APPLIED MACHINE LEARNING

ASSIGNMENT 2

Srinikethan Pusthay (SXP210162)

Goal: Implement a SVM and Decision Tree model on the dataset to predict the rented bike count.

Dataset:
The dataset contains count of public bikes rented at each hour in Seoul Bike haring System with the corresponding Weather data and Holidays information.

Tasks:

Part1:
We have used the data set from assignment 1, and we have modified the features same as previously. To convert the dataset into binary classification problem we have converted the output to class label. We have considered the mean value of bike count 704 as the threshold and transformed the column to 1 if the values are above threshold and 0 if the values are below threshold. The data is then split into train and test by 70% and 30%.

Part2:
For Support Vector Machines, we have used packages from scikit learn. We have performed various experiments using linear, polynomial, radial basis function and sigmoid kernels by varying parameters.

Part3:
For Decision Tree, we have used package from scikit learn, and performed various experiments using gini index and entropy to split on variables.

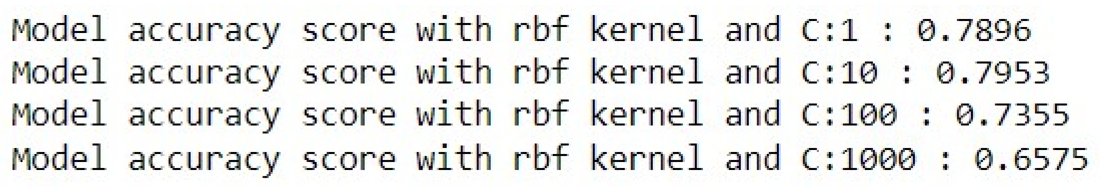
Part4:
For the above Support Vector Machine and Decision Tree models, we have performed Cross Validation and the respective results are mentioned in Experiments and results.

Experiments and Results:

For SVM, we have used packages from Scikit learn. There are two main parameters to define while using this, C and gamma. C is a hyperparameter in SVM to control error. Gamma is used when we use RBF Kernel and gamma decides that how much curvature we want in a decision boundary.

We run the data with different models varying the C value and print out the accuracy score of test data.

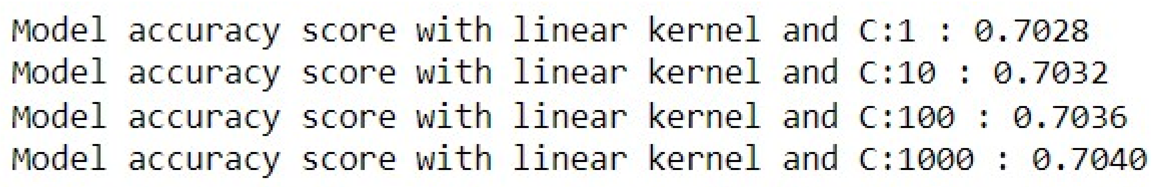
- SVM using RBF kernel varying C



```
Model accuracy score with rbf kernel and C:1 : 0.7896
Model accuracy score with rbf kernel and C:10 : 0.7953
Model accuracy score with rbf kernel and C:100 : 0.7355
Model accuracy score with rbf kernel and C:1000 : 0.6575
```

  From the above result, we can observe that accuracy is highest at C=10. And as C increases accuracy decreases.

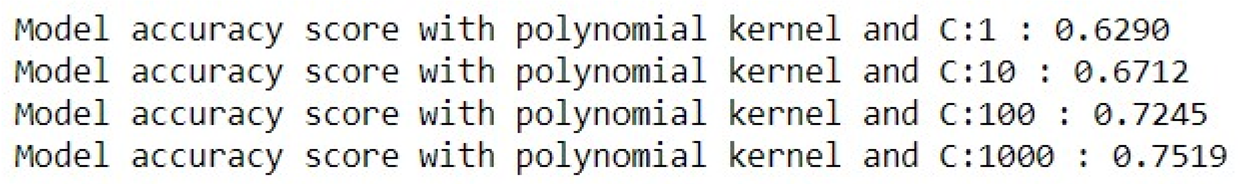
- SVM using Linear kernel varying C



```
Model accuracy score with linear kernel and C:1 : 0.7028
Model accuracy score with linear kernel and C:10 : 0.7032
Model accuracy score with linear kernel and C:100 : 0.7036
Model accuracy score with linear kernel and C:1000 : 0.7040
```

  Accuracy is around 70% and as C increase, accuracy increase but in a very small rate.

