ASSIGNMENT 2 REPORT – MANIKANDAN LALITPRASAD (mxl190001)

Dataset- SGEMM GPU Kernel Performance

This data set measures the running time of different parameter combinations of parameterizable SGEMM GPU kernel. The dataset contains 241600 records of these feasible combinations. For each tested combination, 4 runs were performed and recorded (measured in milliseconds)

There are 14 parameters, the first 10 are ordinal and can only take up to 4 different powers of two values, and the 4 last variables are binary.

Attribute Information:

Independent variables:
1-2. MWG, NWG: per-matrix 2D tiling at workgroup level: {16, 32, 64, 128} (integer)
3. KWG: inner dimension of 2D tiling at workgroup level: {16, 32} (integer)
4-5. MDIMC, NDIMC: local workgroup size: {8, 16, 32} (integer)
6-7. MDIMA, NDIMB: local memory shape: {8, 16, 32} (integer)
8. KWI: kernel loop unrolling factor: {2, 8} (integer)
9-10. VWM, VWN: per-matrix vector widths for loading and storing: {1, 2, 4, 8} (integer)
11-12. STRM, STRN: enable stride for accessing off-chip memory within a single thread: {0, 1} (categorical)
13-14. SA, SB: per-matrix manual caching of the 2D workgroup tile: {0, 1} (categorical)

Output:
15-18. Run1, Run2, Run3, Run4: performance times in milliseconds for 4 independent runs using the same parameters. They range between 13.25 and 3397.08.
We take the average of these four runs as our target variable and added as a column in our data frame

OBJECTIVE:
To implement three learning algorithms using this dataset
1) Support Vector Machines
2) Decision Trees
3) Boosting

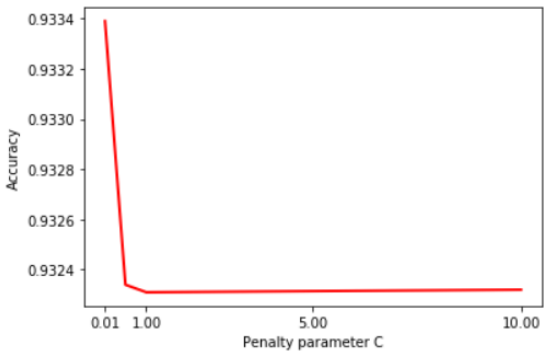
Algorithm 1 – Support Vector Machines (SVM):

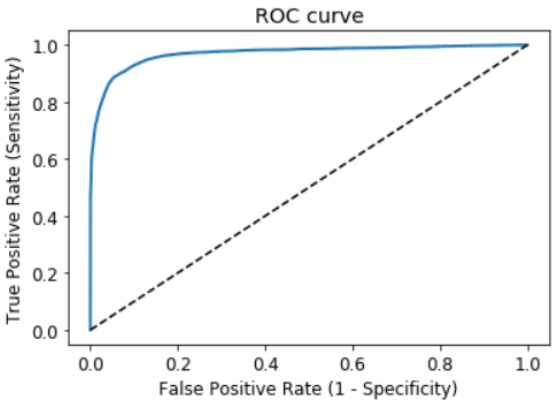
SVM is a very powerful algorithm for classification, which models linearly separable data well and also very good for modeling data that is not linearly separable
For this algorithm I have chosen to use 'sklearn' package from python for ease in changing Kernels – Linear, Polynomial and Radial Basis Function Kernel.

Linear Kernel:

Linear Kernel is used when the data is Linearly separable, that is, it can be separated using a single Line.

