Classification Assignment

Ashraqat Torky, Mahmoud Helmy, Sarah Elmasry, Yomna Abdelsattar

May 2021

1 Introduction

1.1 Data preparation

Data used involves five different samples of Gutenberg digital books. The books are of five different authors of the same genre which is fiction. Books names are: Stories to Tell to Children by Sara Cone Bryant, Hamlet by William Shakespeare, Emma by Jane Austen, Moby Dick by Herman Melville, The Wisdom of Father Brown by G. K. Chesterton Each book is partitioned into 200 random samples, each contains 100 words then we remake the same process using the cross validation and use samples of 30 words

1.2 checking word frequencies using 2 methods

1.2.1 Scatter Text:

it is a tool for finding distinguishing terms in small-to-medium-sized corpora, and presenting them in a interactive scatter plot with non-overlapping term labels. in this algorithm we analyze words frequencies in one document against the others . imported from scattertext in python

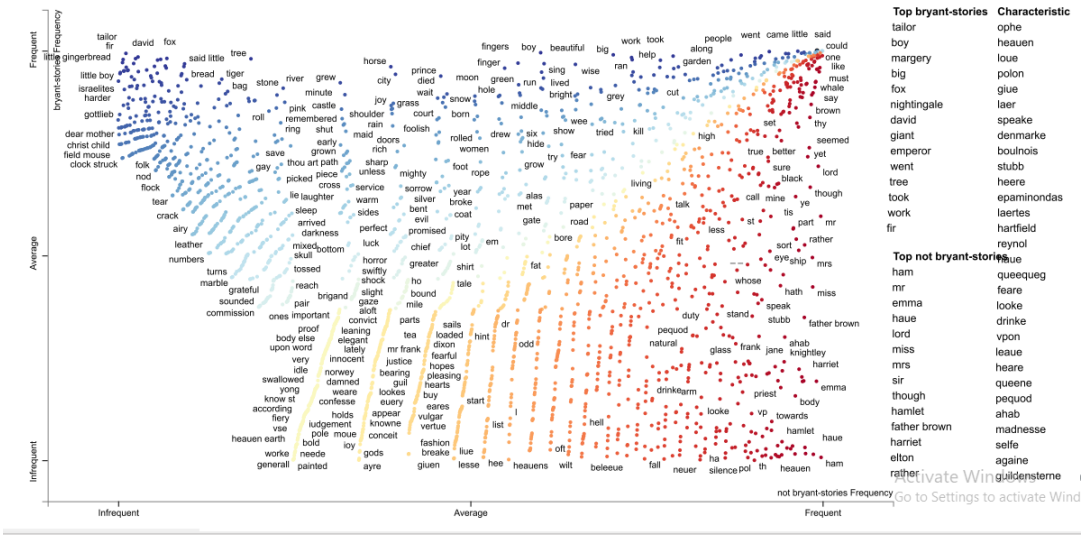


Figure 1: the repetition of words in Bryant's document against the other 4 documents