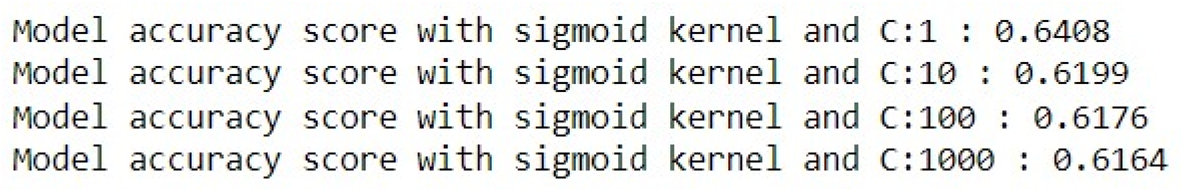
- SVM using Polynomial kernel varying C



```
Model accuracy score with polynomial kernel and C:1 : 0.6290
Model accuracy score with polynomial kernel and C:10 : 0.6712
Model accuracy score with polynomial kernel and C:100 : 0.7245
Model accuracy score with polynomial kernel and C:1000 : 0.7519
```

  With C as 1, we have the lowest accuracy of the model so far. And as C increases, accuracy increases. But it isn’t the best model considering the high value of C.

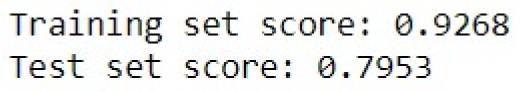
- SVM using Sigmoid kernel varying C



```
Model accuracy score with sigmoid kernel and C:1 : 0.6408
Model accuracy score with sigmoid kernel and C:10 : 0.6199
Model accuracy score with sigmoid kernel and C:100 : 0.6176
Model accuracy score with sigmoid kernel and C:1000 : 0.6164
```

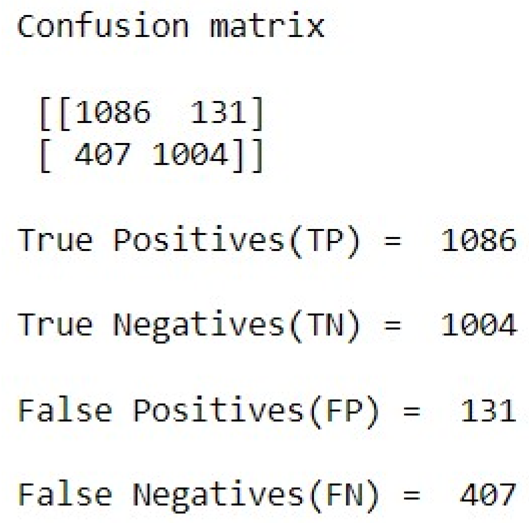
  Here, as C increases accuracy decrease which is the opposite of what we have seen so far.

From the above experiments, the best Kernel is RBF (Radial Basis Function) with C=10. Let’s calculate few performance parameters for this model.



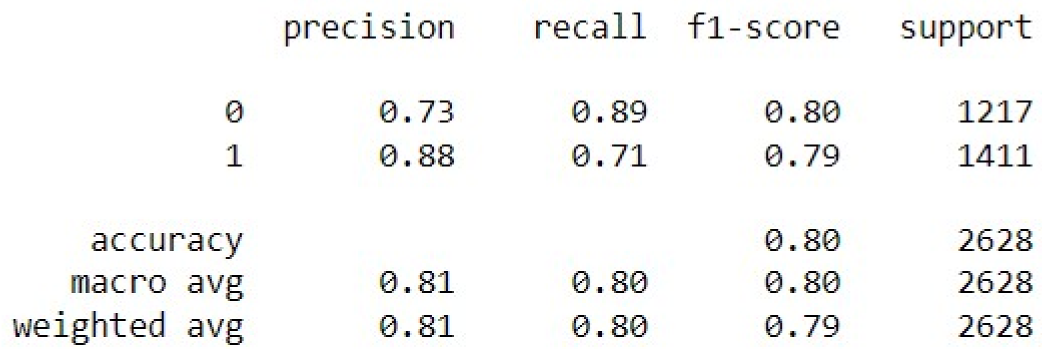
```
Training set score: 0.9268
Test set score: 0.7953
```

Confusion Matrix of Test Data:

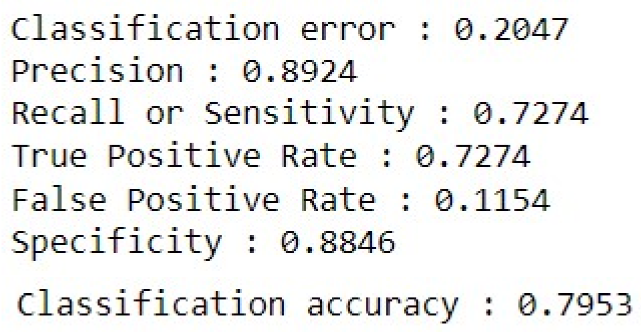


```
Confusion matrix

 [[1086  131]
 [ 407 1004]]

True Positives(TP) =  1086

True Negatives(TN) =  1004

False Positives(FP) =  131

False Negatives(FN) =  407
```

So, this confusion matrix shows (1086+1004=) 2090 correct predictions and (131+407=) 538 incorrect predictions.

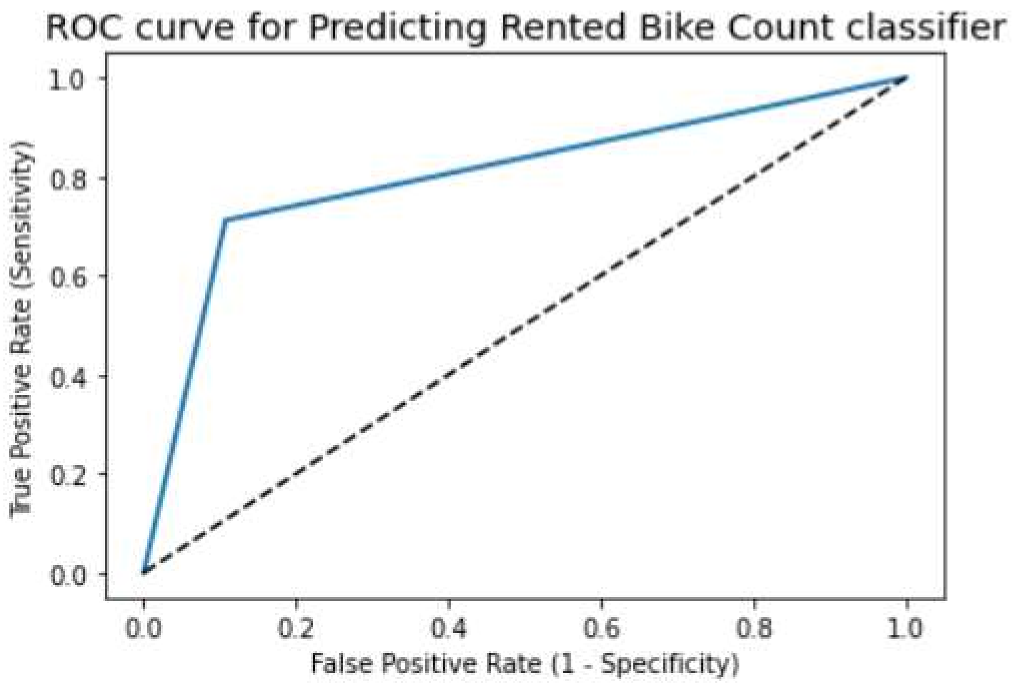


```
              precision    recall  f1-score   support

           0       0.73      0.89      0.80      1217
           1       0.88      0.71      0.79      1411

    accuracy                           0.80      2628
   macro avg       0.81      0.80      0.80      2628
weighted avg       0.81      0.80      0.79      2628
```



```
Classification error : 0.2047
Precision : 0.8924
Recall or Sensitivity : 0.7274
True Positive Rate : 0.7274
False Positive Rate : 0.1154
Specificity : 0.8846

Classification accuracy : 0.7953
```

ROC Curve stands for Receiver Operating Characteristic Curve. An ROC Curve is a plot which shows the performance of a classification model at various classification threshold levels. An ROC Curve plots TPR vs FPR at different classification threshold levels. If we lower the threshold levels, it may result in more items being classified as positive.



ROC AUC stands for Receiver Operating Characteristic - Area Under Curve. It is a technique to compare classifier performance. In this technique, we measure the area under the curve (AUC). A perfect classifier will have a ROC AUC equal to 1, whereas a purely random classifier will have a ROC AUC equal to 0.5.

```
ROC AUC : 0.8020
```

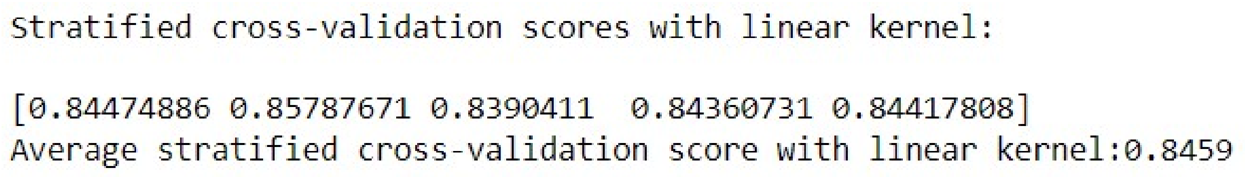
```
Cross validated ROC AUC : 0.9157
```

Cross validation has increased the value of ROC AUC.

