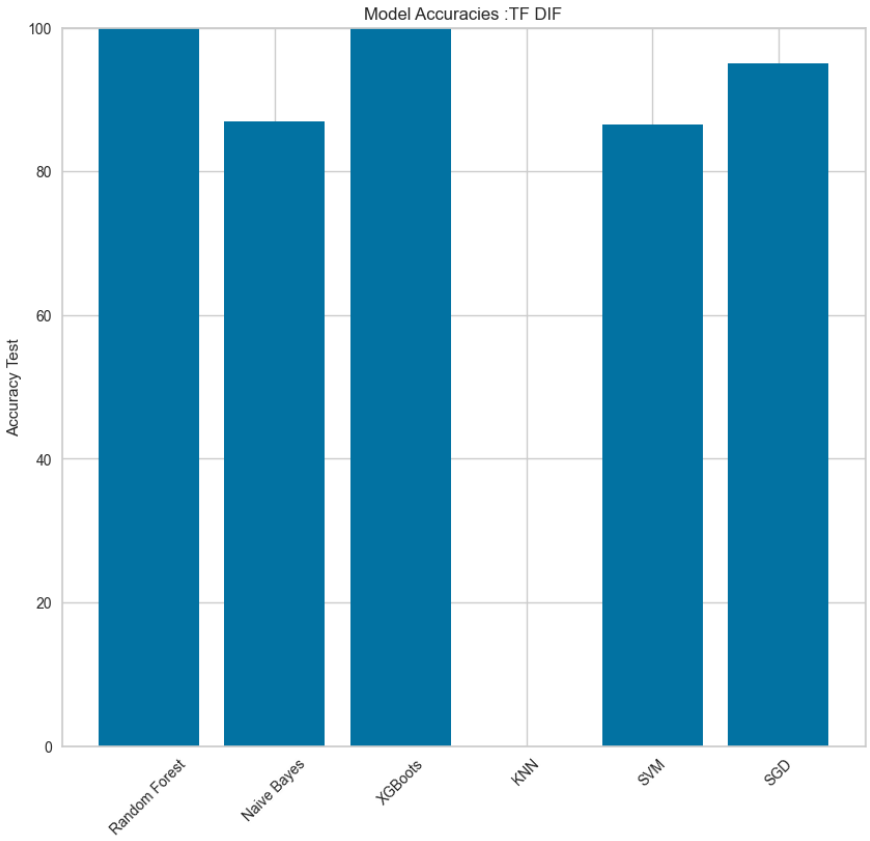


Fig18: Testing Accuracy Of Each Model With TF-IDF

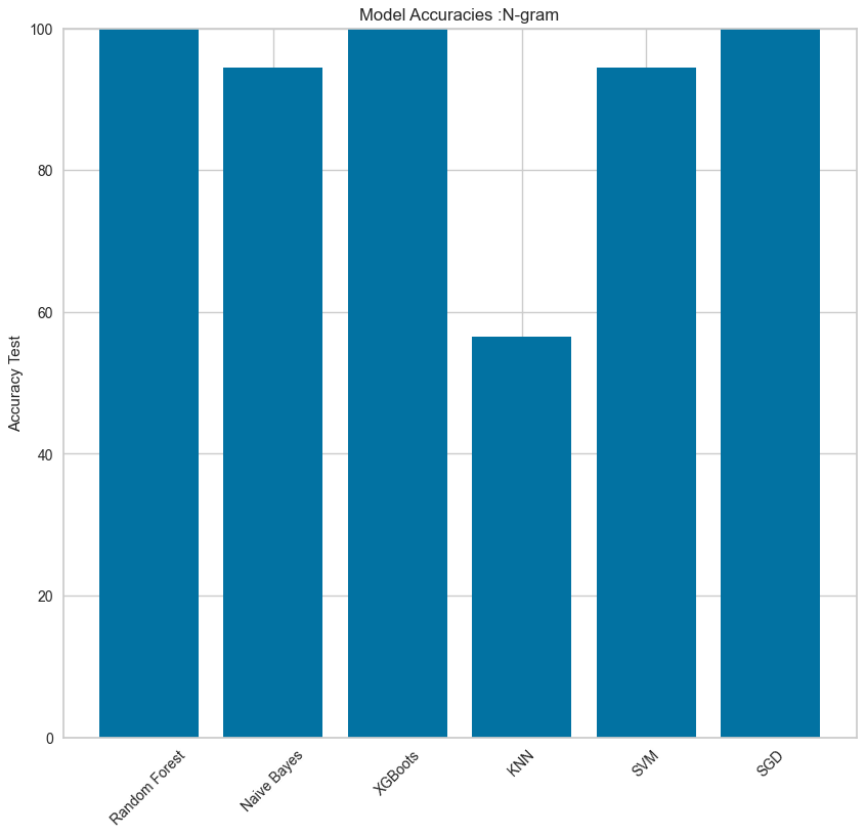