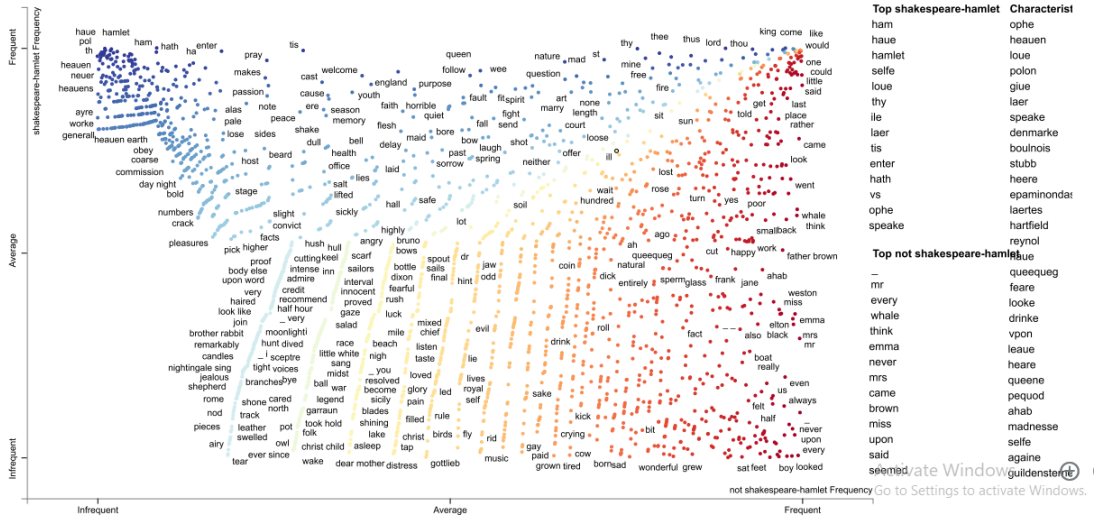


Figure 2: the repetition of words in Shakespeare's document against the other 4 documents

