uOttawa

Faculté de génie
Faculty of Engineering

DTI5125: Data Science Applications

Assignment group 1 (Text Classification)

Group 8

Table of Contents

Table of Figures:

Loading books:

Using NTLK, five books were chosen from fiction category.

```
chosen_books = ['austen-emma.txt', 'melville-moby_dick.txt', 'bryant-stories.txt', 'edgeworth-parents.txt', 'chesterton-ball.txt']
```

Preparing data:

A specific number of partitions will be chosen from each book. Each partition contains a specific number of words. The text that will be taken from the book will be cleaned by removing titles, headlines, special characters, digits. It's optional to remove stop words and stemming the words. Chosen partitions as a string and the list of words in these partitions will be put along with it's label (the book name) in the data frame as shown in the following figure:

	text	list_of_words	book_name
833	cried macian violently oh you have long words ...	[cried, macian, violently, oh, you, have, long...	chesterton-ball
195	subject or animation or of courage and opportu...	[subject, or, animation, or, of, courage, and,...	austen-emma
412	shoemaker bought leather for four pairs with t...	[shoemaker, bought, leather, for, four, pairs,...	bryant-stories
404	and he said to david am i a dog that thou come...	[and, he, said, to, david, am, i, a, dog, that...	bryant-stories
34	know that he ever had any such plan nay i had ...	[know, that, he, ever, had, any, such, plan, n...	austen-emma
719	as an affair of chance the consequence was tha...	[as, an, affair, of, chance, the, consequence,...	edgeworth-parents
393	this custom has now become obsolete turn we th...	[this, custom, has, now, become, obsolete, tur...	melville-moby_dick
462	baser than the beasts you hunt you are furious...	[baser, than, the, beasts, you, hunt, you, are...	bryant-stories
705	convinced us whichever you pleased said townse...	[convinced, us, whichever, you, pleased, said,...	edgeworth-parents
118	agreeing till he is really here but i am very ...	[agreeing, till, he, is, really, here, but, i,...	austen-emma

Figure 1 A sample of 10 partitions from the data

Text transformation, Feature Engineering:

To build a model that can classify text, we need to perform a feature extraction process in which we prepare the input to be in the shape that the model can understand. In this assignment, we used three main techniques of feature extraction: BOW, TF-IDF, and nGrams.

We generalized the code so we could use each method many times with different algorithms without implementing the technique again.

Models:

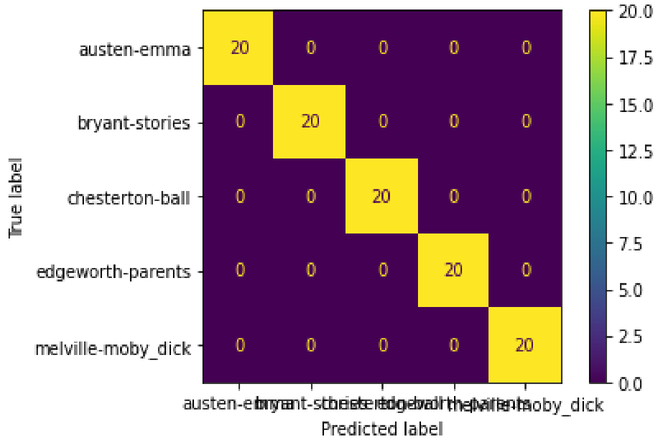
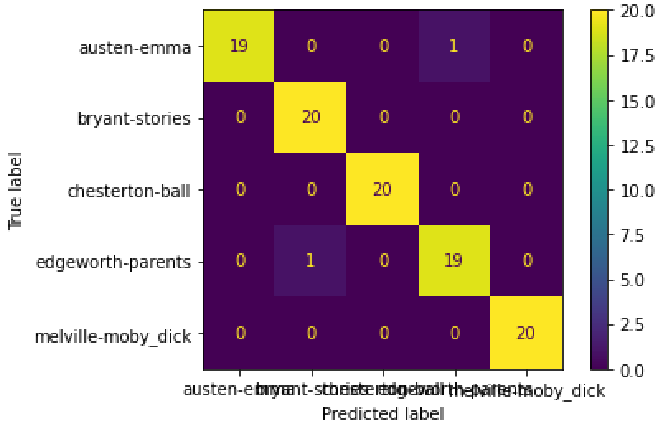
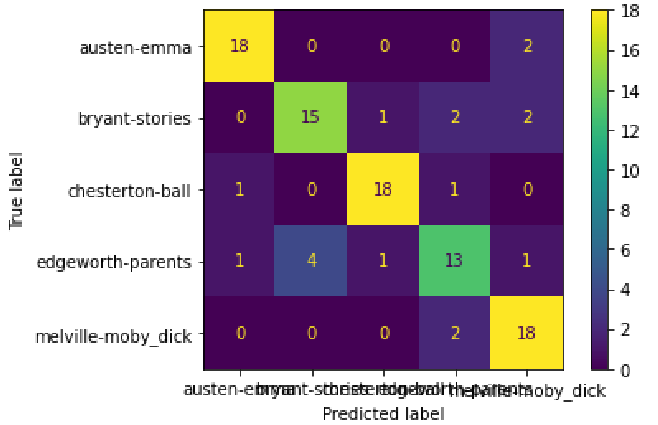
in this step we train multi models (logistic regression, SVM, Decision Tree, RandomForest Classifier, GradientBoosting, k-Nearest Neighbour and naive bayes) each model train with each features engineering techniques (Bog ,TF-IDF and 2_gram) and track validation accuracy and variance for each combination between models and feature engineering techniques

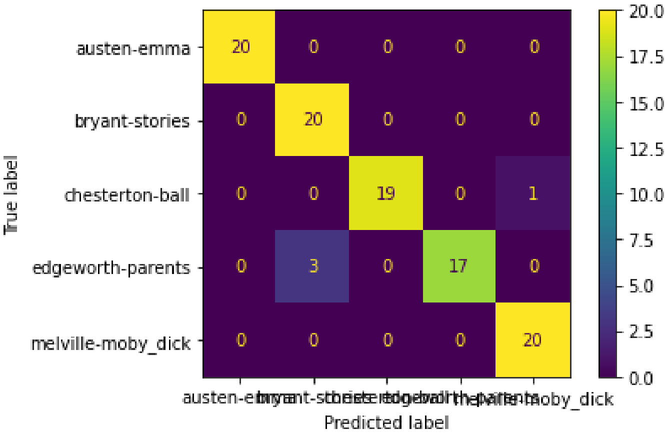
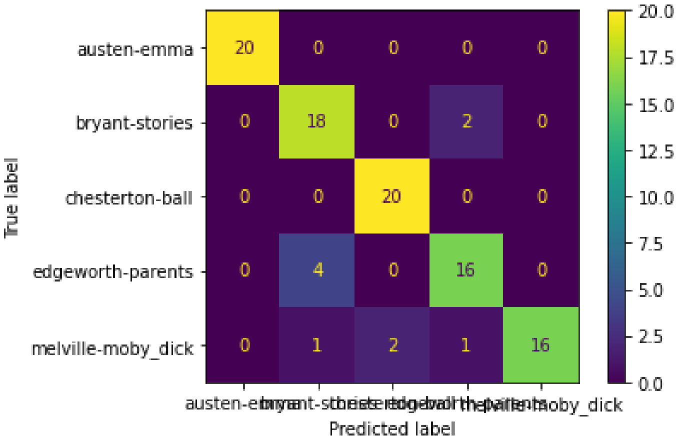
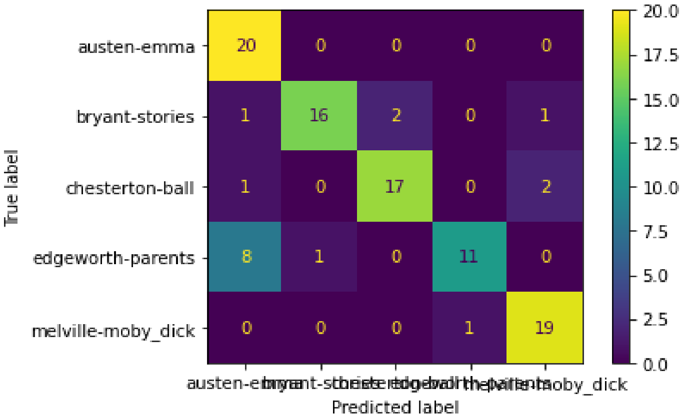
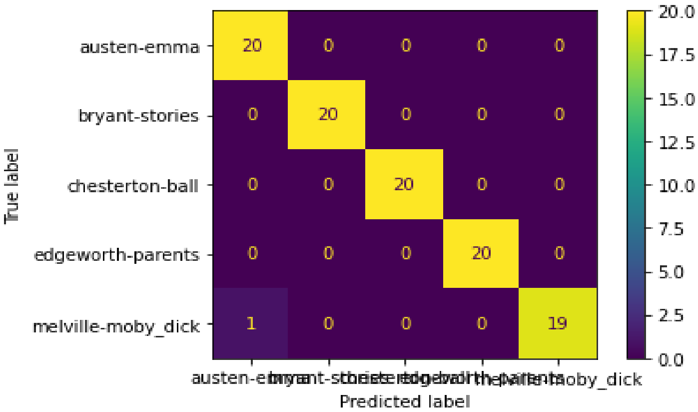
Evaluation:

Model validation technique is used to check if the train that is applied to all models and their all feature engineering combination has a good performance or not, by applying a cross-validation technique and checking the mean accuracy of the cross-validation models and their variance and select the champion model based on its cross-validation performance.