Next, we are performing Cross Validation on these various kernels. And then compare the results.

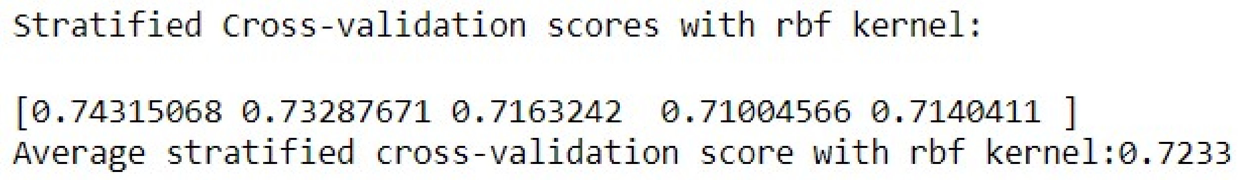
As we have an imbalanced dataset, we are using Stratified K-Fold cross validation. In this, we split the data such that the proportions of between classes are the same in each fold as they are in whole dataset.

- CV for linear kernel



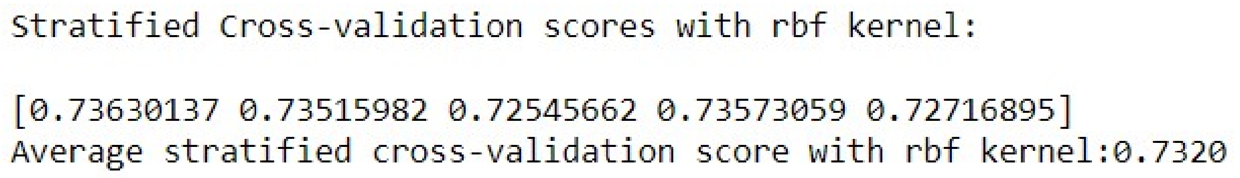
```
Stratified cross-validation scores with linear kernel:

[0.84474886 0.85787671 0.8390411  0.84360731 0.84417808]
Average stratified cross-validation score with linear kernel:0.8459
```

- CV for RBF kernel



```
Stratified Cross-validation scores with rbf kernel:

[0.74315068 0.73287671 0.7163242  0.71004566 0.7140411 ]
Average stratified cross-validation score with rbf kernel:0.7233
```

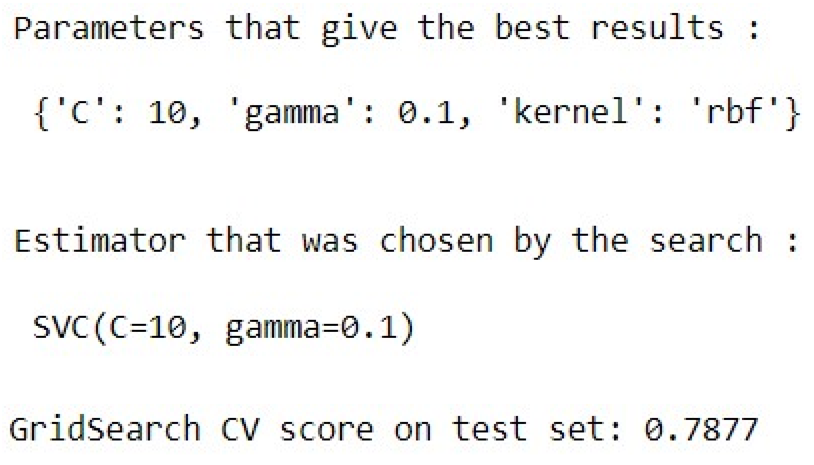
- CV for Polynomial kernel



```
Stratified Cross-validation scores with rbf kernel:

[0.73630137 0.73515982 0.72545662 0.73573059 0.72716895]
Average stratified cross-validation score with rbf kernel:0.7320
```

From the above results, we can say that scores have been increased for Linear kernel compared to previous highest linear kernel accuracy at 0.70, but for RBF kernel it doesn't improve the model performance.