1.2.2 Word Cloud:

is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance

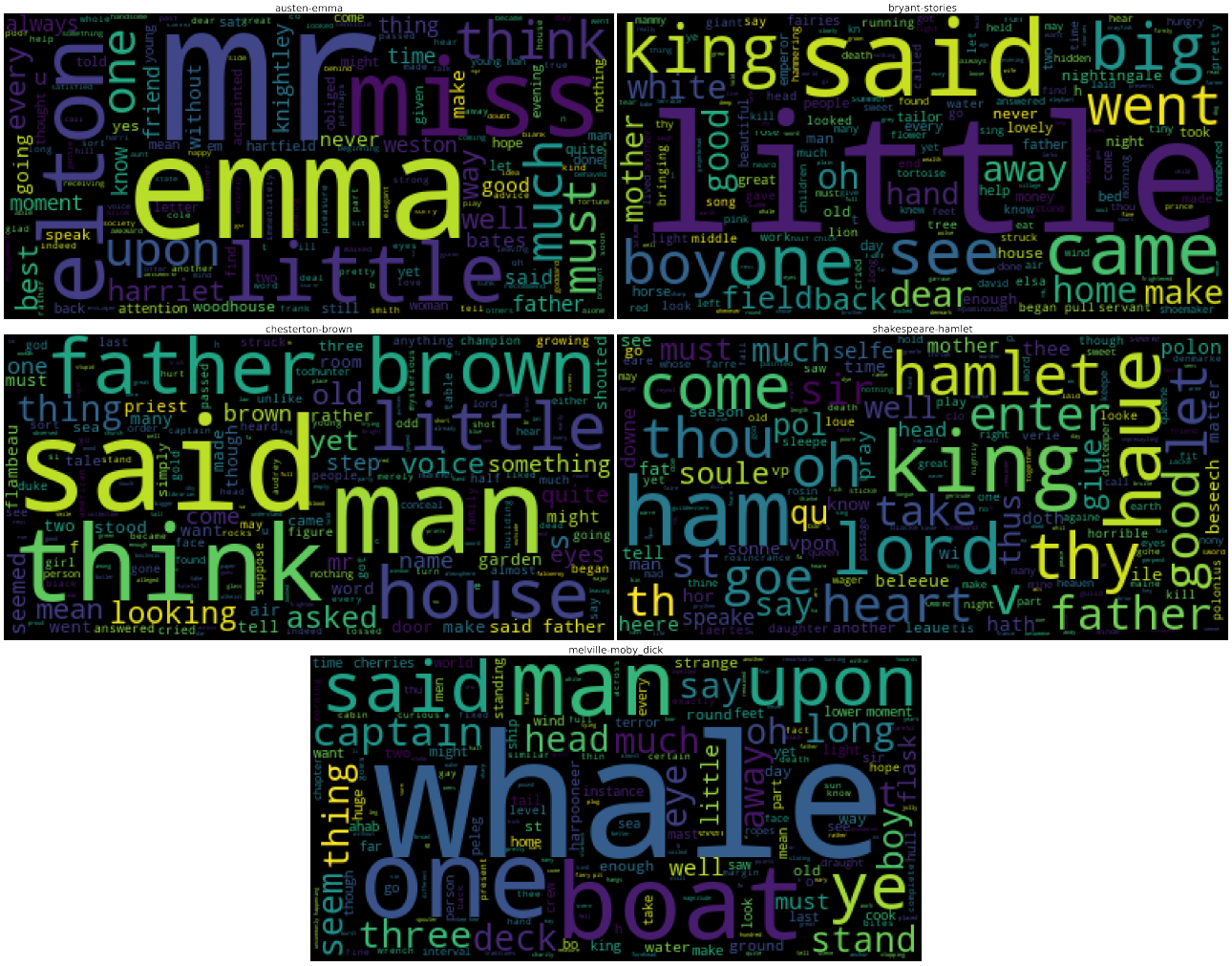


Figure 3: Word cloud for each document ,**Top left**:Emma by Jane Austen,**Top right**:Stories to Tell to Children by Bryant,**Bottom left**:The Wisdom of Father Brown by Chesterton ,**Bottom right**:Hamlet by William Shakespeare,**middle Bottom**: Moby Dick by Herman Melville

1.3 Data prepossessing

Stop words, punctuation, and special characters are eliminated from the data

2 Transformation and Feature Engineering

For feature engineering, different techniques are used: BOW, TF-IDF, n-gram and doc2word

1. BOW creates a set of vectors containing the count of word occurrences in the document
2. TF-IDF Contains information on the more important words and the less important ones as well.
3. n-gram takes a text variable as input and produces strings corresponding to sliding a window of (user-configurable) n words
4. doc2word represents documents as a vector and is a generalizing of the word2vec method

3 Training a model

For machine learning to predict which author or book a specific document is related to. SVM (Support Vector Machine), Decision Tree, k-Nearest Neighbor are used.

3.1 SVM

In the SVM algorithm, each data point is plotted in n-dimensional space (where n is number of features you have). Then, classification is done by finding the hyper-plane that separates the different classes.

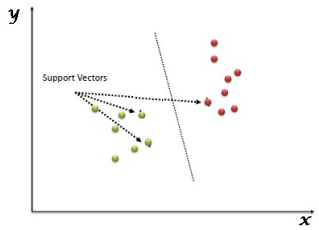


Figure 4: Support Vector Machine

3.2 Decision Tree

a tree-structured classifier, where internal nodes represent the features of a data set, branches represent the decision rules and each leaf node represents the outcome. The decisions or the test are performed on the basis of features of the given data set.

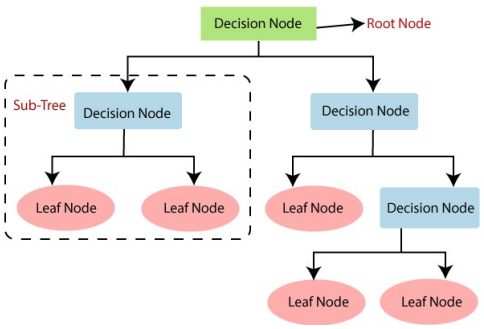


Figure 5: Decision Tree

3.3 k-Nearest Neighbor

K-nearest neighbors (KNN) algorithm uses feature similarity to predict the values of new data points and how each data point will be assigned a value based on how closely it matches the points in the training set.

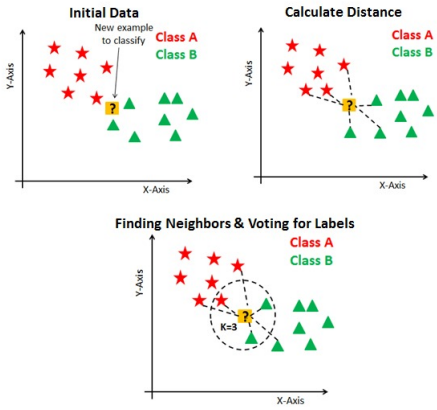


Figure 6: Decision Tree

4 Accuracy Scoring

4.1 100 Word per partition

	BOW	TF-IDF	ngram	doc2vec
SVM	98.3 %	98.3 %	76 %	98.6 %
Decision Tree	51 %	94 %	84 %	55 %
KNN	83.3 %	86 %	77 %	56 %