Results:

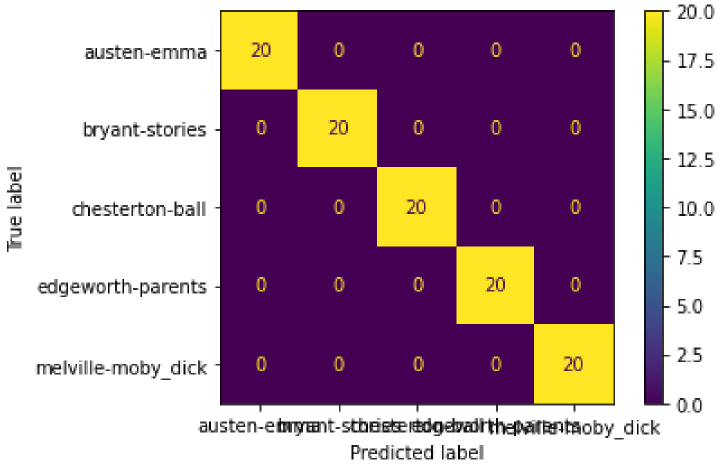
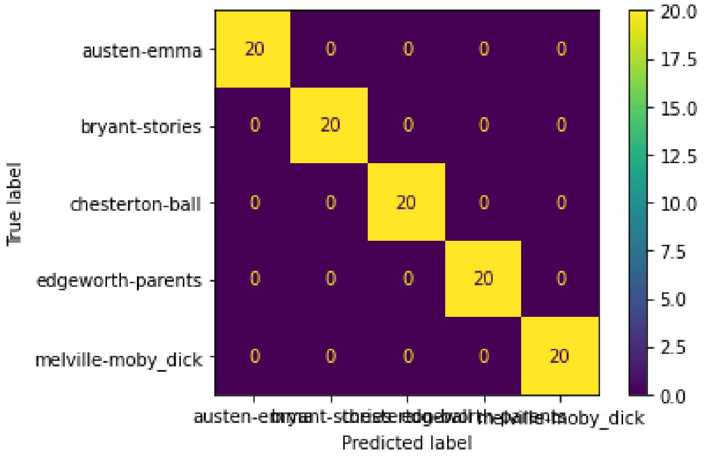
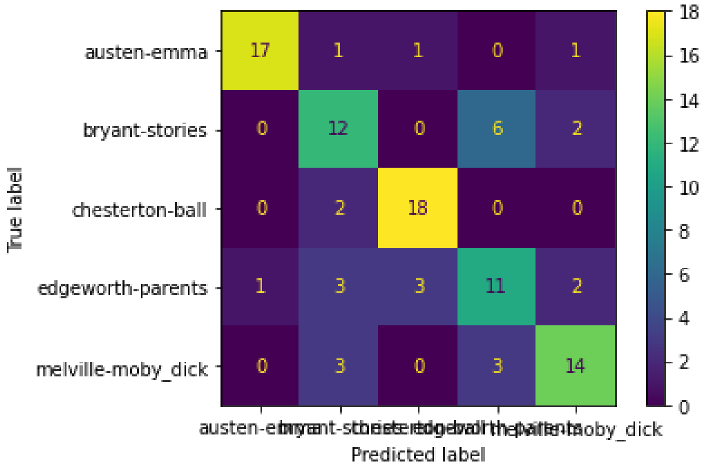
BOW with splitting data into train and validate (not using cross validation)

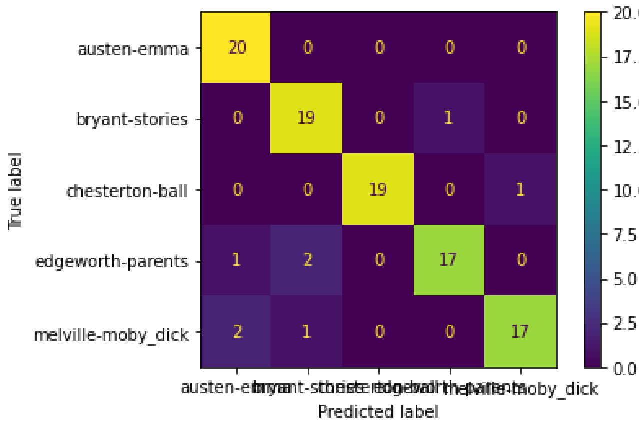
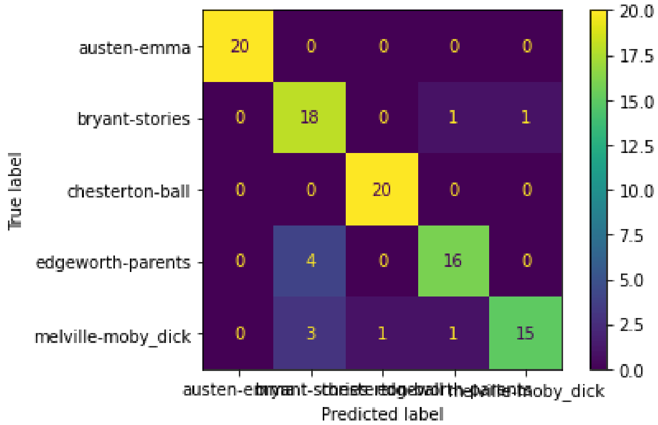
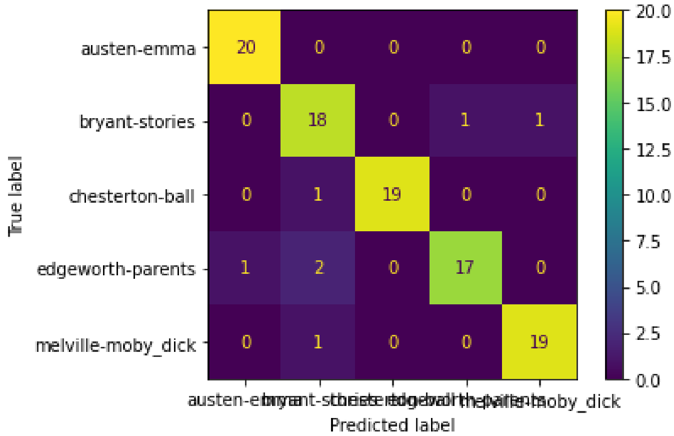
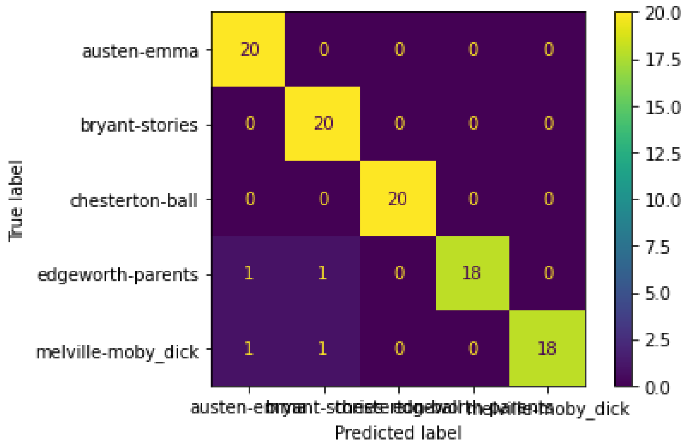
	Confusion Matrix on validation data	Train Accuracy	Validation Accuracy
Logistic regression		100%	100%
SVM		100%	98%
Decision Tree		100%	82% Overfitting

<table>
<tr><td>RandomForest</td><td>
</td><td>100%</td><td>96%</td></tr>
<tr><td>GradientBoosting</td><td>
</td><td>100%</td><td>97%</td></tr>
<tr><td>KNN</td><td>
</td><td>91.875%</td><td>83%
Overfitting</td></tr>
<tr><td>Naïve bayes</td><td>
</td><td>100%</td><td>99%</td></tr>
</table>

From the above results, it's concluded that using BOW features with all models gave good results except decision tree and KNN models as they experienced an overfitting as the difference between the train accuracy and the validation accuracy isn't low.