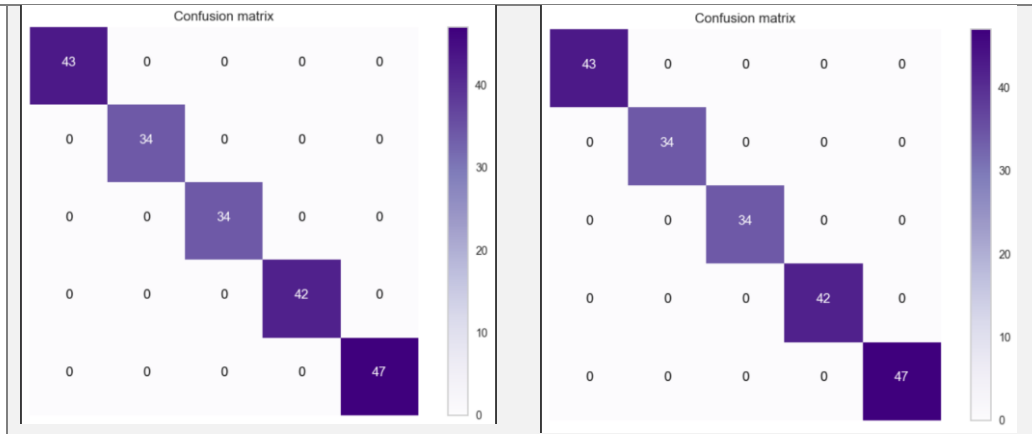
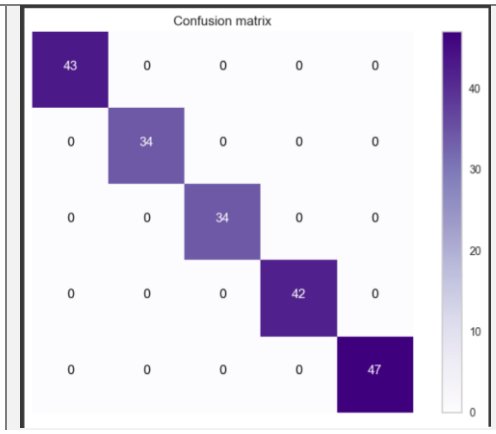
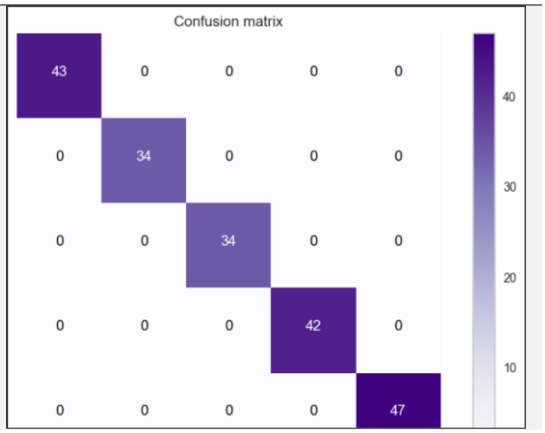