For optimizing both C and gamma, we are using GridSearch CV, to obtain the best model parameters.



```
Parameters that give the best results :

 {'C': 10, 'gamma': 0.1, 'kernel': 'rbf'}


Estimator that was chosen by the search :

 SVC(C=10, gamma=0.1)


GridSearch CV score on test set: 0.7877
```

Decision Tree Algorithms for the given Dataset:

We are using GINI Index as the criteria for developing the decision tree. In information Gain, due to involvement of logarithmic terms, model processing time increases. So, we are using GINI index to process and evaluate.

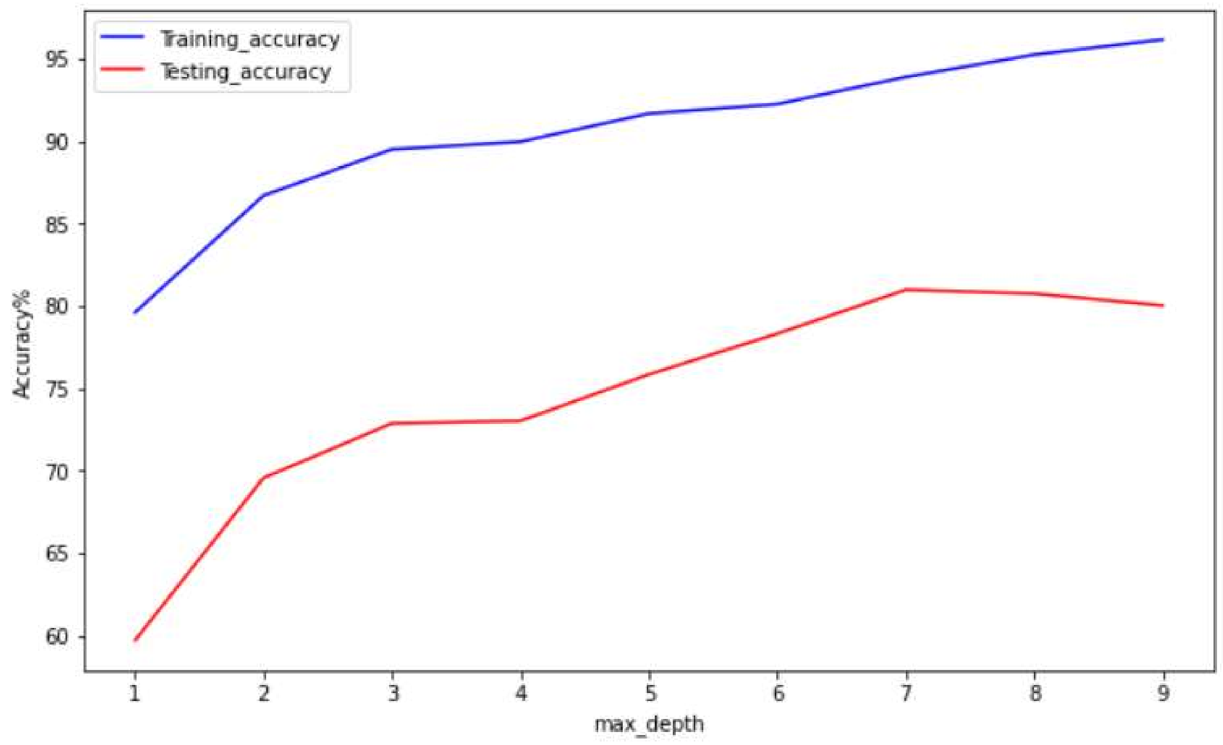
Accuracy of Decision Tree with GINI criteria.

```
Accuracy with Decision Tree: 78.35%
```