C is the hyperparameter which gives a penalty and tell the SVM the amount to avoid misclassification from each training example. For large values of C, the optimization will choose a smaller-margin hyperplane and for a very small value of C will cause the optimizer to look for a larger-margin separating hyperplane. For very small values of C, it is more likely that you will get a misclassified example. To choose the right value of C, 5-Fold Cross validation method is used. Different C values have been used to find the one with the best accuracy and it turns out that for C=0.01 we get the highest accuracy of **93.33%** for this model with linear kernel. The ROC curve and AUC for this model is shown on the right





Confusion matrix:

[[51734 1974]

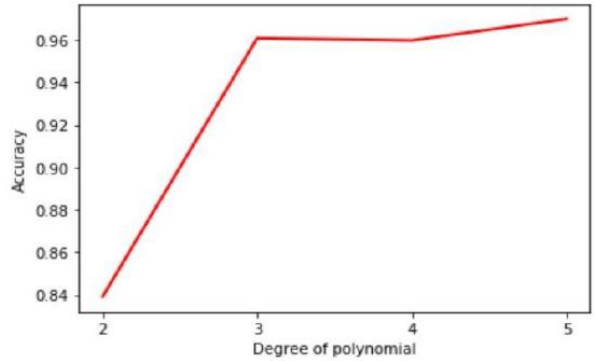
[2915 15857]]

AUC – 0.96947

Polynomial Kernel:

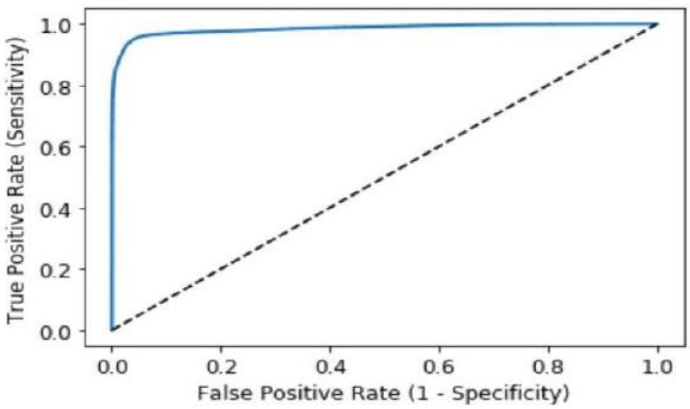
We need to choose a value of degree for Polynomial kernel. As the degree of polynomial increases, the model tends to overfit and hence the value of test error increases to a large extent. We then use 5-fold cross validation with different values of degrees such as 2,3,4 and 5 and then run the model again. We can clearly see that degree=5 gives the best accuracy with 96.97%. On the right is the confusion matrix and the ROC curve when we run the model with degree=5. We get an accuracy of **96.13%**

AUC – 0.9849



[[52720 988]

[1815 16957]]

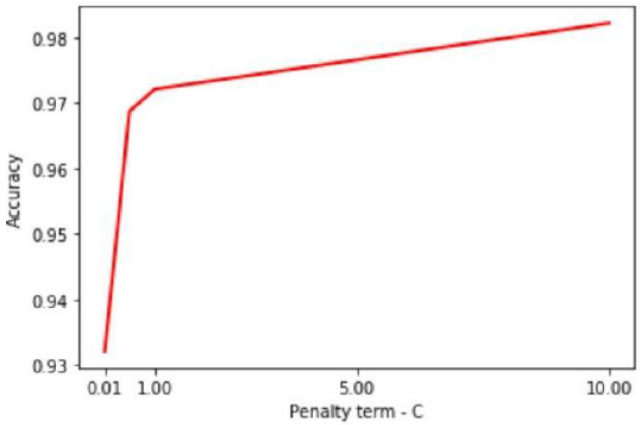


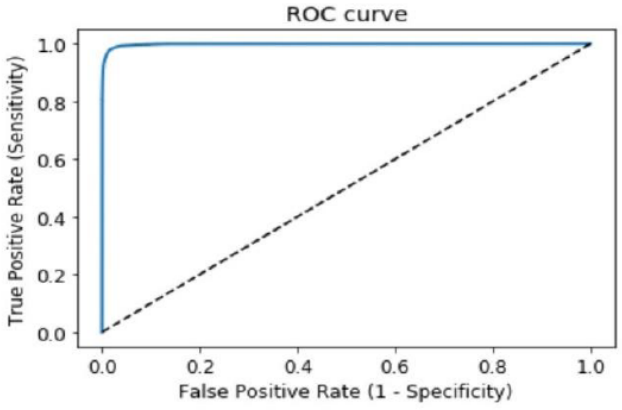
Radial Basis Function Kernel:

We need to choose the right value of C for RBF Kernel. We need to choose a value such that the model does not underfit or overfit. We experiment with different values of C such as 0.001, 0.5, 1 and 10. 5-fold cross validation method is used to choose the right value

of C for this model. We can clearly see that c=10 gives the highest accuracy of 98.21% and hence we use c=10 for testing the dataset. Below is the confusion matrix for the test dataset and the ROC curve. This gives a high accuracy of 98%.