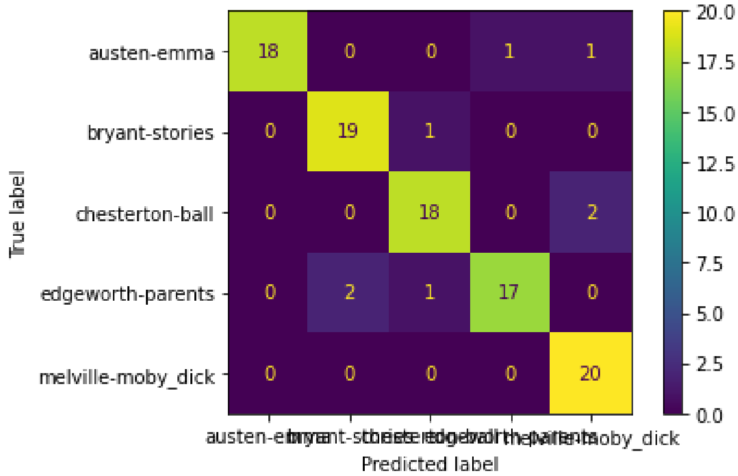
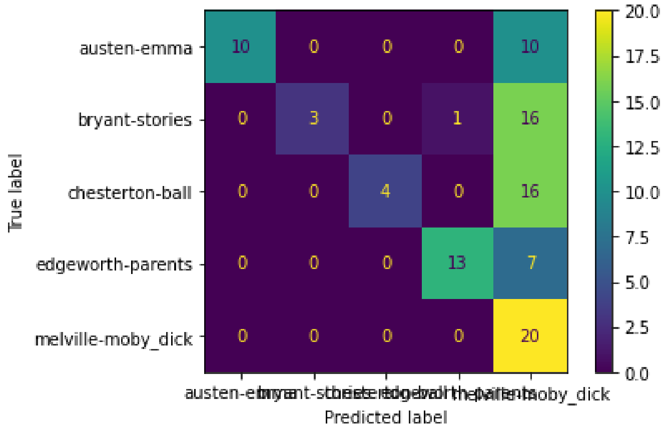
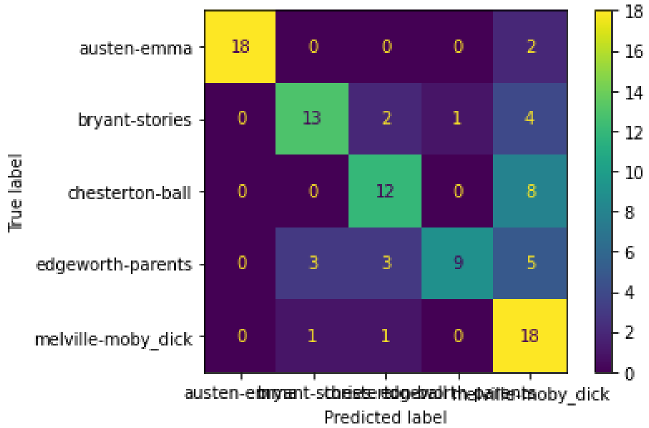
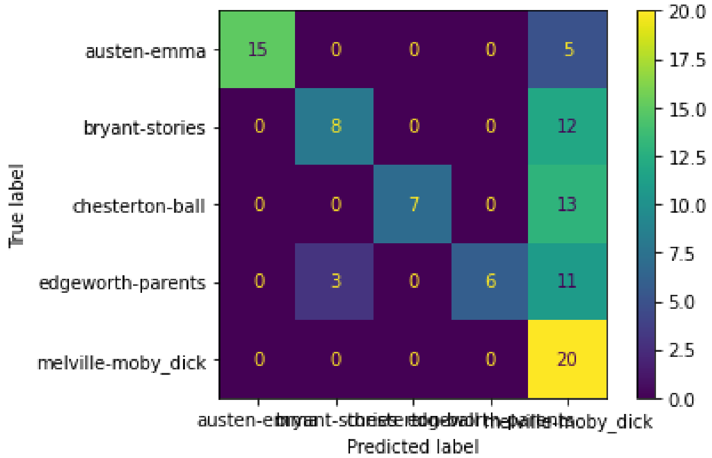
TFIDF with splitting data into train and validate (not using cross validation)

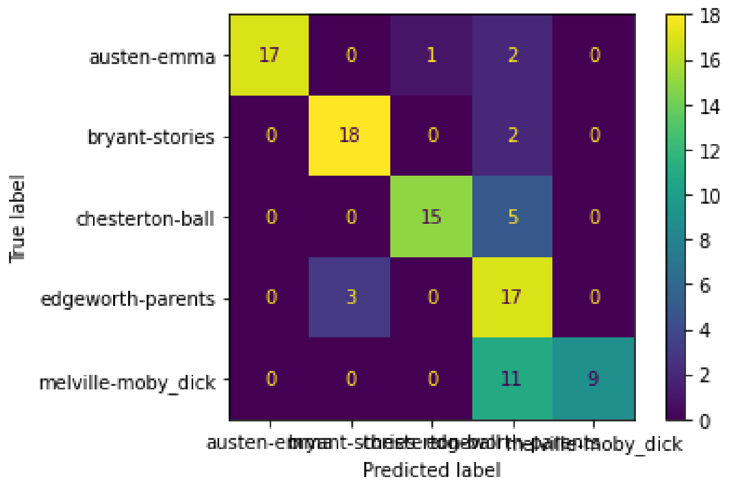
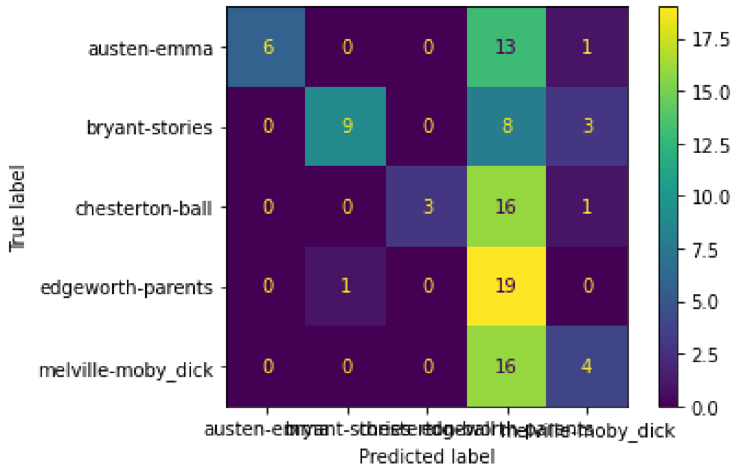
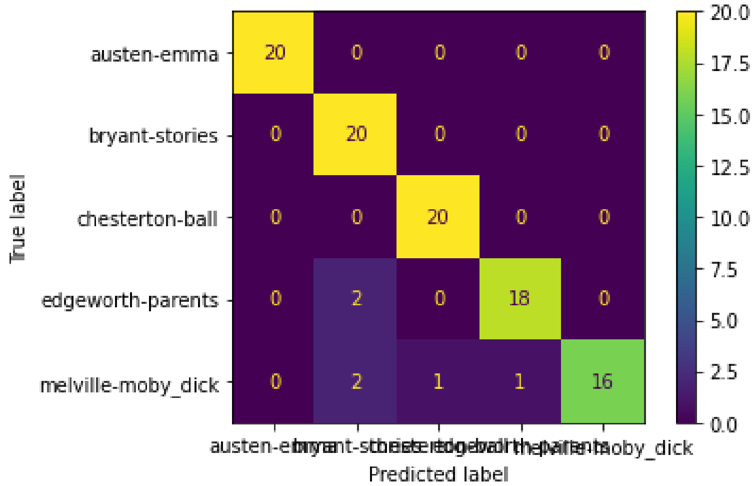
	Confusion Matrix on validation data	Train Accuracy	Validation Accuracy
Logistic regression		100%	100%
SVM		100%	100%
Decision Tree		100%	72% Overfitting

RandomForest		100%	92% Overfitting
GradientBoosting		100%	89% Overfitting
KNN		96.625%	93%
Naïve bayes		99.875%	96%

From the above results, it's concluded that using TFIDF features with Naïve bayes, KNN, SVM, and logistic regression models gave good results while decision tree, random forest, and gradient boosting models experienced an overfitting.

Bigram with splitting data into train and validate (not using cross validation)

<table>
<tr><th></th><th>Confusion Matrix on validation data</th><th>Train Accuracy</th><th>Validation Accuracy</th></tr>
<tr><td>Logistic regression</td><td>
</td><td>100%</td><td>92%
Overfitting</td></tr>
<tr><td>SVM</td><td>
</td><td>100%</td><td>50%
Overfitting</td></tr>
<tr><td>Decision Tree</td><td>
</td><td>100%</td><td>70%
Overfitting</td></tr>
<tr><td>RandomForest</td><td>
</td><td>100%</td><td>56%
Overfitting</td></tr>
</table>

GradientBoosting		100%	76% Overfitting
KNN		71.5%	41% Underfitting
Naïve bayes		100 %	94%

From the above figures it's concluded that all models that used bigram features overfit except Naïve bayes model that gave nearly acceptable performance.