5 Evalution

5.1 Reducing words to 30 words per partition and using Cross Validation

	BOW	TF-IDF	ngram	doc2vec
SVM	88.9 %	91 %	52 %	88 %
Decision Tree	71 %	69 %	58 %	55 %
KNN	76 %	80 %	66 %	57 %

6 Visualization

6.1 Error Analysis

6.1.1 Bag Of Word

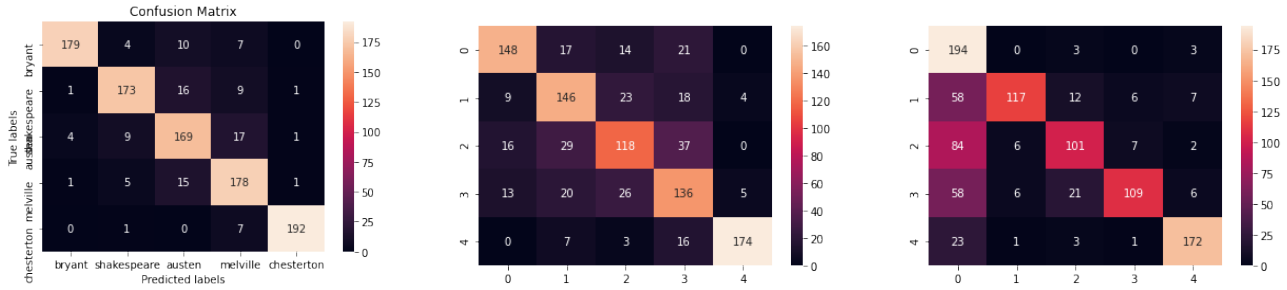


Figure 7: SVM Confusion Matrix

Figure 8: Decision Tree Confusion Matrix

Figure 9: KNN Confusion Matrix

6.1.2 TF-IDF

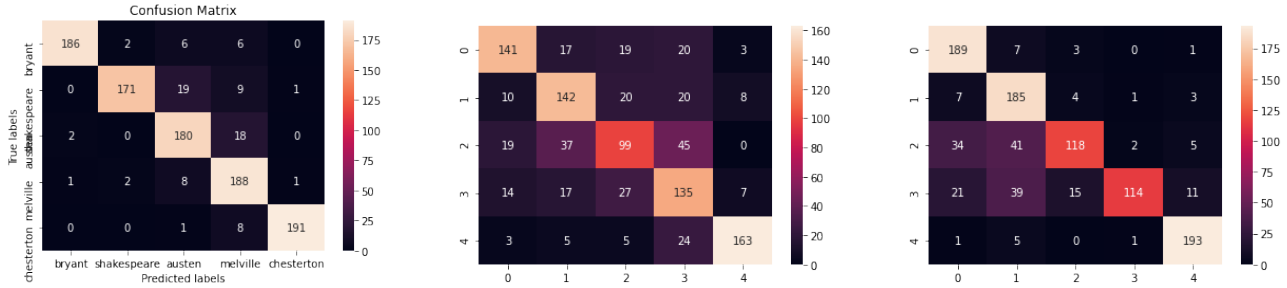
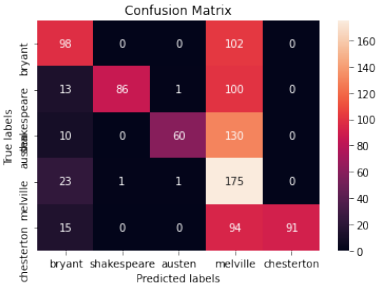


Figure 10: SVM Confusion Matrix

Figure 11: Decision Tree Confusion Matrix

Figure 12: KNN Confusion Matrix

6.1.3 NGrams



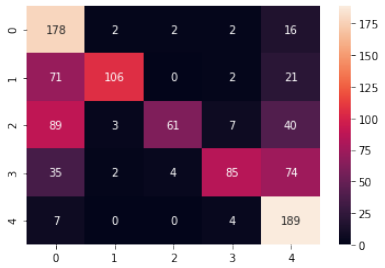
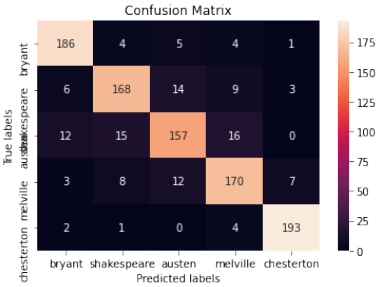