Fig19: Testing Accuracy Of Each Model With N-gram

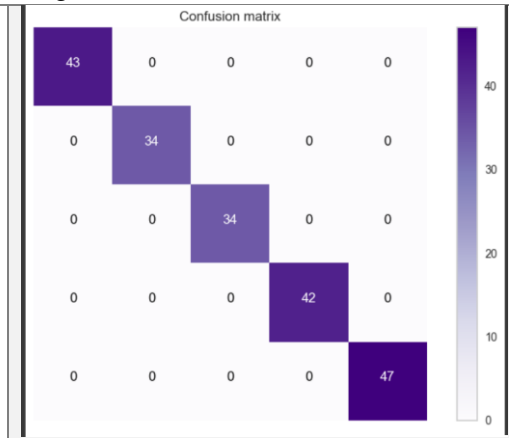
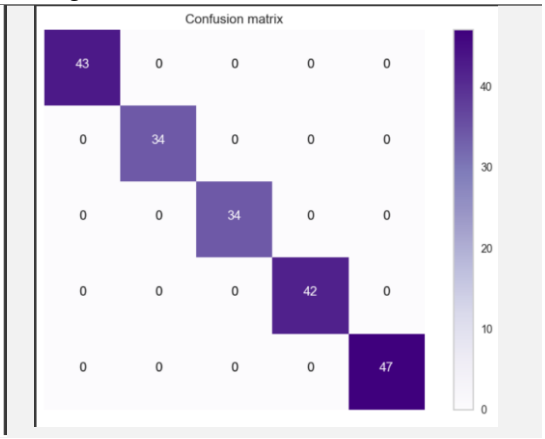
- **BOW with Random Forest Classifier and XG-Boost:**

	BOW with Random Forest	*BOW with XGboost*
Confusion matrix		

- **TF-IDF with Random Forest Classifier and XG-Boost:**

	TF-IDF with Random Forest	*TF-IDF with XGboost*
Confusion matrix		

- **N-gram with Random Forest Classifier and XG-Boost:**

	N-gram with Random Forest	*N-gram with XGboost*
Confusion matrix		

Our text classification problem involved training six different models, using ten-fold cross-validation to evaluate their performance, and identifying that Random Forest and XG-Boost achieved the highest accuracy while k-NN had the worst accuracy. By using ensemble learning algorithms, we were able to improve the accuracy of the predictions and achieve our goal of classifying text into different authors. The champion models are Random Forest with BOW or with n-gram and XG-Boost with accuracy 0.995.

2.5 Cross Validation

We evaluate the model using ten-fold cross-validation. We split the data-set into k number of subsets (known as folds) then we perform training on the all the subsets but leave one(k-1) subset for the evaluation of the trained model. In this method, we iterate k times with a different subset reserved for testing purpose each time.