Comparing train time of all previous models

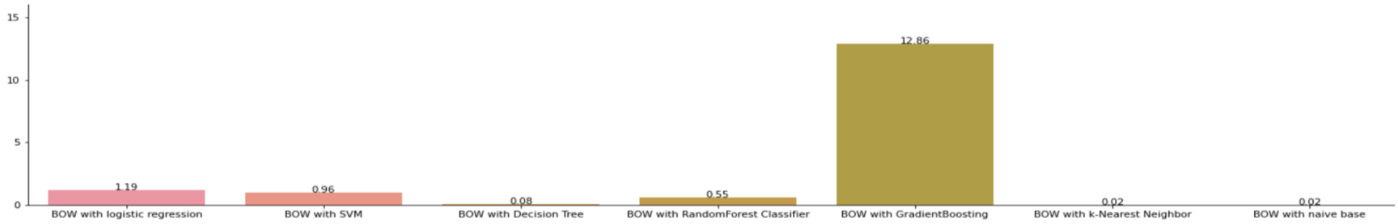


Figure 2 comparing between train time of all models using BOW features

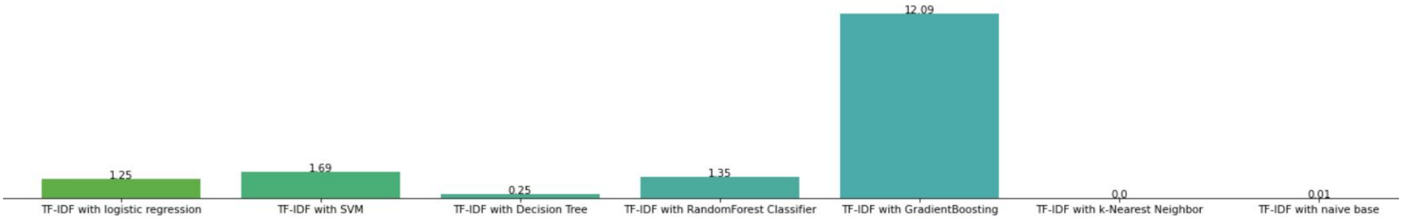


Figure 3 comparing between train time of all models using TFIDF features

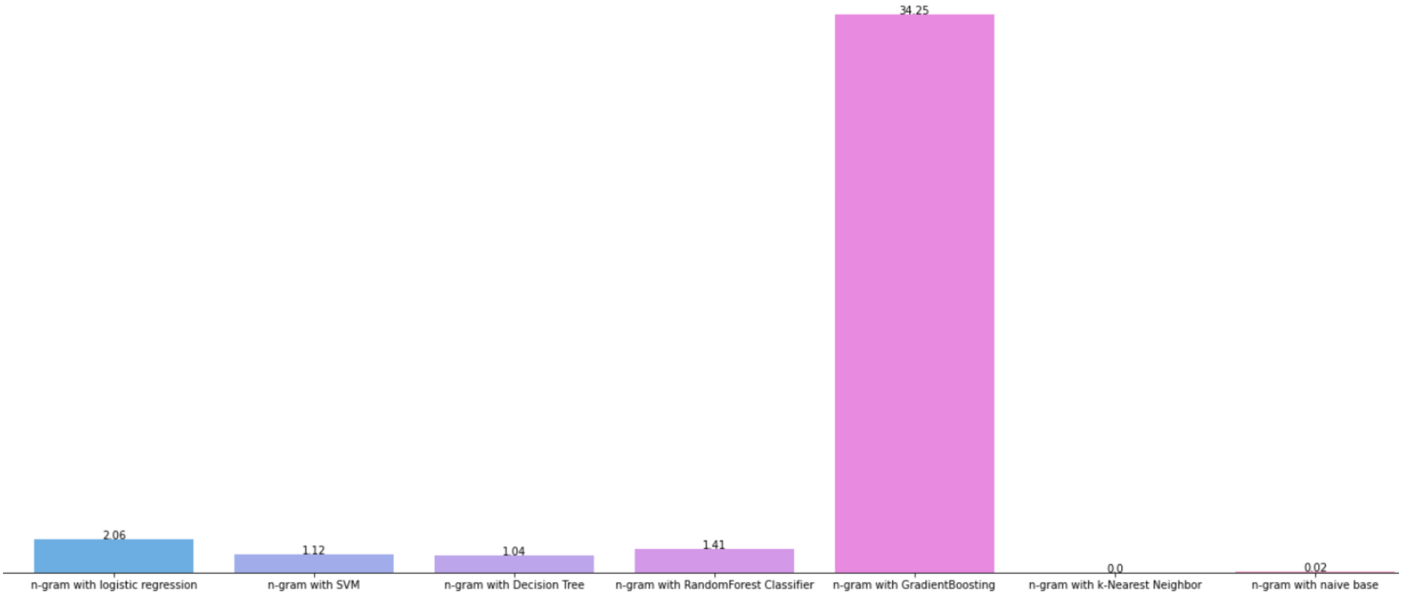
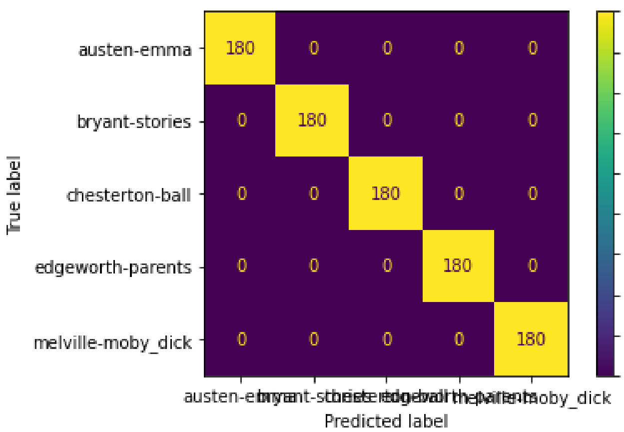
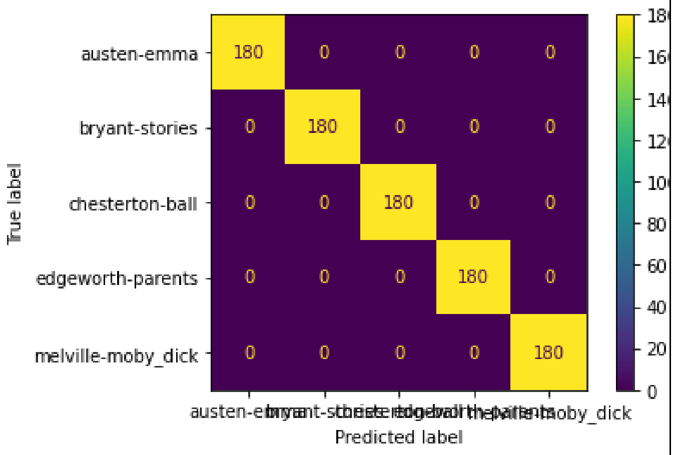
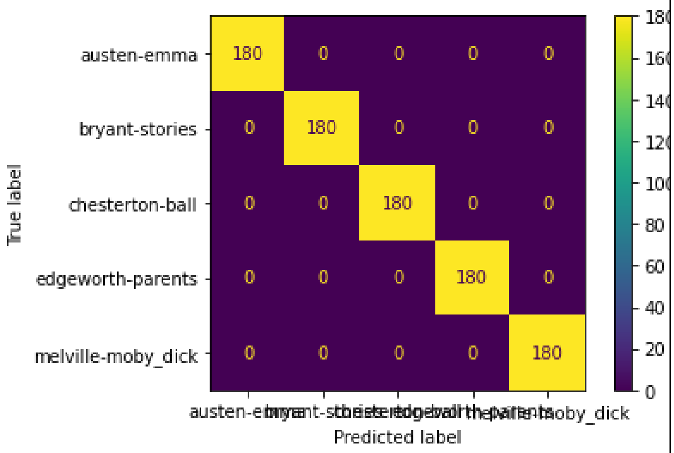
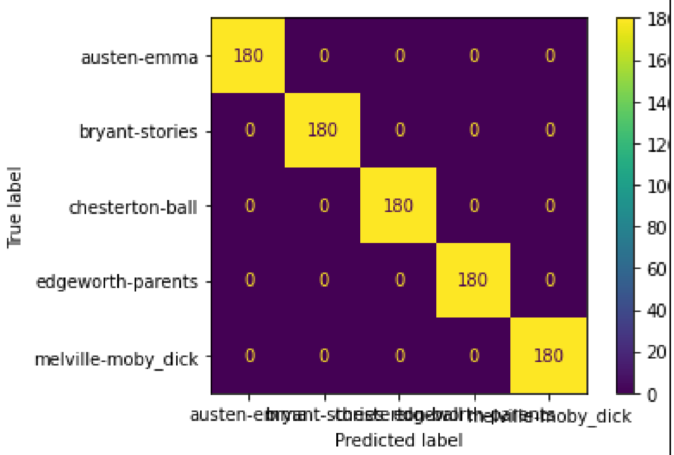
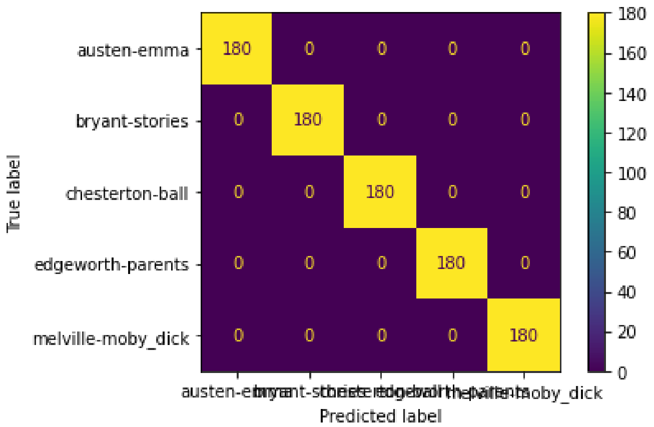