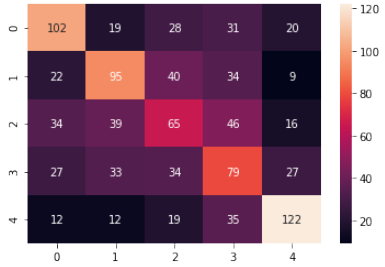
Figure 13: SVM Confusion Matrix

Figure 14: Decision Tree Confusion Matrix

Figure 15: KNN Confusion Matrix

6.1.4 Doc2Vec





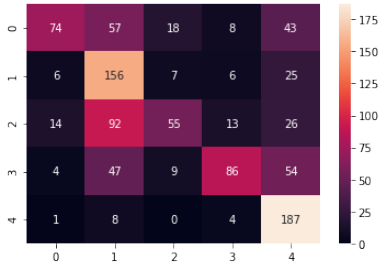


Figure 16: SVM Confusion Matrix

Figure 17: Decision Tree Confusion Matrix

Figure 18: KNN Confusion Matrix

6.2 Learning Curves

6.2.1 Bag Of Word

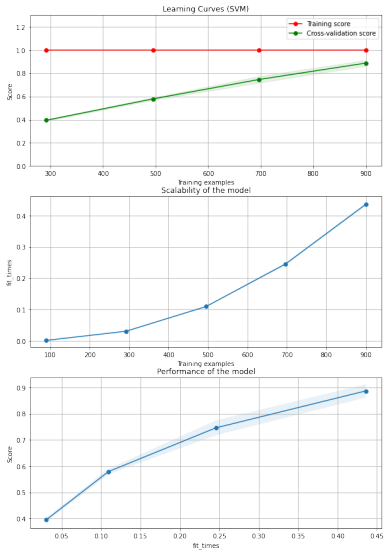


Figure 19: SVM Learning Curves

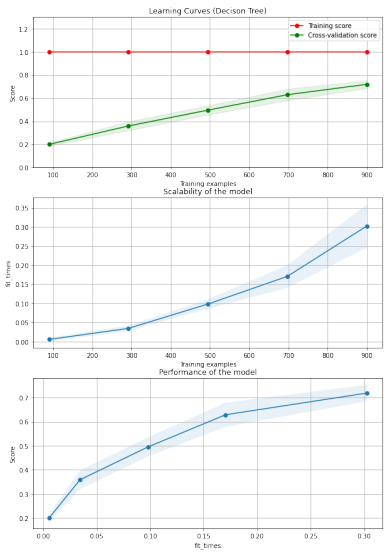


Figure 20: Decision Tree Learning Curves

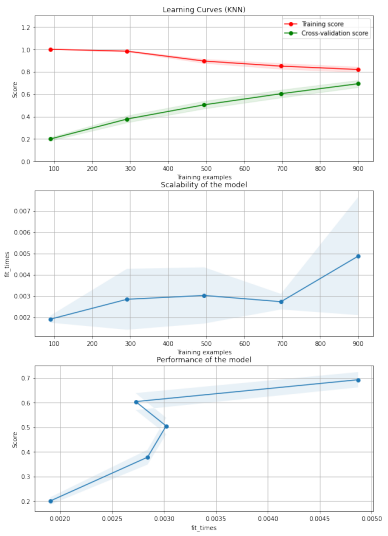


Figure 21: KNN Learning Curves

6.2.2 TF-IDF

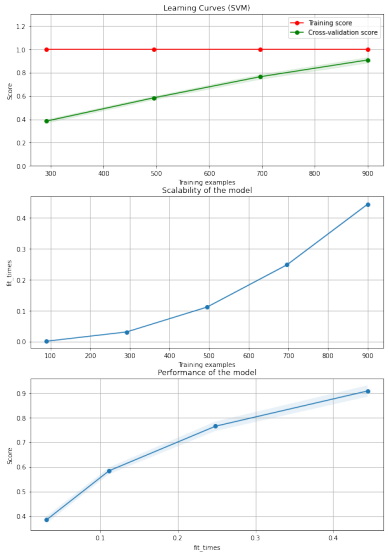


Figure 22: SVM Learning Curves

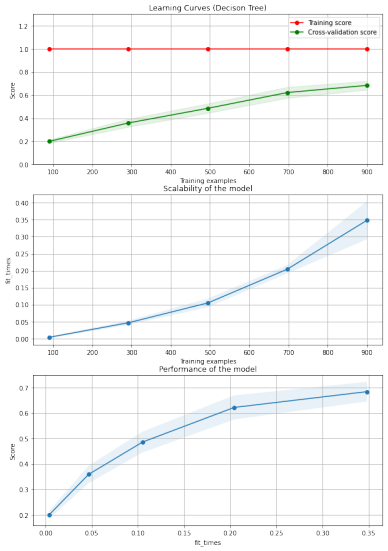


Figure 23: Decision Tree Learning Curves

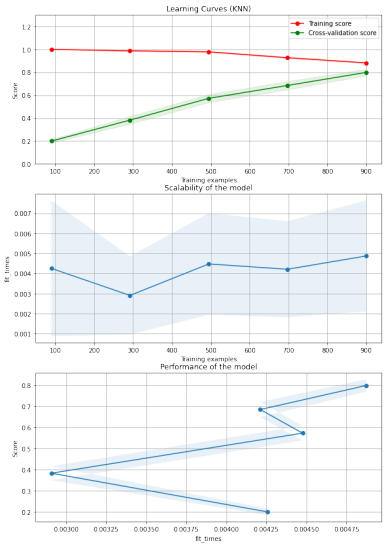


Figure 24: KNN Learning Curves

6.2.3 NGrams

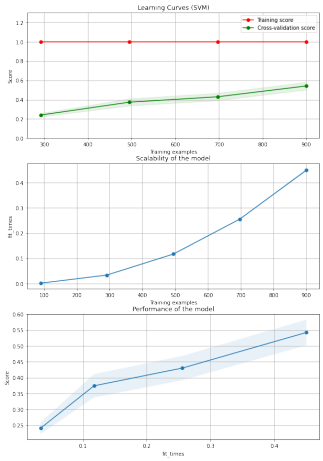


Figure 25: SVM Learning Curves

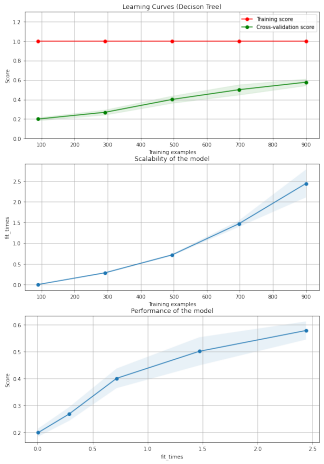


Figure 26: Decision Tree Learning Curves

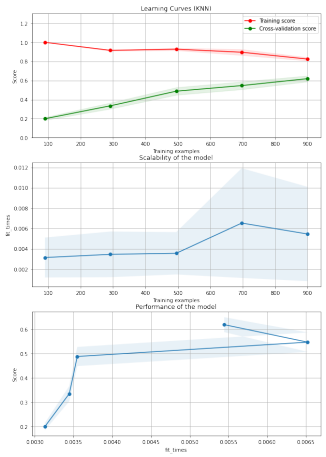


Figure 27: KNN Learning Curves

6.2.4 Doc2Vec

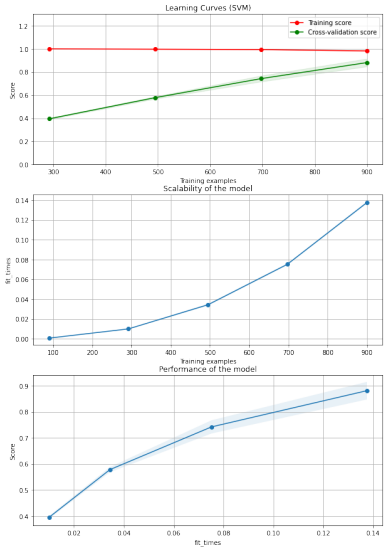


Figure 28: SVM Learning Curves

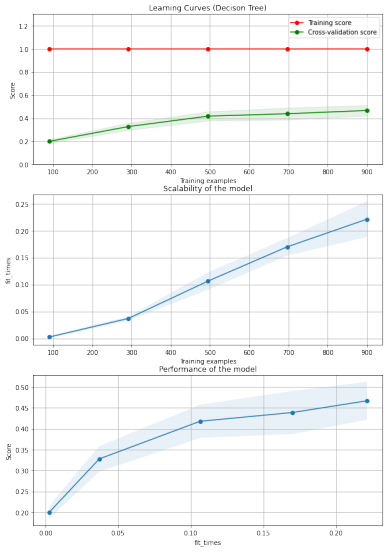


Figure 29: Decision Tree Learning Curves