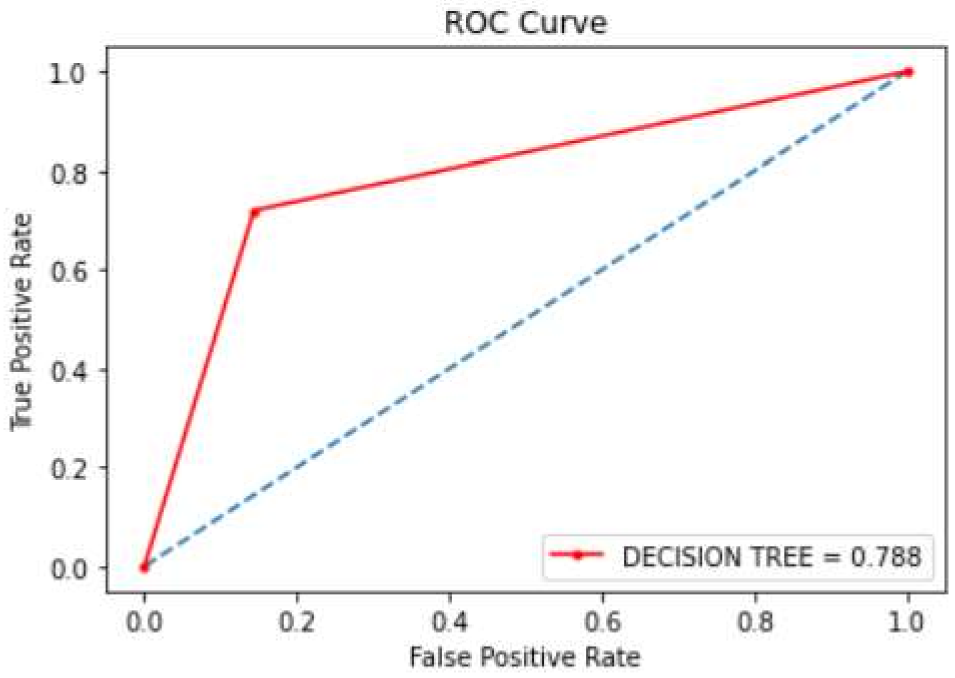
And accuracy after performing Cross Validation is as below.

```
Accuracy with DECISION TREE and K-FOLD CROSS VALIDATION: 0.69 (+/- 0.26)
```

So, from the results we can say Cross Validation didn't help in improving the model.



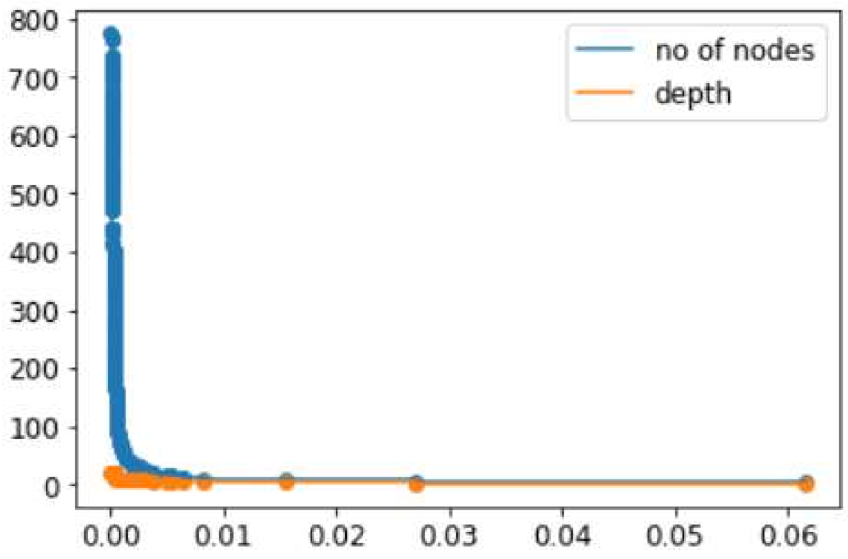
This graph is about the testing and training accuracy with change in depth of decision tree. As depth increases, training accuracy is increases. But in test, accuracy is increased till 7 depths, and there after it is decreasing. So optimum depth will be 7.



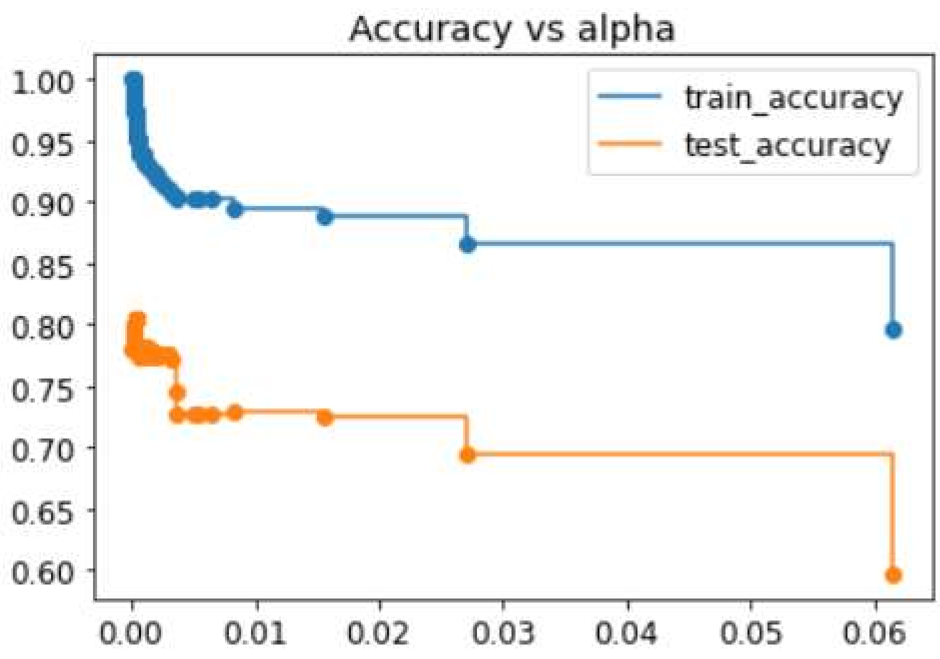
Above is the ROC curve for decision curve, and ROC AUC value is 0.788