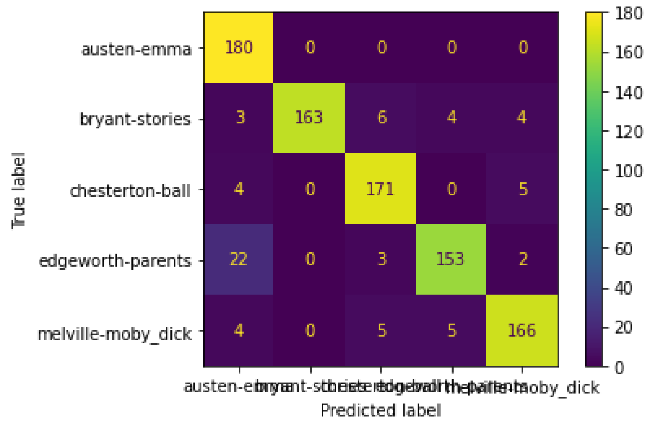
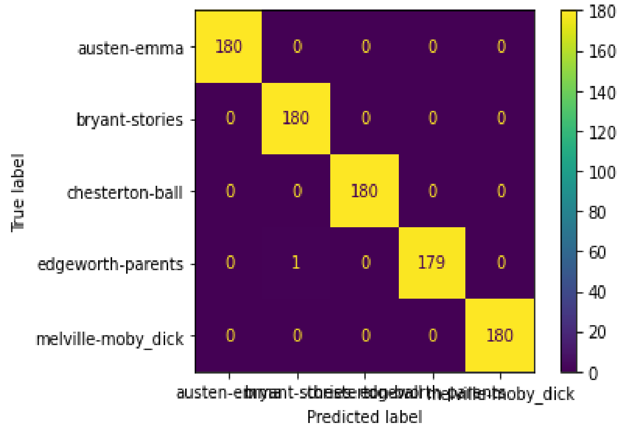
Figure 4 comparing between train time of all models using Bigram features

From the above figures, it’s concluded that training the gradient boosting took the longest time to train regardless the type of features that were used.

BOW with 10 folds cross validation

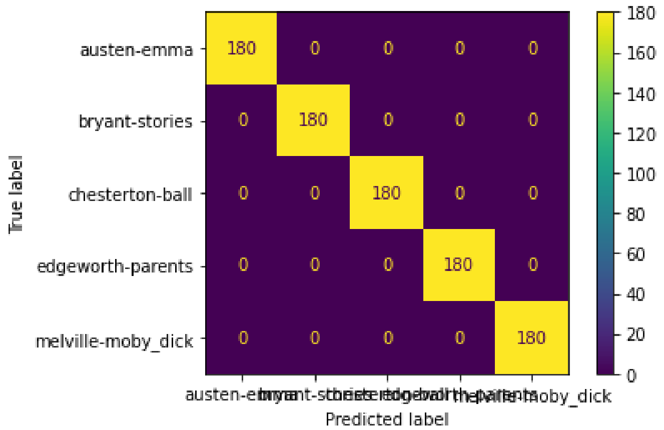
	Confusion Matrix on training data	Train Accuracy	Mean of validation Accuracies	Variance
Logistic regression		100%	97.22%	1.94

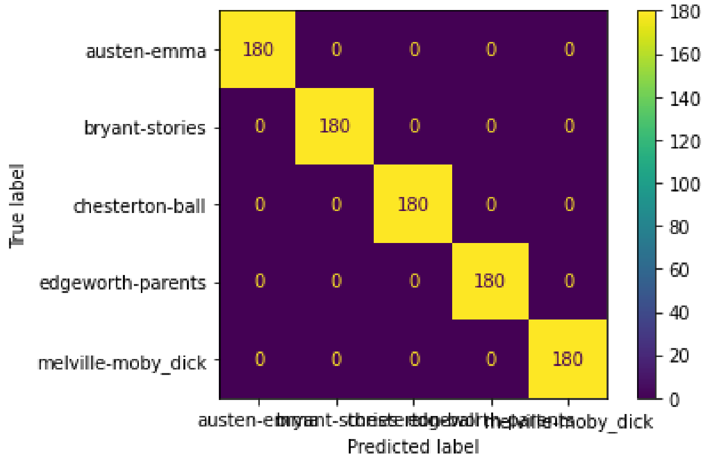
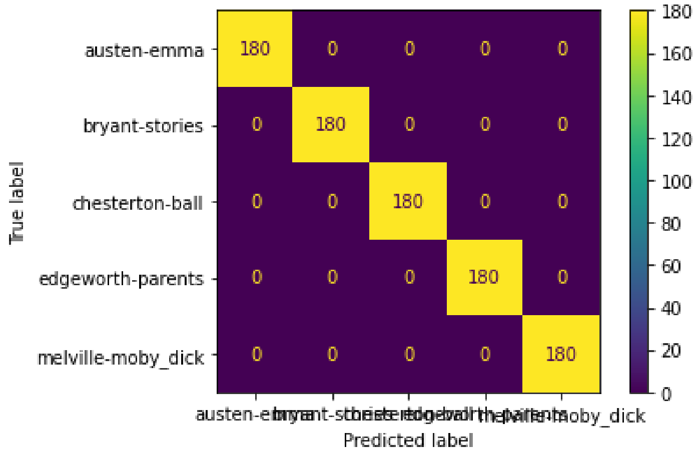
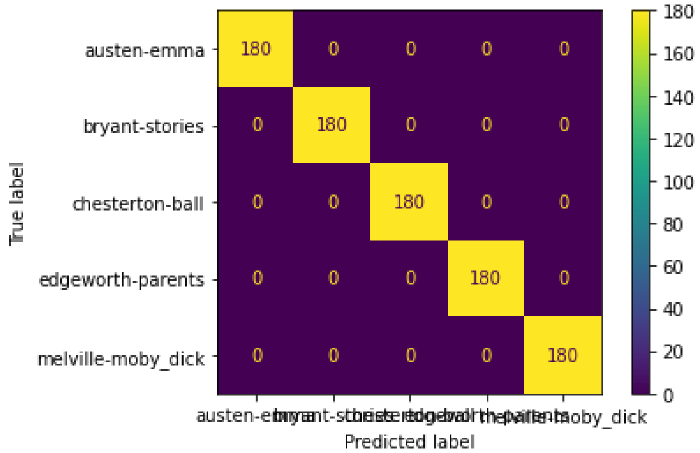
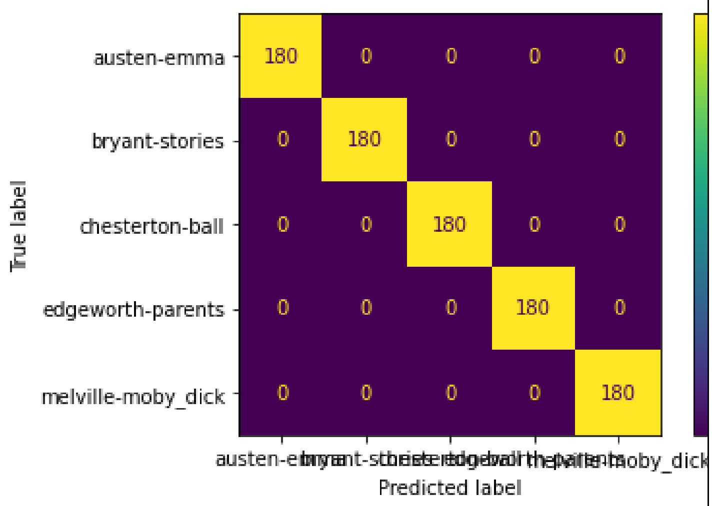
SVM		100%	95.88%	1.65
Decision Tree		100%	80.11% Overfitting	3.76
RandomForest		100%	94.22%	1.63
GradientBoostin g		100%	93% Overfitting	4.15