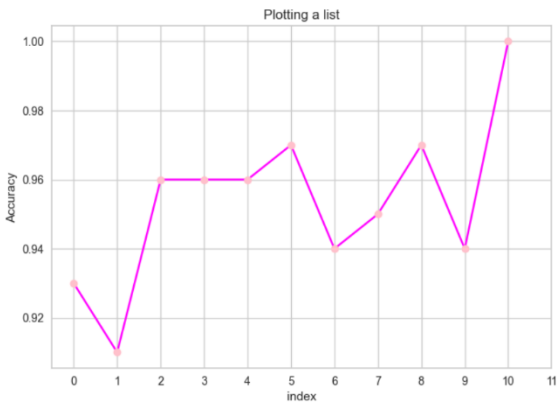


Fig20: accuracy of k-folds with Random Forest with BOW

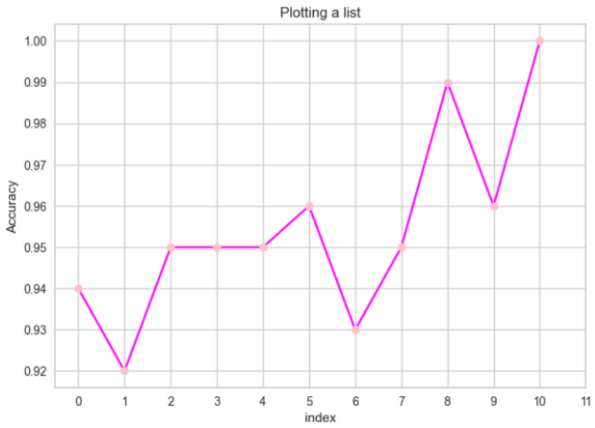


Fig21: accuracy of k-folds with Random Forest with TF-IDF

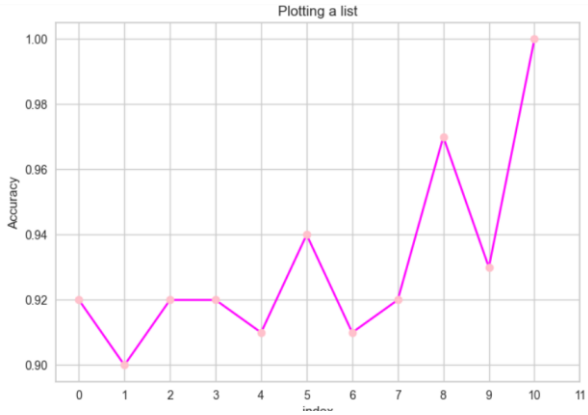


Fig22: accuracy of k-folds with Random Forest with N-gram

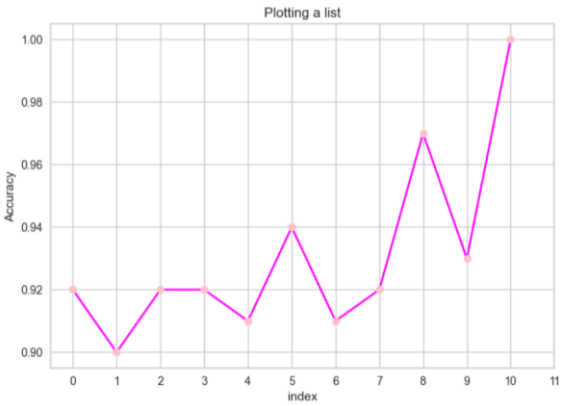


Fig23: accuracy of k-folds with XG-Boost with BOW

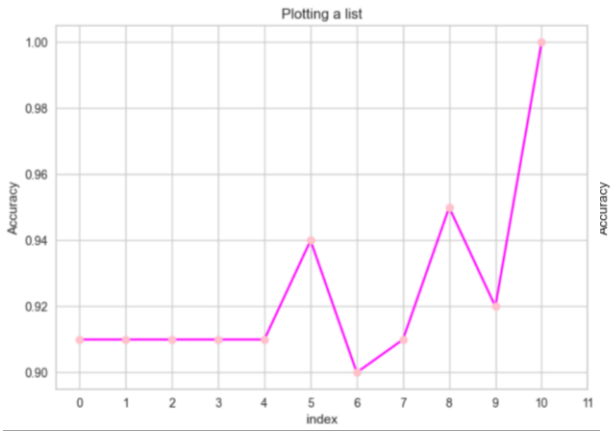


Fig24: accuracy of k-folds with XG-Boost with N-gram

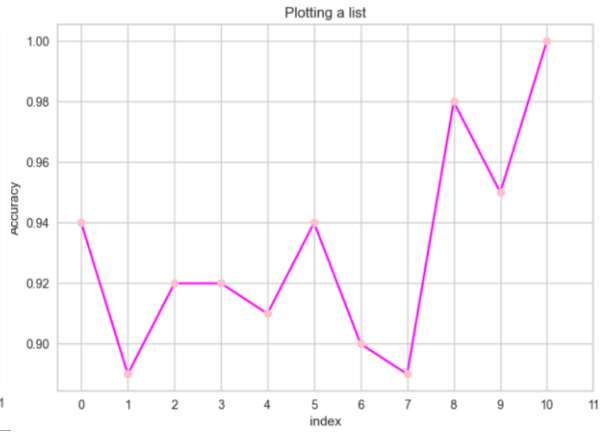


Fig25: accuracy of k-folds with XG-Boost with N-gram

2.6 Error Analysis

We also perform error analysis to identify the characteristics of the champion model. We changed the number of words from 100 words to 30, 50, and 70 words. We noticed that random forest with 50 words has accuracy of 92% with BOW and 91% with TF-IDF and N-gram and the speed of training was high, and when words were 70 the XG-Boost the accuracy was 92% with BOW and N-gram, and 94% with TF-IDF and the speed was lower than speed of Random Forest.