Confusion Matrix:
```
[[53126   582]
 [  633 18139]]
```
AUC: 0.99845

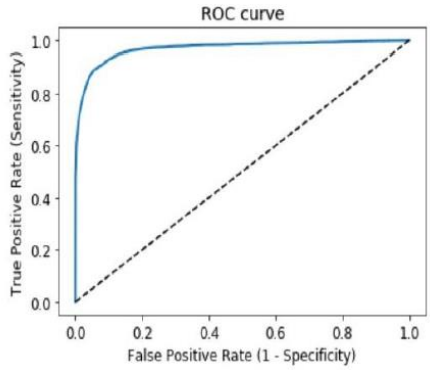




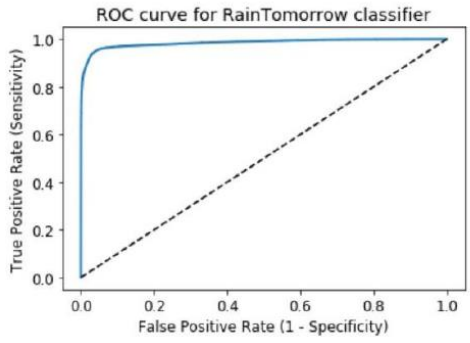
0.99845

Comparison between Kernels:

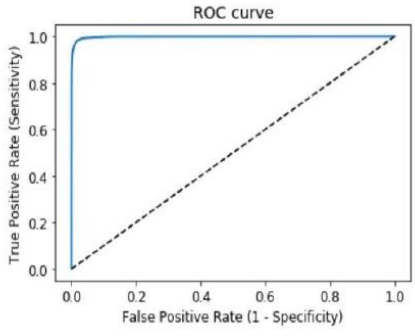
When comparing between linear, polynomial and RBF kernels, we observe that the test accuracies are **93.33%, 96.13% and 98%** respectively. Hence, we can conclude that **RBF kernel** gives the highest accuracy for this dataset. While observing the ROC curve, it is seen that the curve is closer to the top left for RBF kernel and has the highest AUC of 0.99845.



0.96947



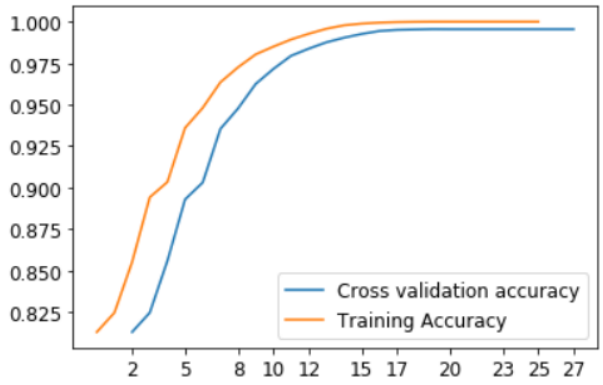
0.9849



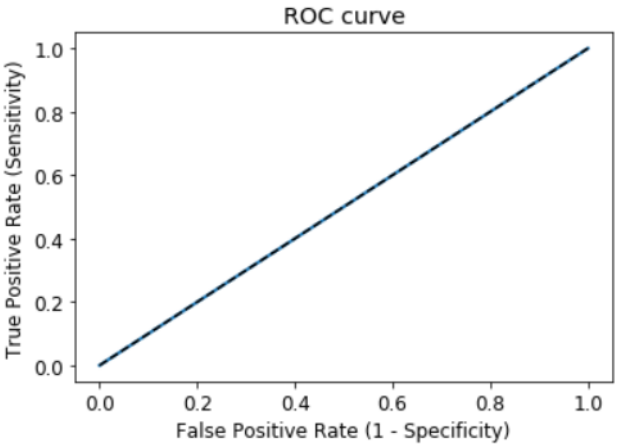
0.99845

Algorithm 2 – Decision Trees:

Information gain represents how effective an attribute is, in classifying the data. Information gain is the reduction in entropy, which refers to impurity in examples. Information gain computes the difference between entropy before and after the split. The Imbalances in the data set will be dealt with by doing this. We get a training error of 99.62% and a testing error of 99.59%.