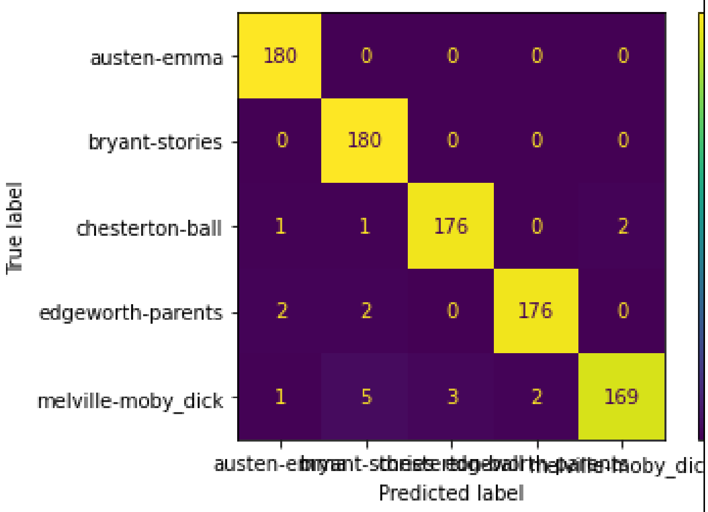
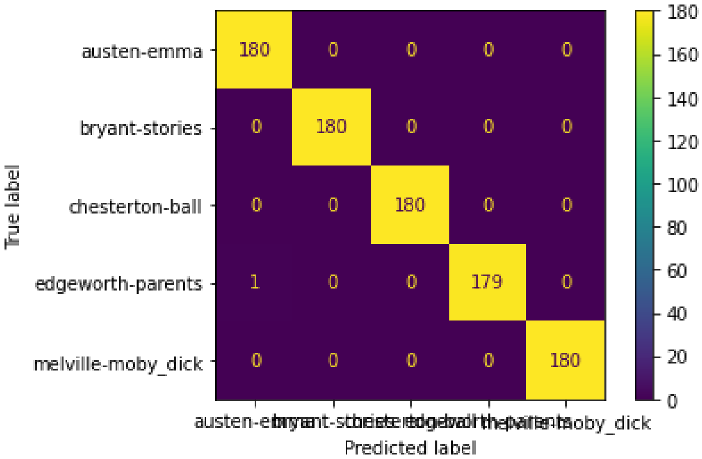
KNN		92.55%	84.88% Overfitting	4.13
Naïve bayes		99.88%	98.55%	1.49

From the above results, it's concluded that using BOW features with Naïve bayes, SVM, random forest, and logistic regression models gave good results while decision tree, random forest, KNN, and gradient boosting models experienced an overfitting as the difference between the training accuracy and the mean of validation accuracies isn't low also the variance of the accuracies is considered high when it's compared with the other good models.

TFIDF with 10 fold cross validation

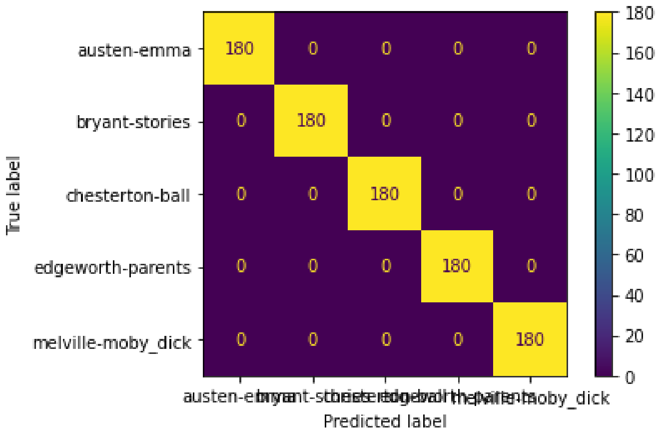
	Confusion Matrix on training data	Train Accuracy	Mean of validation Accuracies	Variance
Logistic regression		100%	98.22%	1.23

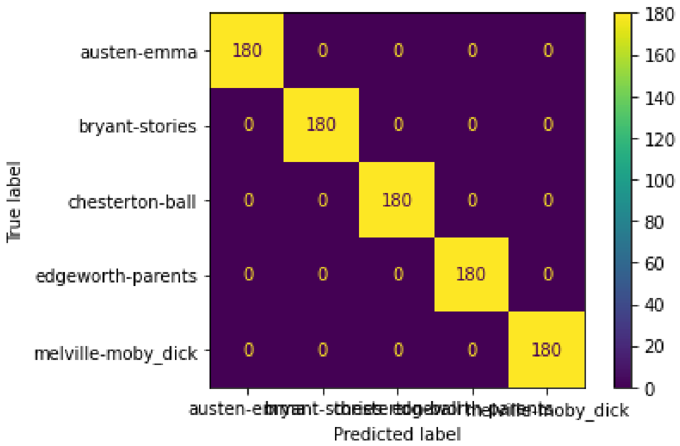
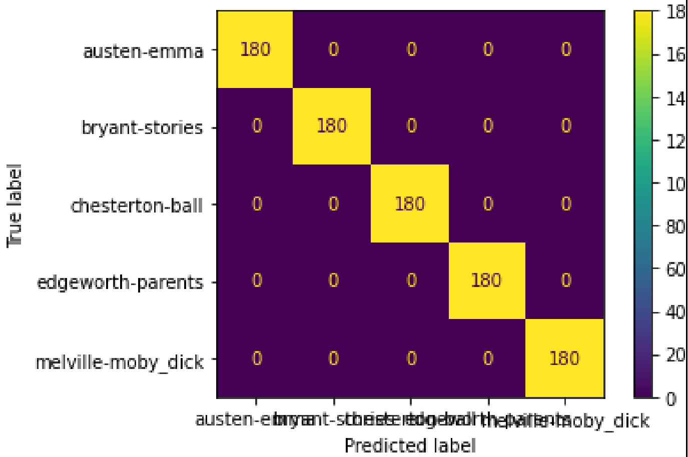
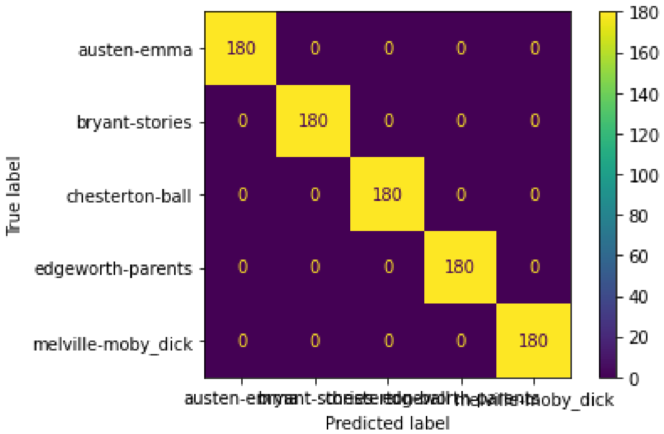
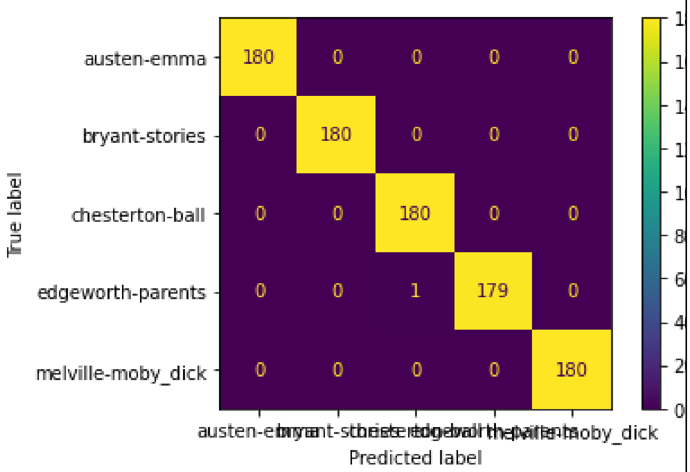
<table>
<tr><td>SVM</td><td>
</td><td>100%</td><td>98.55%</td><td>1.49</td></tr>
<tr><td>Decision Tree</td><td>
</td><td>100%</td><td>77.11%
Overfitting</td><td>2.59</td></tr>
<tr><td>RandomForest</td><td>
</td><td>100%</td><td>93.22%
Overfitting</td><td>1.89</td></tr>
<tr><td>GradientBoostin
g</td><td>
</td><td>100%</td><td>92.67%
Overfitting</td><td>3.69</td></tr>
</table>

