CM50265 - Group 39
Coursework 2 : Sentiment Analysis

Contribution in Percentage Group 39

SR2226	100
SSS243	100
PK724	100

Department of Computer Science
University of Bath
April 2021

List of Figures

1 Introduction

2 What is SVM?

Support Vector Machines (SVM) have recently gained prominence in the field of machine learning and pattern classification. [Vapnik, 1999] Classification is achieved by realizing a linear or non-linear separation surface in the input space. [Vishwanathan and Murty, 2002]
SVM is a powerful method for building a classifier. It aims to create a decision boundary between two classes that enables the prediction of labels from one or more feature vectors. [Noble, 2006]

3 Built-in Kernel

3.1 Linear Kernel

A Linear SVC's (Support Vector Classifier) aim is to fit the provided data, which return best fit Hyperplane which in-turns categorised data.
For this Kernel, We have used K-Fold to split the train/test data. K-Fold cross-validator gives us train/test indices to split data in train/test sets where, we use one fold as validator and rest K-1 fold act as training test. Figure below explains the K-fold technique. Mathematical expression for Linear kernel:

$$K(X,Y) = X^T Y$$

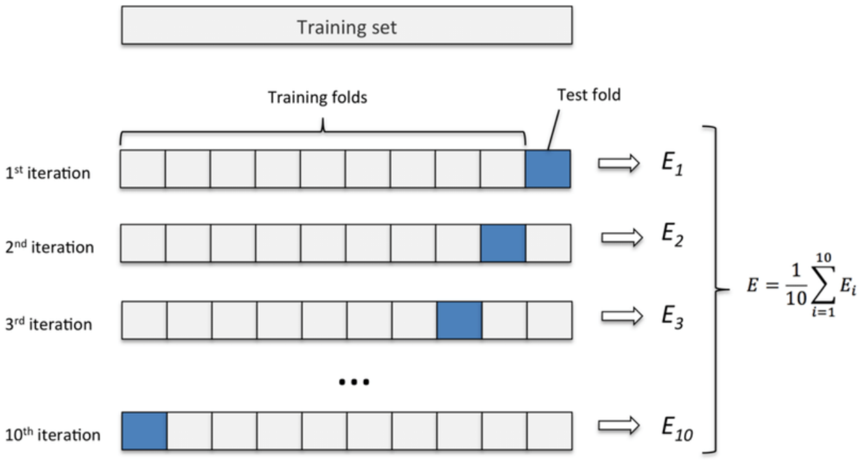


Figure 1: Diagram of k-fold cross-validation with k = 10. Image from [Ashfaque and Iqbal, 2019]

3.1.1 Hyper-Parameter Tuning for Linear SVM

SVC poses a quadratic optimization problem which is nothing but the misclassification of data. To prevent this we have used to Penalty parameter(C) which helps in setting the degree of importance of misclassification.

C: [0.1 , 1, 10, 100, 1000]

For Hyper-Parameter tuning we have used SKLearn's GridSearchCV. It aids in fitting the model to training data by looping over specified hyper-parameters. Finally, we can choose the optimal parameters from the hyper-parameters obtained.
Figure 2 shows the accuracy of the linear kernel after performing the hyper-parameter tuning. Final accuracy was 0.8452 at C = 1.

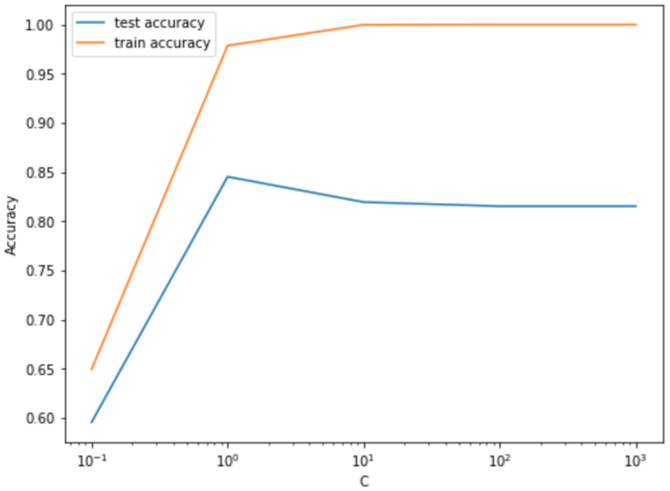


Figure 2: Linear Kernel accuracy after Hyper-paramter Tuning for different values of Regularization Parameter(C)

4 RBF Kernel

RBF (Radial Basis Function) kernel) is SKLearn's inbuilt Kernel which is used when data set is linearly inseparable. RBF Kernel uses two hyper-parameter:

1. gamma - Defines the extent to which a single training example has an impact. Low values of gamma means 'far' and high values means 'close'. Value of gamma : [1e-2, 1e-3, 1e-4,1e-5]

2. C (Penalty/Regularization Parameter). Value of C chosen: [1, 10, 100, 1000]

Mathematical expression for RBF kernel:

$$K(X_1, X_2) = exp(\frac{-||X_1 - X_2||^2}{2\sigma^2})$$

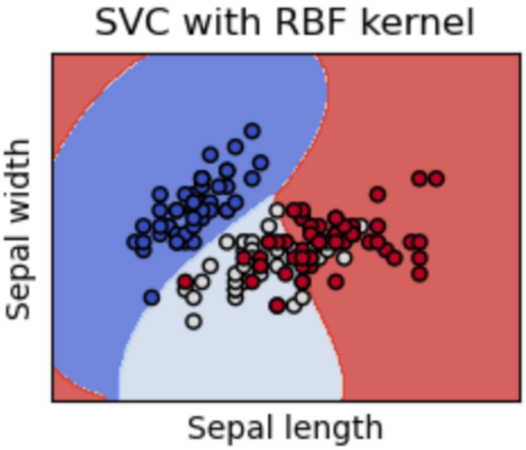


Figure 3: Illustration of classification in RBF Kernel

Figure 4 shows the accuracy of the RBF Kernel after Hyper-paramter tuning. It was observed that best accuracy was 85.06 where gamma = 1e-04 and c =1.

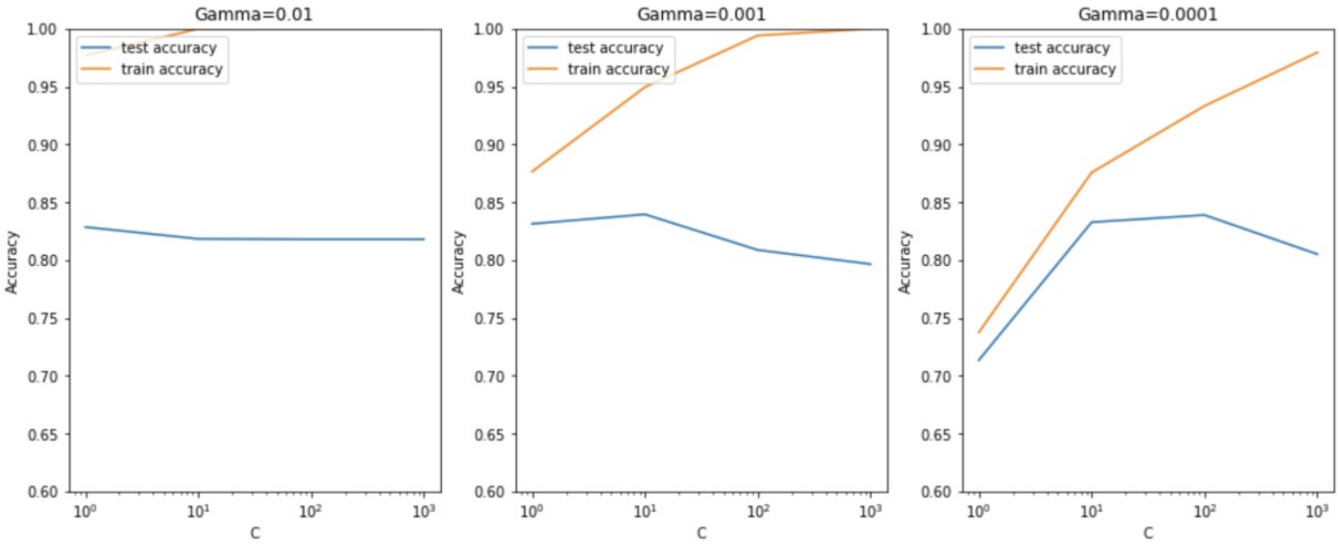


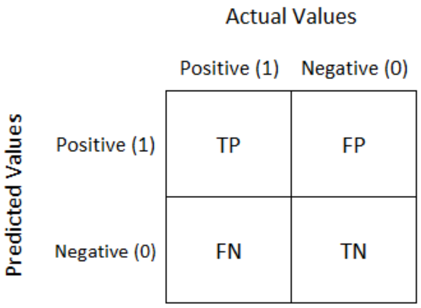
Figure 4: RBF Kernel accuracy after Hyper-parameter tuning

5 Custom Kernel

In the custom kernel we are passing the x train and y train data and the parameter which is equal to three. The mathematical expression for this operation chosen is:

$$K(X,Y) = (1 + X^T Y)^p$$

We create and used an instance of SVM(Simple Vector Machine) to fit our data.
clf = svm.SVC(kernel=my_kernel)
clf.fit(X_train, Y_train)
The predict() function is used to perform a a prediction for each X_test instance.
y_pred = clf.predict(X_test)
Then a confusion matrix is created for the y_pred and y_test data set. A confusion matrix is a performance reading mechanism where machine learning classification problem where output can be two or more classes.