We need to prune the model to get a better fit. Hence we use cross validation and experiment with 10 different depth values between 2 to 27. There is underfitting in this model. The model converges at a depth of 15 to be constant. Hence, we fix the depth of 15 for pruning. The accuracy of this model is 81.29%

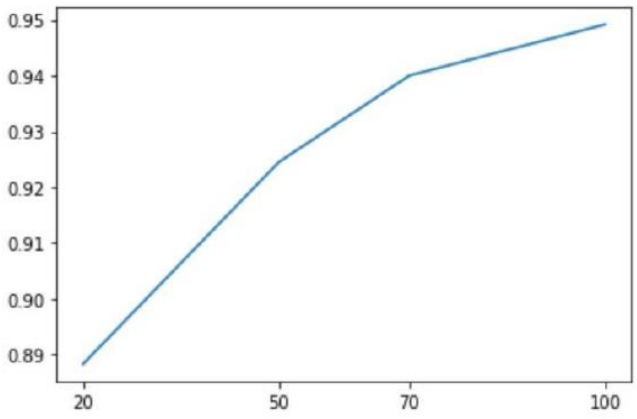



```
Confusion Matrix:

[[51352  2356]
  [ 2297 16475]]

AUC – 0.5
```

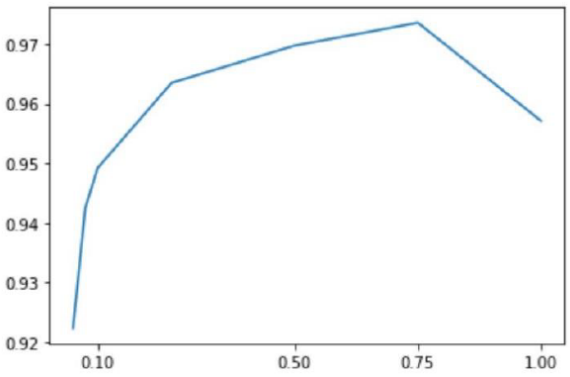
Algorithm 3: Boosted Decision Tree – Gradient Descent:

Gradient Descent will be used for boosted version of Decision Tree. It uses gradient descent algorithm which can optimize any differentiable loss function. Gradient Boosting is a combination of Gradient Descent and Boosting algorithm.

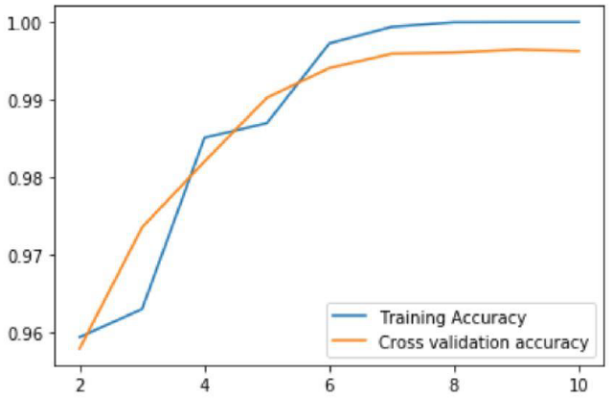
First, the number of boosting stages have to be selected. Hence, we experiment with 20, 50, 70 and 100 and perform cross validation. We observe that n_est = 100 gives the best accuracy of 95%.