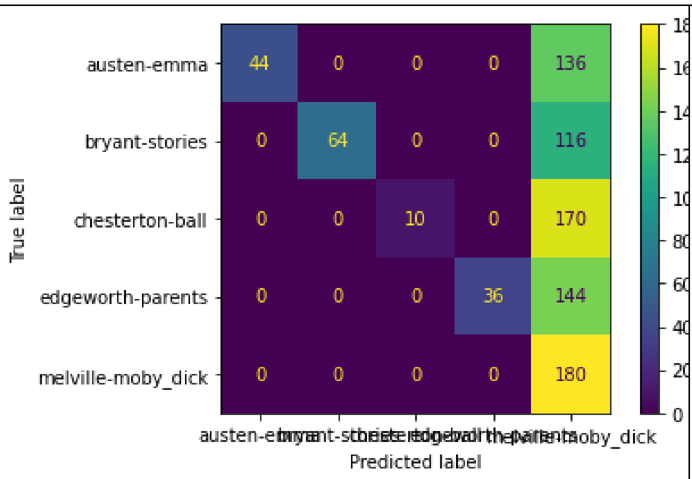
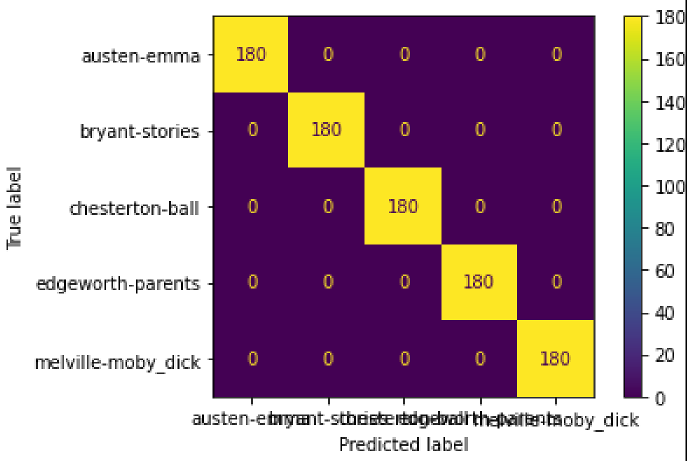
KNN		97.88%	94.44% Overfitting	2.98
Naïve bayes		99.89%	96.77%	2.65

From the above results, it's concluded that using TFIDF features with Naïve bayes, SVM, random forest, and logistic regression models gave good results while decision tree, random forest, KNN, and gradient boosting models experienced an overfitting as the difference between the training accuracy and the mean of validation accuracies isn't low also the variance of the accuracies is considered high when it's compared with the other good models.

Bigram with 10 fold cross validation

	Confusion Matrix on training data	Train Accuracy	Mean of validation Accuracies	Variance
Logistic regression		100%	88% Overfitting	5.11

SVM		100%	67.77% Overfitting	5.85
Decision Tree		100%	63.77% Overfitting	4.56
RandomForest		100%	56.22% Overfitting	3.85
GradientBoostin g		99.89%	69.67% Overfitting	2.11

KNN		37.11%	29.22% Underfitting	4.42
Naïve bayes		100 %	93.77%	1.5

From the above figures it's concluded that all models that used bigram features overfit except Naïve bayes model that gave nearly acceptable performance and KNN model that underfit.

Comparing train time of all previous models

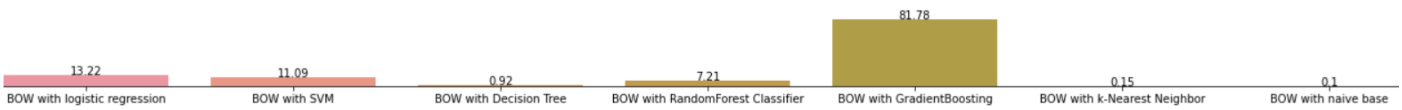


Figure 5 comparing between train time of all models using BOW features

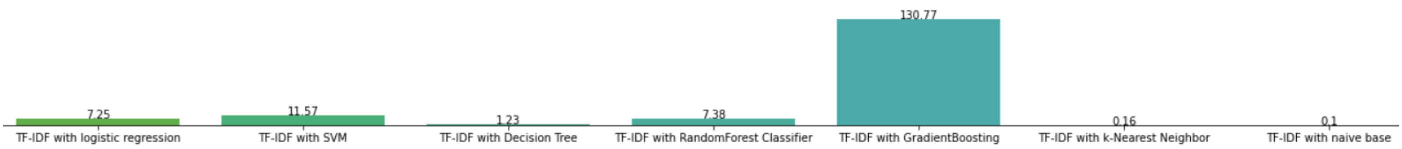


Figure 6 comparing between train time of all models using TFIDF features

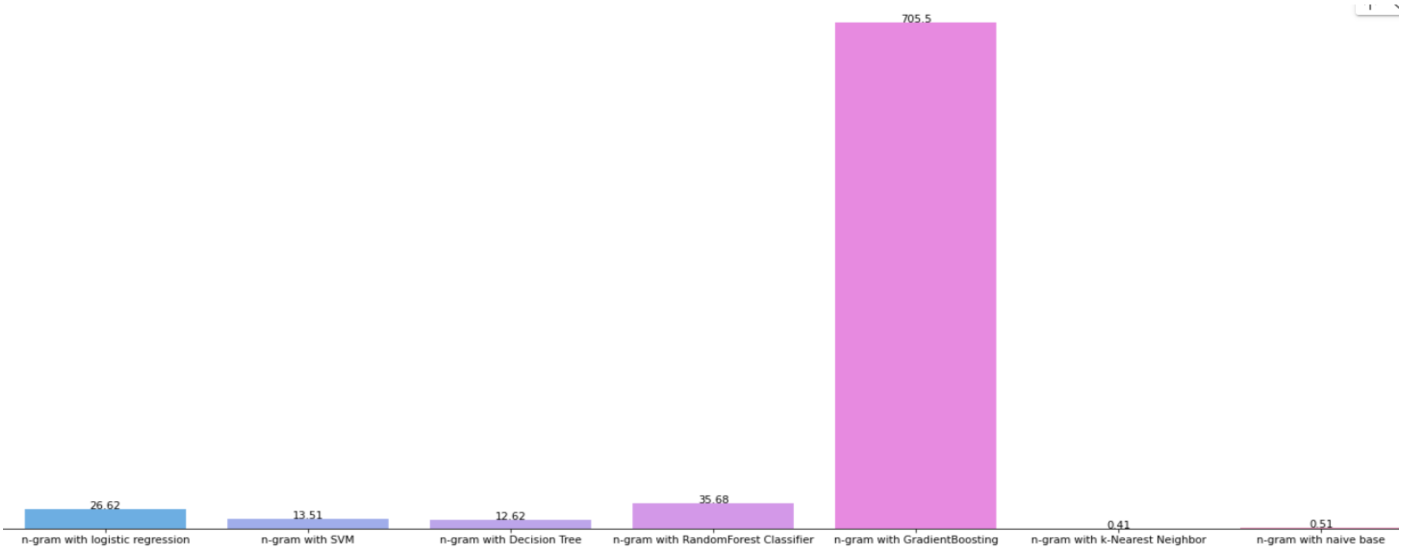


Figure 7 comparing between train time of all models using Bigrams features

From the above figures, it's concluded that training the gradient boosting took the longest time to train regardless the type of features that were used.

Choosing the champion model:

```
def get_champion_model(results):
  f=True
  for key,value in results.items():
    if f:
      best=key
      f=False
    if value[2]>=results[best][2]:
      if value[3]<=results[best][3]:
        best=key
  return best
```

we get the champion model be search about highest models which get validation accuracy that search on them about model get low variance that will be the champion model.

After apply the function we get the TF-IDF with logistic regression is champion model because have highest validation accuracy and low variance.

Varying the number of partitions and the number of words in each partition:

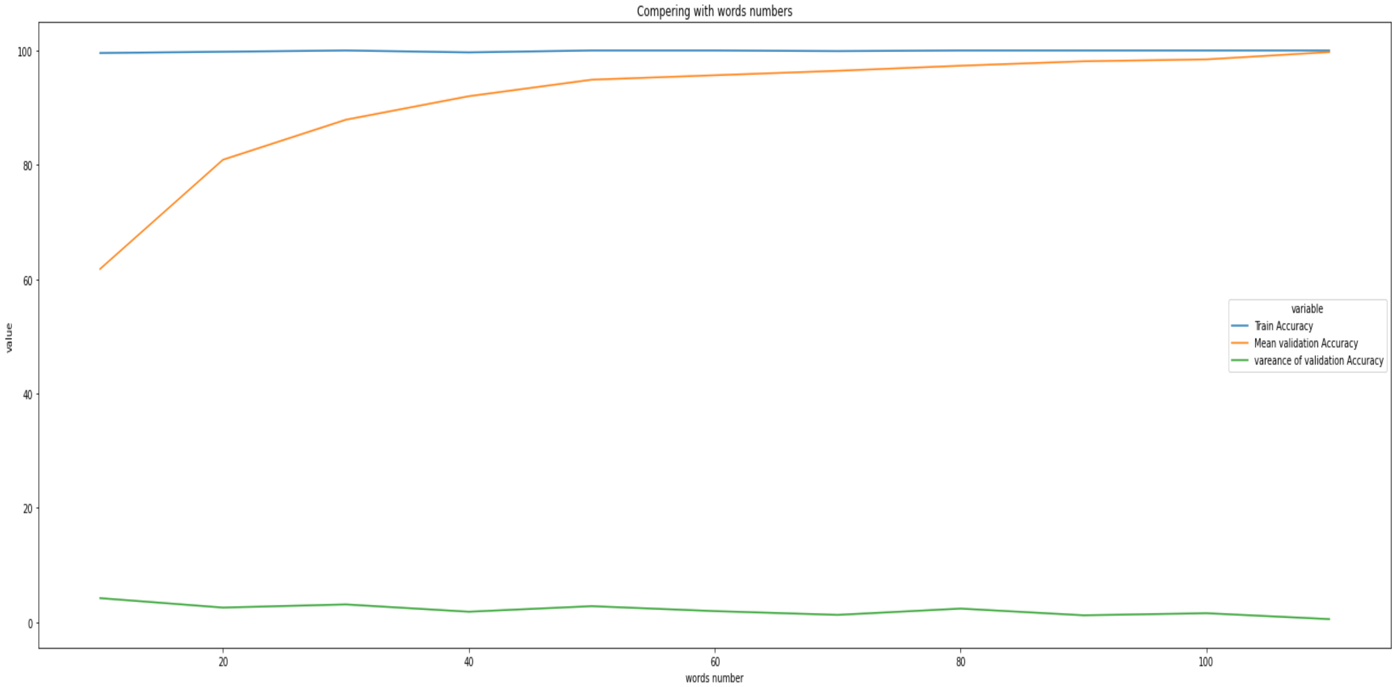


Figure 8 Varying the number of words per partition and comparing accuracies using the champion model

From the above figure, choosing the number of words to be ten words per partition caused overfitting and this is what can conclude from comparing the train accuracy with the mean validation accuracy (the mean validation accuracy is less than train accuracy be about 40%). Increasing the number of words per partitions will decrease the overfitting (the difference between train accuracy and mean validation accuracy).

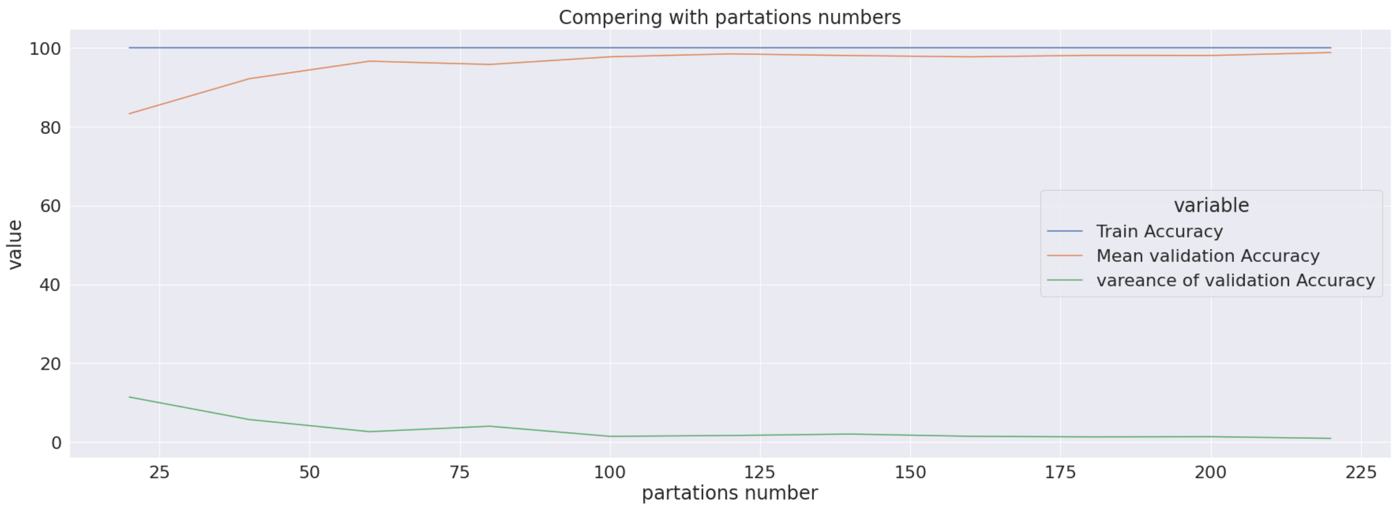


Figure 9 Varying the number of partitions and comparing accuracies using the champion model

From the above figure, choosing the number of partitions to be twenty partition per book caused overfitting and this is what can conclude from comparing the train accuracy with the mean validation accuracy (the mean validation accuracy is less than train accuracy be about 20%). Increasing the number of partitions per book will decrease the overfitting (the difference between train accuracy and mean validation accuracy).

Error Analysis:

after we chose the champion model, we used the test set to test it, then we compared the predicted data and target test data and we analysed the wrong labels to find the most class that our model cannot predict its values, and we said "why?", so we thought that may be because:

1-the book class partitions have general words and are not strongly related to that book.

2-the wrong predicted book class has similarities in words with the actual book class.

The following figure shows the confusion matrix of the champion model on the test data:

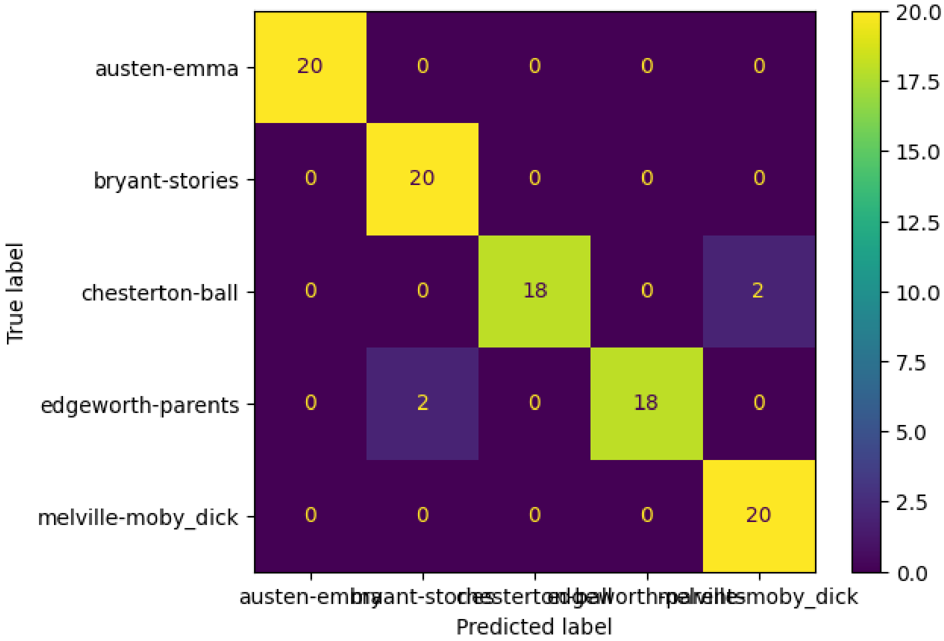


Figure 10 Confusion matrix on the test data

The test accuracy of the champion model is 96% and it can be concluded from the above confusion matrix as the model failed to truly predict four samples from the test data. The misclassified samples is shown in the following figure:

	Partitions	austen-emma.txt	melville-moby_dick.txt	bryant-stories.txt	edgeworth-parents.txt	chesterton-ball.txt	bryant-stories	melville-moby_dick
0	[[would, worst, work, whilst, way, wash, villa...						--X--	NaN
1	[[worn, word, white, whisper, whilst, whenc, w...						--X--	NaN
2	[[within, time, strict, send, second, return, ...						NaN	--X--
3	[[within, wind, whenev, wave, twilight, trust,...						NaN	--X--

Figure 11 The misclassified examples from the test data

The following table shows the word cloud of the four misclassified samples.

The above word clouds show that the word “within” is frequently included in two samples and the word “whilst” too.

Data Augmentation:

The pretrained GPT2 model is used to augment the data by nearly doubling the number of words in each partition. Then the mixture of the real data from the books and the augmented data using the pretrained GPT2 model is used to train the champion model using 10 folds cross validation.

The training accuracy of the model that is trained on the mixed data is: 98.67%

The mean of validation accuracies is: 64.44%

From the above accuracies we can conclude that the model is overfitting, and this is predictable as the augmented data isn’t aligned with our labels that’s why Reinforcement Learning is used in [1] to direct the augmented data to be aligned with the wanted labels.

References:

[1] Liu, Ruibo & Xu, Guangxuan & Jia, Chenyan & Ma, Weicheng & Wang, Lili & Vosoughi, Soroush. (2020). Data Boost: Text Data Augmentation Through Reinforcement Learning Guided Conditional Generation.