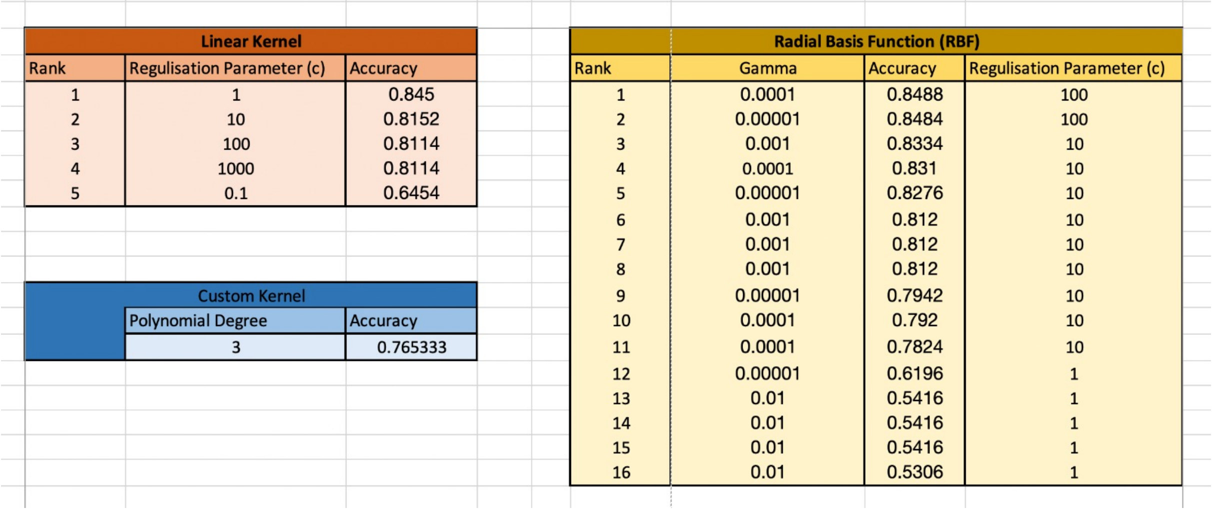
		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Figure 5: Understanding Confusion Matrix

The accuracy, precision and sensitivity of Y_pred and Y_test is printed after that.
The results of this kernel was not as good as the built-in kernels.

6 Comparison of different Kernel

The table shows the results after hyper-parameter tuning of different SVM models having -Linear, RBF and customized kernel. The highest accuracy for linear kernel is achieved when the regularisation parameter(C) is 1. In case of RBF the highest accuracy is achieved when the regularisation parameter is 0.001 and the gamma is .For customized kernel we achieve the highest accuracy when p(polynomial degree) is 3.



Linear Kernel		
Rank	Regulisation Parameter (c)	Accuracy
1	1	0.845
2	10	0.8152
3	100	0.8114
4	1000	0.8114
5	0.1	0.6454

Custom Kernel		
	Polynomial Degree	Accuracy
	3	0.765333

Radial Basis Function (RBF)			
Rank	Gamma	Accuracy	Regulisation Parameter (c)
1	0.0001	0.8488	100
2	0.00001	0.8484	100
3	0.001	0.8334	10
4	0.0001	0.831	10
5	0.00001	0.8276	10
6	0.001	0.812	10
7	0.001	0.812	10
8	0.001	0.812	10
9	0.00001	0.7942	10
10	0.0001	0.792	10
11	0.0001	0.7824	10
12	0.00001	0.6196	1
13	0.01	0.5416	1
14	0.01	0.5416	1
15	0.01	0.5416	1
16	0.01	0.5306	1

Figure 6: Comparison of different Kernel

7 Misclassification

For Misclassification analysis we have plotted graph (Figure 8) between number of misclassified reviews with respective word length. We conclude that misclassification was higher when review length is in the range of 100 to 150, misclassification were significantly high.