Compared between SVM and Decision Tree, ROC AUC value is higher for SVM model. So SVM is better model for prediction for the given data set.

Decision trees can easily overfit. One way to avoid it is to limit the growth of trees by setting constrains. But a most effective way is to use post pruning methods like cost complexity pruning. This helps to improve test accuracy and get a better model.
Cost complexity pruning is all about finding the right parameter for alpha. We will get the alpha values for this tree and will check the accuracy with the pruned trees.

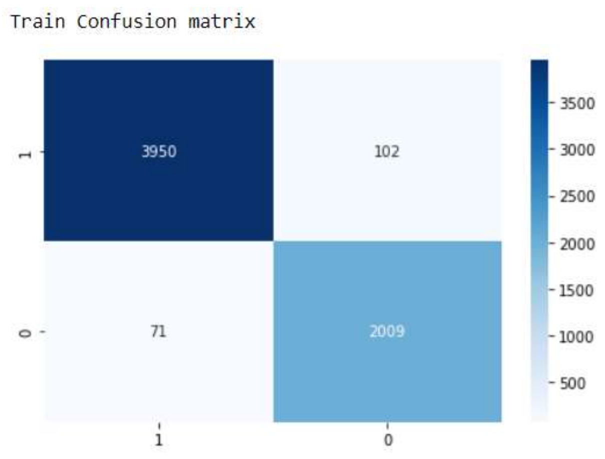


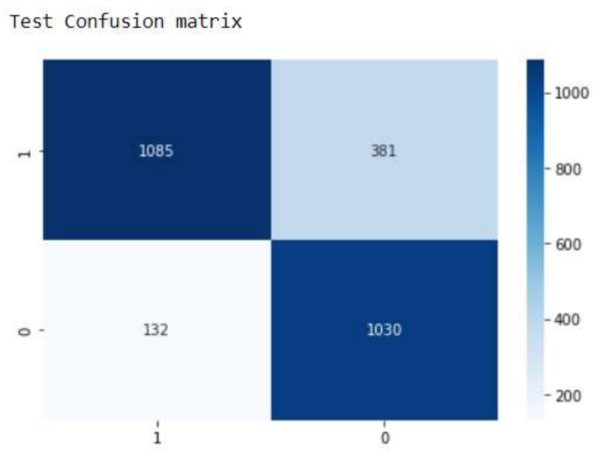
As alpha, number of nodes and depths decreases.