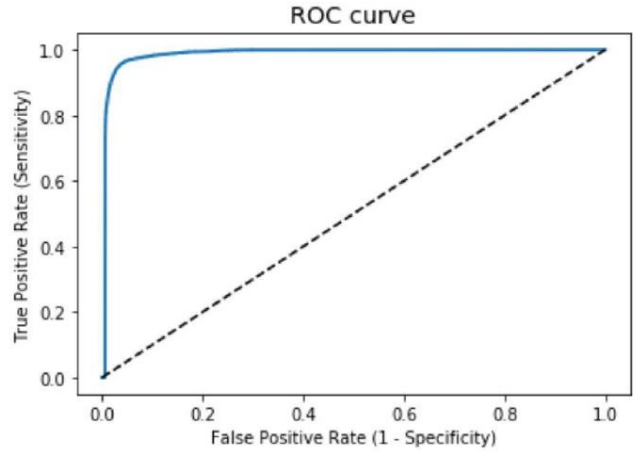
.



Next, we need to choose the learning rate and hence we experiment with values of 0.05, 0.075, 0.1, 0.25, 0.5, 0.75, 1 and we find that alpha=0.75 gives the highest accuracy at 97.35%



Finally, we experiment with various values of depth from 2 to 10. We can see a case of overfitting as after depth=3 the training accuracy starts to overshoot and reaches 100% accuracy whereas the CV accuracy reaches a constant. Hence, we choose 3 as the optimal depth value which gives 96.32%. We finally train the model with n_est=100, alpha=0.75 and depth=3. We get accuracy of 96.07%. Below is the confusion matrix, classification metrics and the ROC curve.

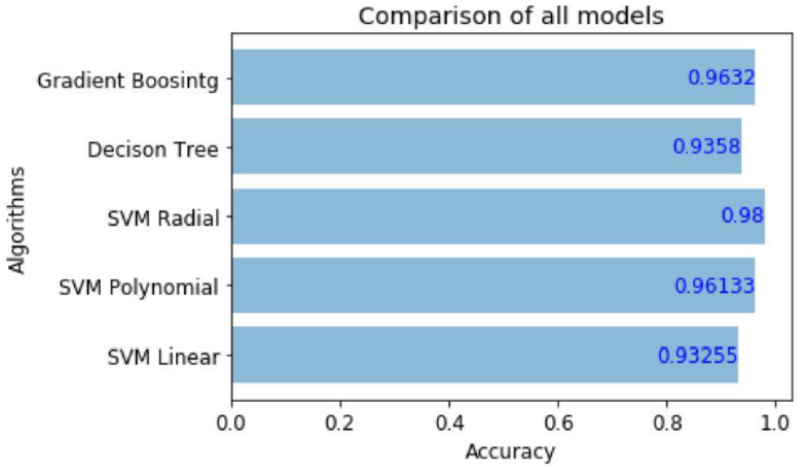


Confusion Matrix: [[52512 1196]
[1471 17301]]

AUC – 0.98828

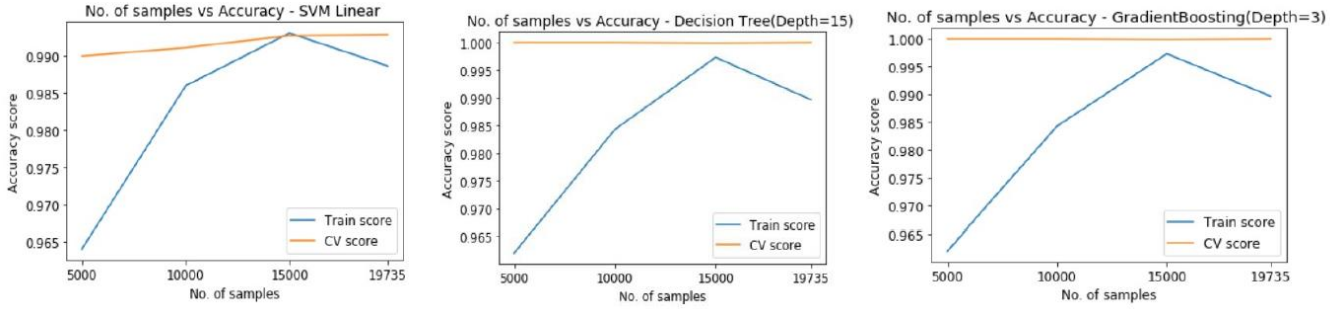
	precision	recall	f1-score	support
0	0.97	0.98	0.98	53708
1	0.94	0.92	0.93	18772
accuracy			0.96	72480
macro avg	0.95	0.95	0.95	72480
weighted avg	0.96	0.96	0.96	72480