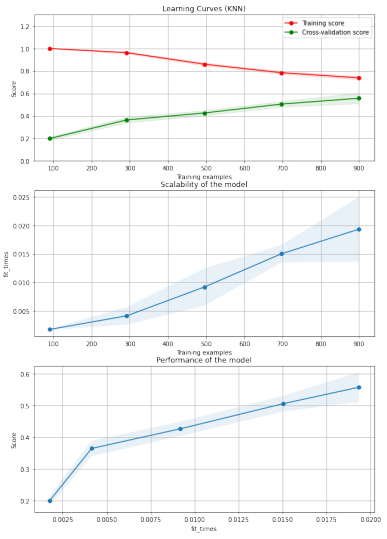


Figure 30: KNN Learning Curves

7 Bias-Variance trade-off

7.1 Brief:

In Machine Learning, the ideal algorithm has low bias and can accurately model the true relationship (training phase), and it has low variability by producing consistent predictions across different data-sets (testing phase) This is done by finding the ideal spot between a simple model (high bias/ low variance) and a complex model (low bias/ high variance)

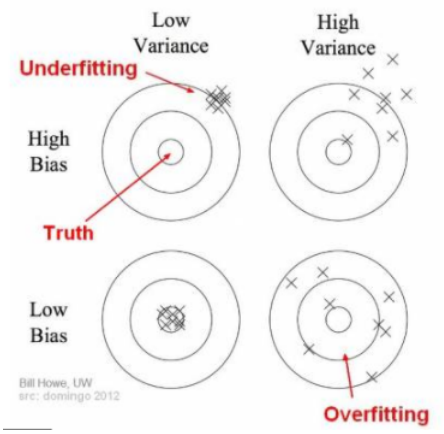


Figure 31: Bias/Variance Trade-off:

7.2 Relation of(Bias/Variance) with learning curve:

When training score and cross-validation score are far away from each other, this is probably due to high variance and low bias so this model has over-fitting issues. Otherwise, when training score and cross-validation score are close to each other, this probably due to low variance and high bias, so this model has under-fitting modeling issues.

7.3 Hyper-parameter tuning for each algorithm:

1. **In support vector machine** (SVM), cost (c) parameter decides bias-variance. A large C gives you low bias and high variance. Low bias because you penalize the cost of misclassification a lot. Large C makes the cost of misclassification high, thus forcing the algorithm to explain the input data stricter and potentially over-fit. A small C gives you higher bias and lower variance. Small C makes the cost of misclassification low, thus allowing more of them for the sake of wider "cushion"

2. **In k-nearest neighbors**, trade-off can be changed by increasing the value of k which increases the number of neighbors that contribute to the prediction and in turn increases the bias of the model and low variance.

3. **In decision trees**, pruning of tree is a method to reduce variance. It reduces the size of decision trees by removing sections of the tree that provide little power to classify instances.

8 Conclusion

Cross-Validation is a very powerful tool. It helps us better use our data, it gives us much more information about our algorithm performance and it handles the situation when we try to achieve a reasonable accuracy of the model ,so when differentiating between algorithms we can conclude that the method of extracting the features plays a crucial role in the accuracy of the model and from graphs it is clearly shown that TF-IDF is the best in comparing with other feature engineering techniques. for comparing the model we can conclude that best results come from SVM .