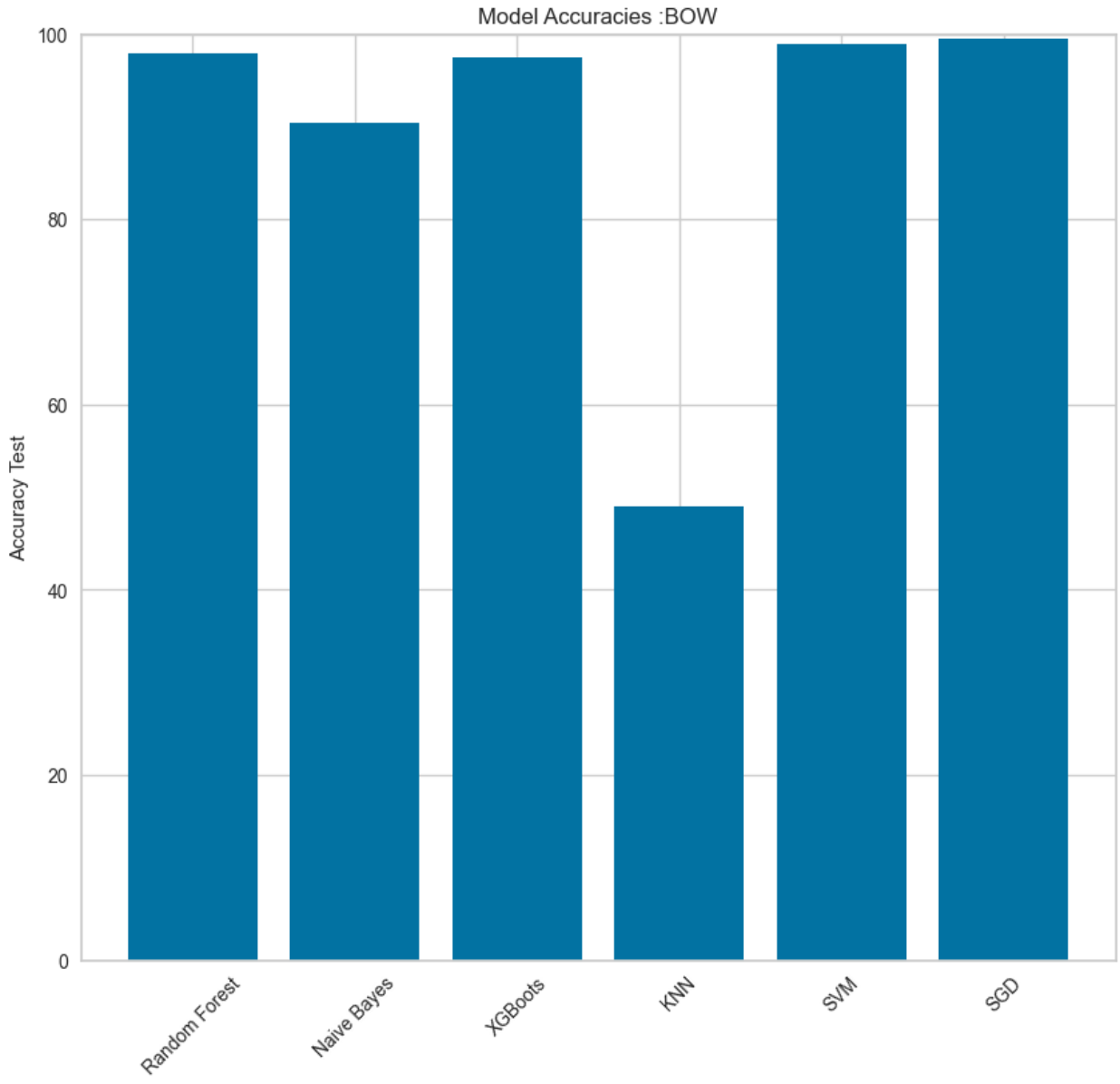


Fig26: Testing Accuracy Of Each Model With BOW when words=70

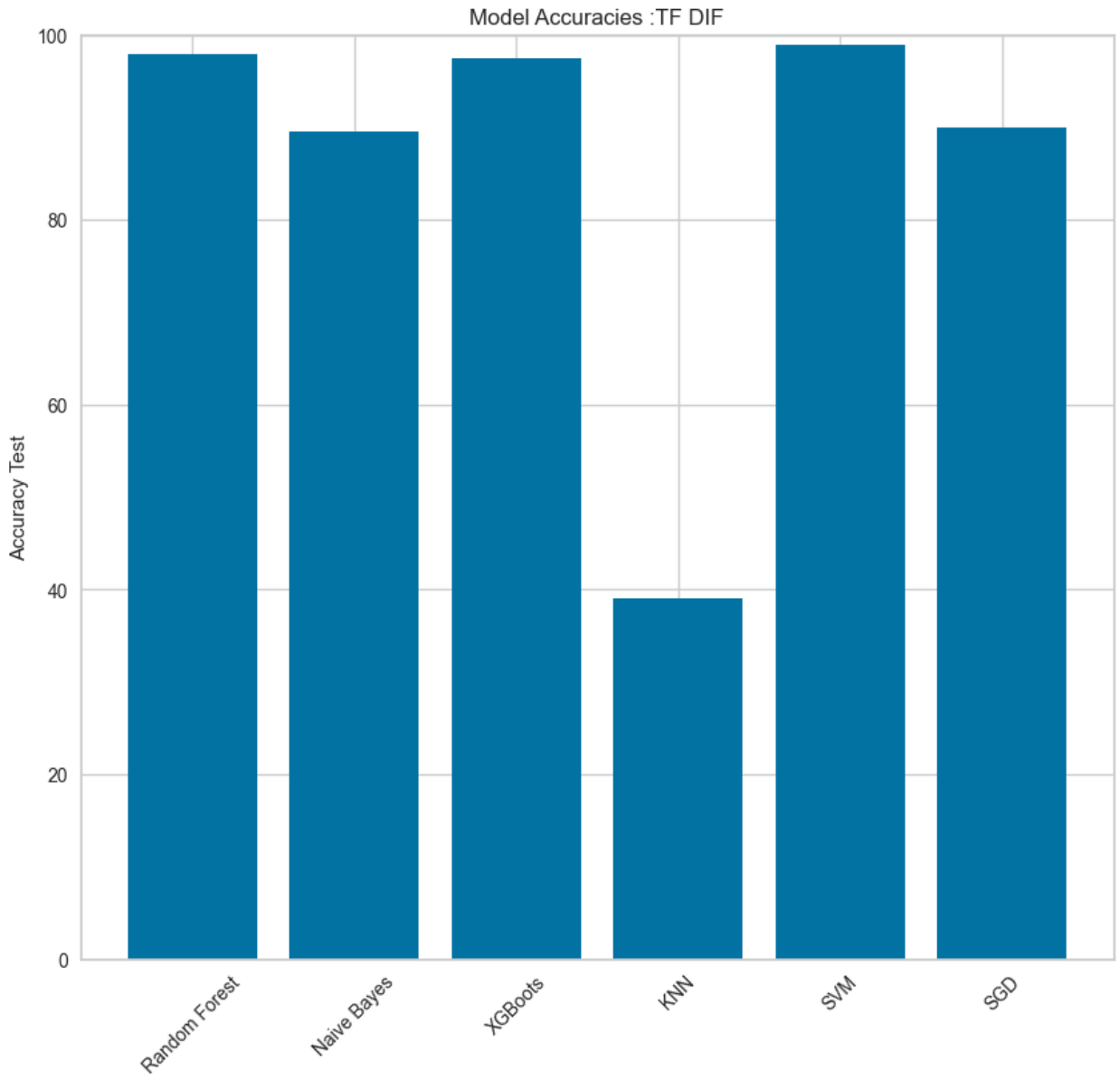


Fig27: Testing Accuracy Of Each Model With TF-IDF when words=70

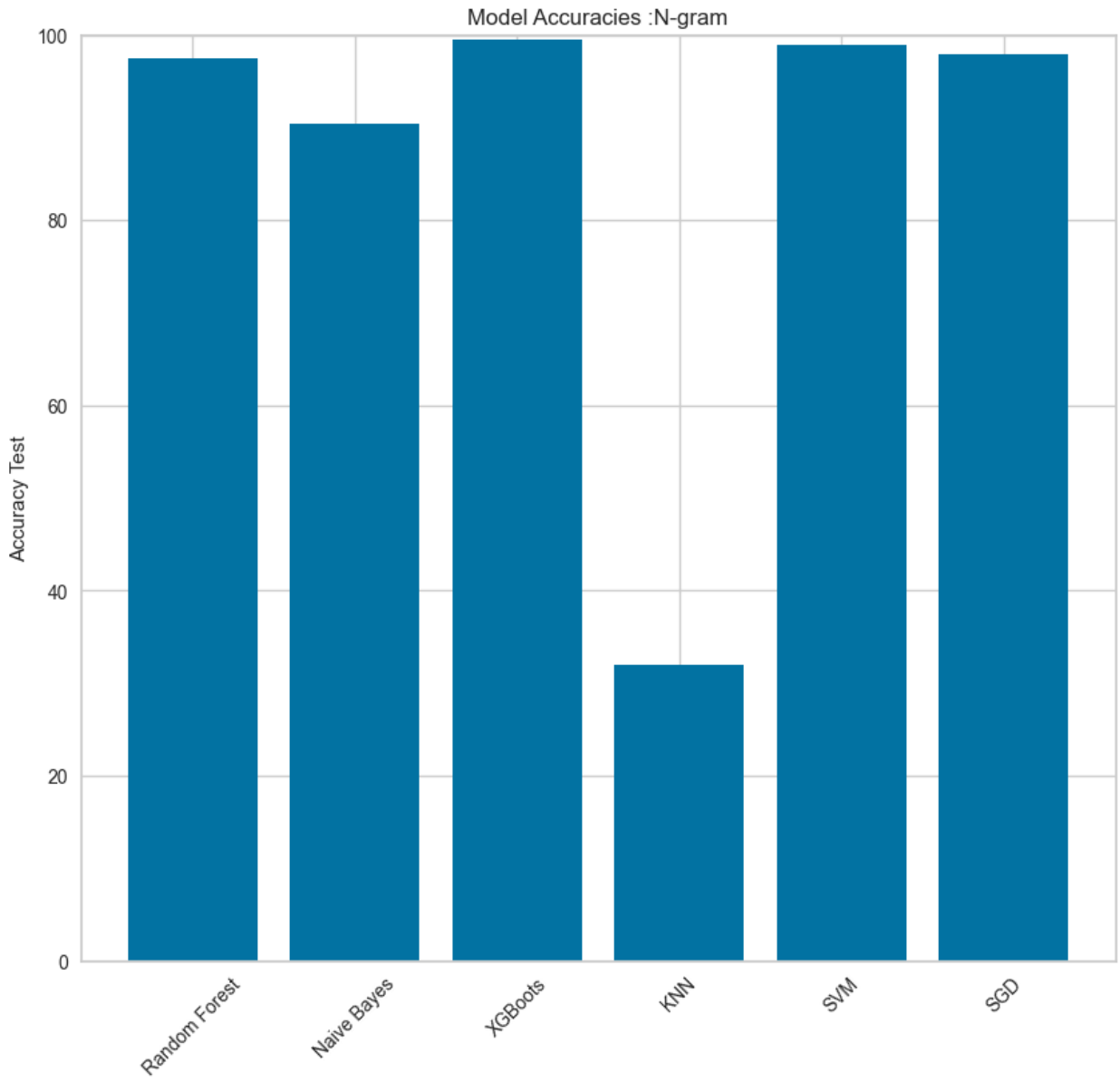


Fig28: Testing Accuracy Of Each Model With N-gram when words=70

2.7 Analysis of Bias

Before performing lemmatization, we found that all books have words: say, said and saying as common words. We performed lemmatization and all words return to base so all books have the same word and this led the model to struggle to distinguish between documents written by authors with similar words.

3. Conclusion

Our results suggest that ensemble learning algorithms like Random Forest and XGBoost are effective for text classification problems, while simpler algorithms like k-Nearest Neighbor may not be as effective. The use of ten-fold cross-validation helped us to obtain more reliable estimates of the models' performance and avoid overfitting to the training data. Adding more features did not significantly improve the model's performance, and reducing the number of words per document than 50 words made it harder for the model to predict but did not reduce the bias and variability.

4. References

1) https://scikitlearn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html
2) 1.4. Support Vector Machines — scikit-learn 1.2.2 documentation
3) 1.9. Naive Bayes — scikit-learn 1.2.2 documentation
4) 1.5. Stochastic Gradient Descent — scikit-learn 1.2.2 documentation
5) https://towardsdatascience.com/build-a-bert-sci-kit-transformer-59d60ddd54a5