<u>Comparison across algorithms:</u>



The test accuracies and ROC curves for all the models have been plotted and has been compared with each other. We can see that both SVM, RBF kernel and Gradient Boosting Algorithm perform well on this dataset. SVM RBF Kernel gives 98% and Gradient Boosting gives 96.07%. The AUC scores also justify the same with SVM RBF giving 0.99 and Gradient Boosting gives 0.98. Decision tree Algorithm is the preferred not to be used because of the low AUC score of 0.5.

Learning curve as a function of Training size:



From the above graphs we can observe that as the number of samples increase, the train score peaks and then starts to dip after a certain point. Around 15,000 samples the train score seems to reach its peak.

Conclusion:

From the Information present above, we can conclude that SVM RBF Kernel is the best algorithm which can be used in this dataset with an accuracy of 98%.

Dataset 2 – Rain in Australia

This dataset contains daily weather observations from numerous Australian weather stations. This dataset contains about 10 years of daily weather observations from numerous Australian weather stations.
The dataset is interesting because it involves many features and provides us with various variables such as Wind speed, humidity, temperature, pressure etc and it will be challenging to work with this dataset. Occurrence of rain is indicated by 1 and non-occurrence is indicated by 0. I have chosen all the continuous variables for my analysis.

Attribute Information:

Independent Variables:
Mintemp, Maxtemp, Rainfall, WindGustSpeed, WindSpeed9am, WindSpeed3pm, Humidity9am, Humidity3pm, Pressure9am, Pressure3pm, Cloud9am, Cloud3pm, Temp9am, Temp3pm
I did random sampling of 20000 observations because the original sample contains 1.4 lakhs. 20000 is a good sample for training the model. Mean normalization is done on the selected features and the data set divided into 70 and 30.

OBJECTIVE-

To implement three learning algorithms using this dataset
1) Support Vector Machines
2) Decision Trees
3) Boosting

Target Variable:

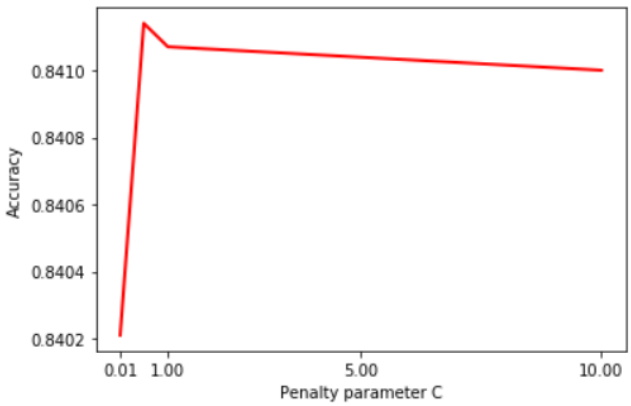
The target variable RainTomorrow means: Did it rain the next day? Yes or No.

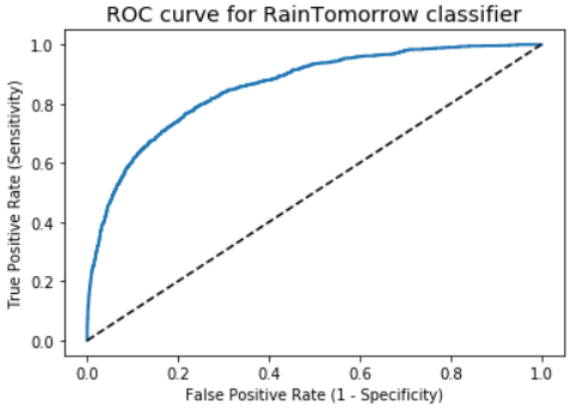
TASKS PERFORMED:

Algorithm 1 - Support Vector Machine (SVM):

Linear kernel:

We use 5-fold cross validation to choose a good value of C. I experiment with values of 0.01,0.5,1 and 10. In the graph, we can see that C=0.5 gives the highest accuracy of 84.13%. Thus, C=0.5 is chosen to test the model. We get an accuracy of **84.343%** for this model with linear kernel.