From the above graph, we are choosing alpha value as 0.003

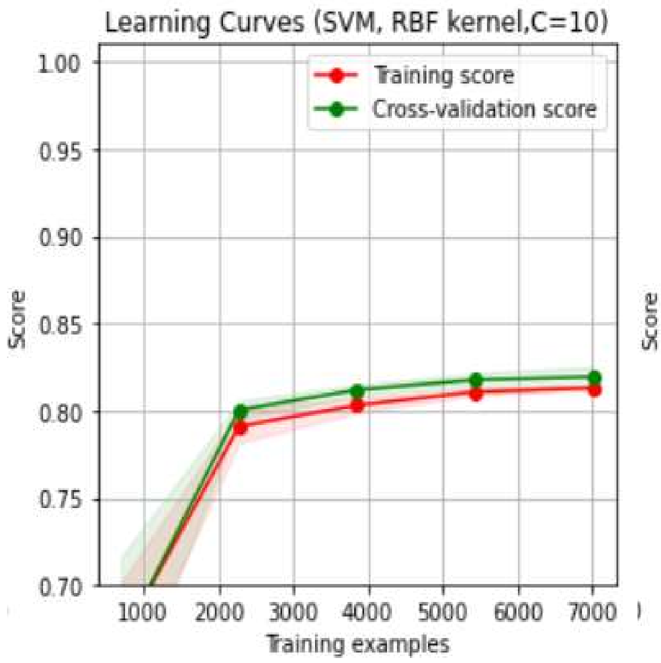
```
Test score 0.8047945205479452
```

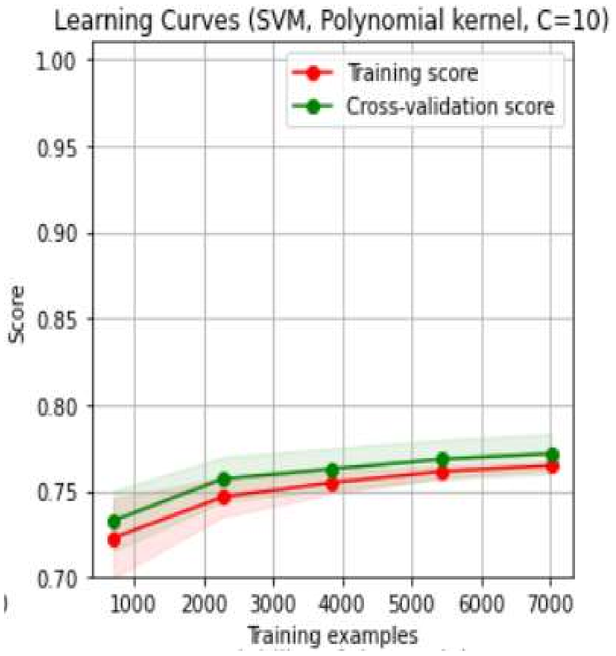


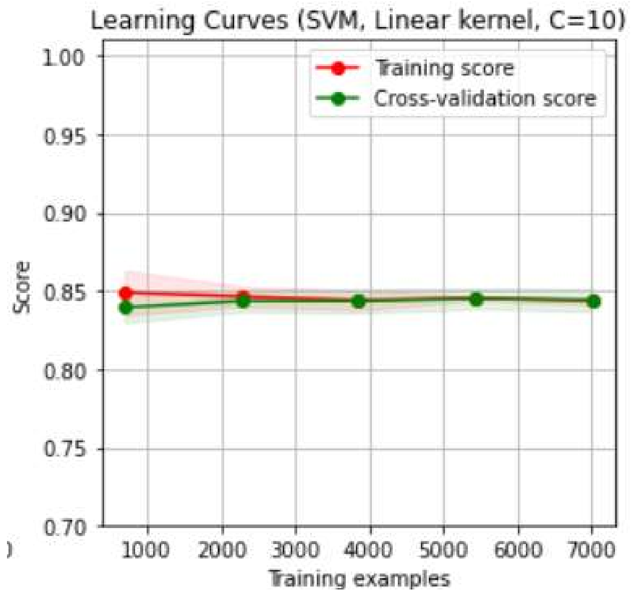


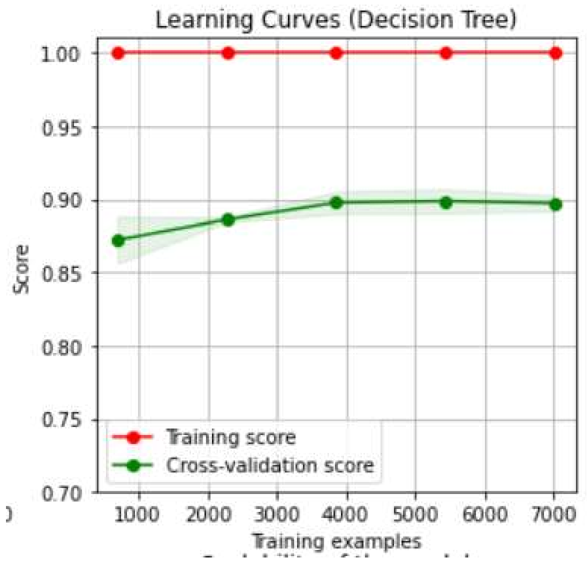
After pruning we can say that our model is not overfitting and test performance has improved.

Learning Curves for different models:









From the learning curves, we can observe that for RBF and polynomial kernel, as number of training examples increase training score, and cross validation score increases. Both are similar in the trend. For linear kernel, training and cross validation score is almost the same for all training examples. For decision tree, cross validation score is slightly increased with increase in training examples.