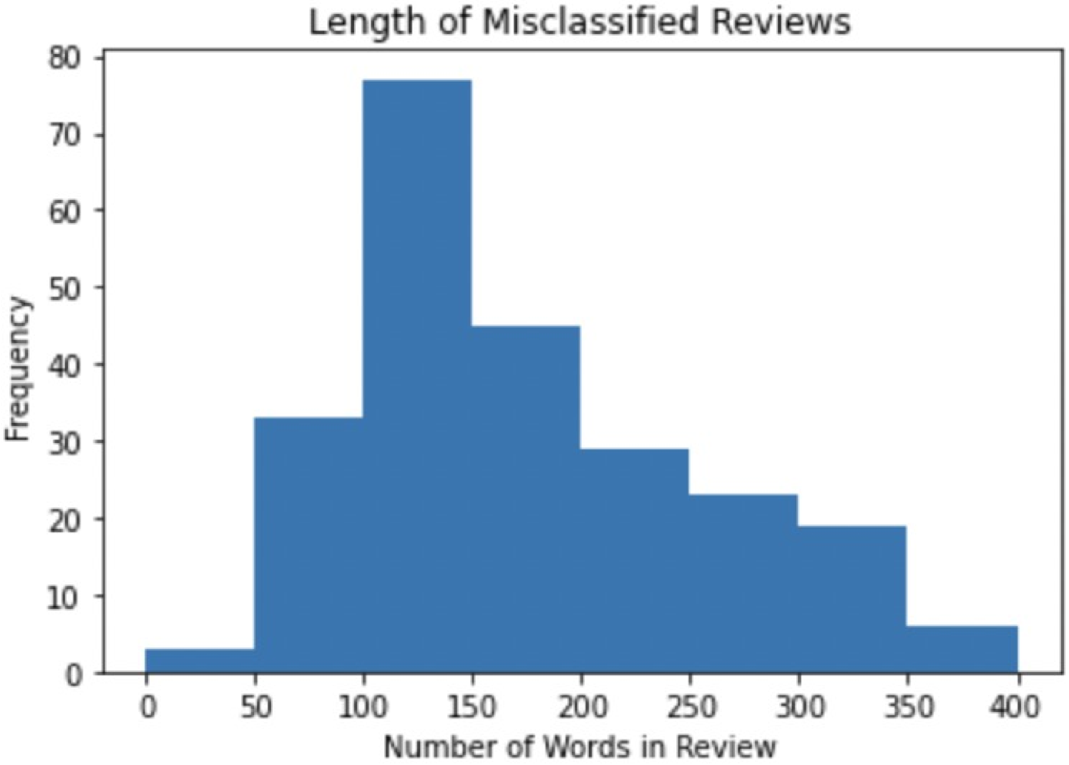


Figure 7: Misclassification analysis

8 Boosting

For boosting we have created a Decision Tree classifier and have used hyper-parameter tuning to get the max-depth. Once we get the max-depth of the Decison Tree we have used it as the base-estimator for the ADABoost algorithm. In order to fine tune the ADABoost we have looped over the n-estimator and got the accuracy of 83.53 for n-estimator=500. Figure 9 shows the ADABoost performance.

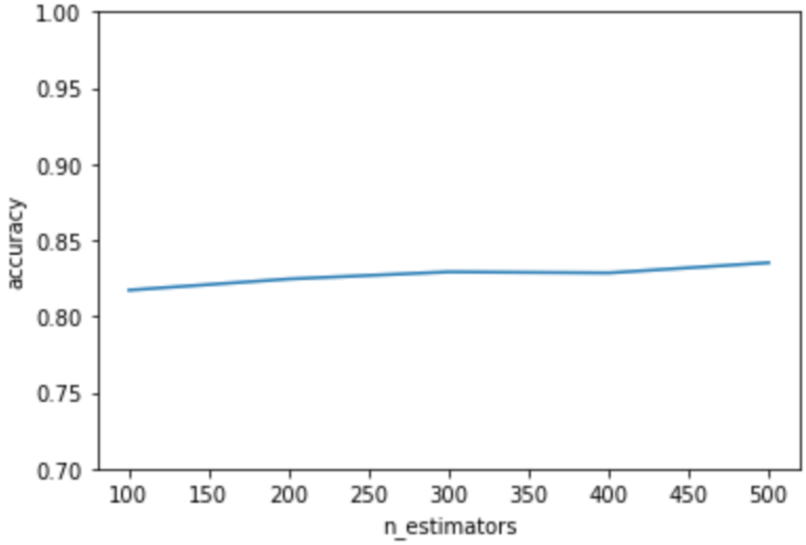


Figure 8: ADABosst performance Accuracy VS Estimator

References

[Ashfaque and Iqbal, 2019] Ashfaque, J. and Iqbal, A. (2019). Introduction to support vector machines and kernel methods.

[Noble, 2006] Noble, W. S. (2006). What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567.

[Vapnik, 1999] Vapnik, V. (1999). *The nature of statistical learning theory.* Springer science & business media.

[Vishwanathan and Murty, 2002] Vishwanathan, S. and Murty, M. N. (2002). Ssvm: a simple svm algorithm. In *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290)*, volume 3, pages 2393–2398. IEEE.