The image shows the Roc curve and AUC for this model.

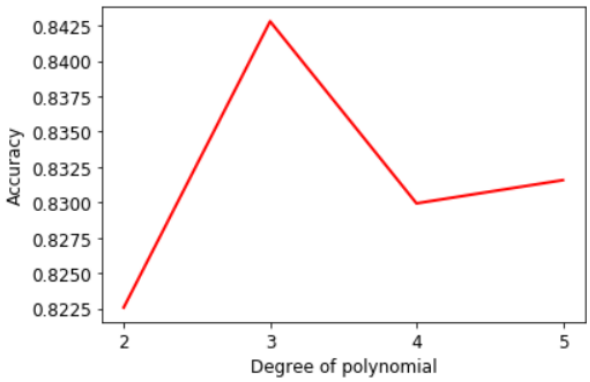
The confusion matrix when C=0.5 is -

```
[[4522  191]
 [ 746  541]]
```

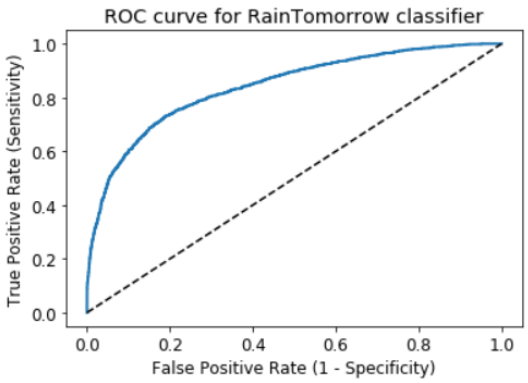
AUC - 0.85673

Polynomial kernel:

We run the model with different values of degree using 5-fold cross validation with degrees 2,3,4 and 5. We can clearly see that degree=3 gives the best accuracy with 84.28%. On the right is the confusion matrix when we run the model with degree=3. We get an accuracy of **83.88%.**

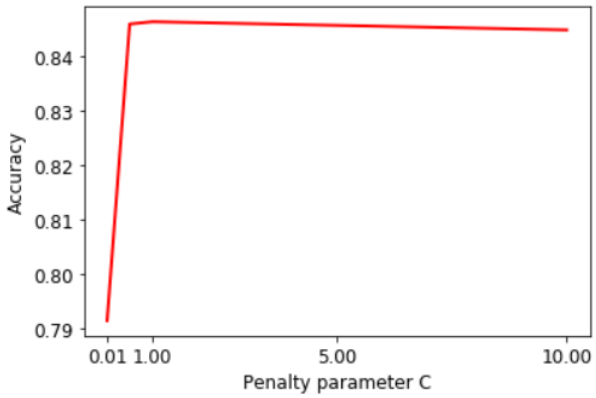


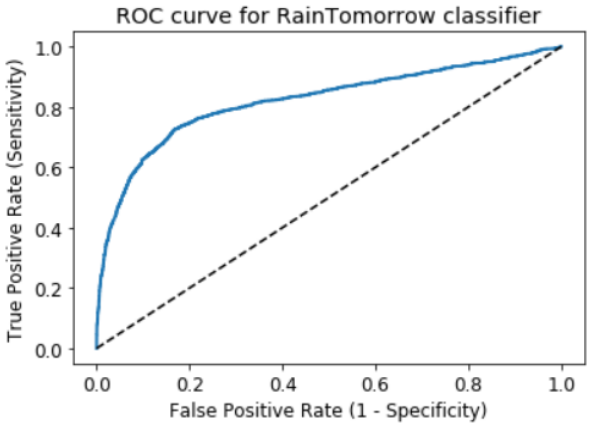
AUC – 0.84082
[[4592 121] [846 441]]



Radial basis function kernel:

We need to choose a good value of C so that the model does not overfit or underfit. We experiment with values of 0.01,0.5,1 and 10 using 5-fold cross validation. The accuracy keeps increasing as the C value is increased.
We can see that c=0.5 gives the highest accuracy of 84.58% and hence we use c=0.5 for testing the dataset.



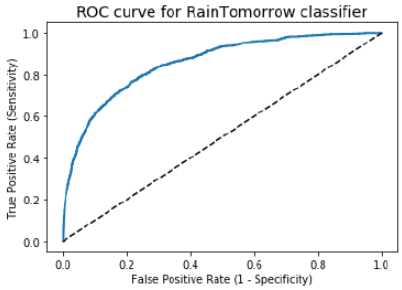


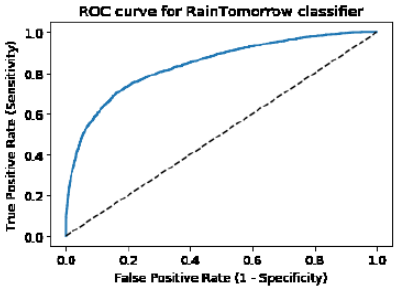
The image shows the confusion matrix for the test dataset and the ROC curve. This gives a high accuracy of **84.8%.**

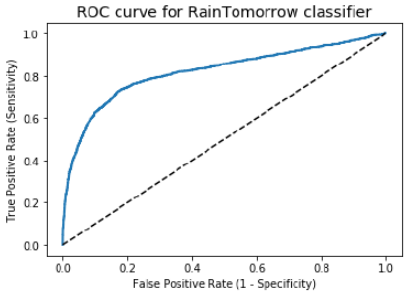
[[4546 167]
[746 541]]

AUC- 0.82043

Comparison Between Kernels:

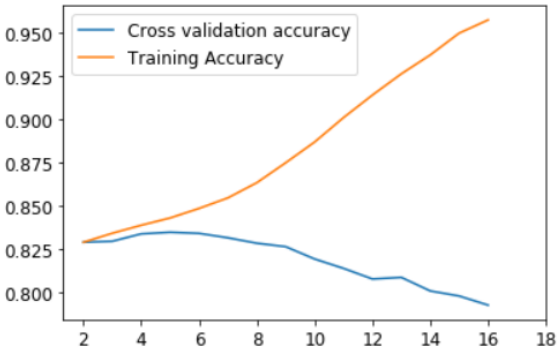






Among linear, polynomial and radial basis kernels, the test accuracies are 84.38%, 83.88% and 84.8% respectively. The linear and radial kernel both gives good accuracies for this dataset, but ROC curves is closer to the top left for linear kernel and has the highest AUC of 0.857. Hence, we choose **Linear kernel** as the best of all kernels for this dataset.

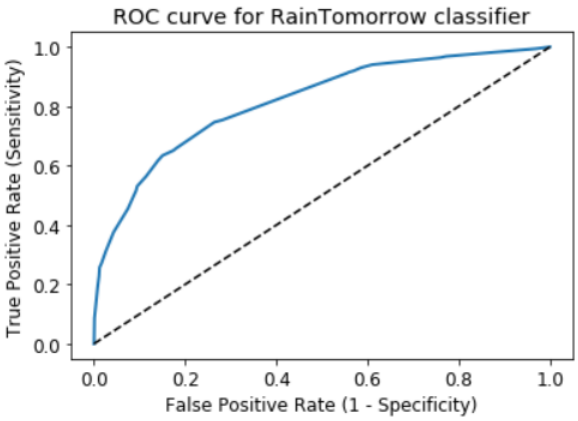
Algorithm 2- DECISION TREE



with various depths from 2 to 16 to find the best fit.

When we run the decision tree, we get
Train Error - 78.17%
Test Error - 78.21%

For Pruning the model to find a better fit for the model, we use cross validation and experiment



The graph is plotted while pruning the model shows that this is a case of overfitting. After depth=5, the training accuracy begins to shoot upwards while the cross-validation accuracy is reaching a level of constancy. This means the train error is low while the test error is very high. Hence, we finally fix upon depth=5 as a good parameter for pruning. For Depth=5, we get an accuracy of 83.46% as train error and **83.23%** as test accuracy, the image shows the confusion matrix for the model followed by the classification metrics and ROC curve

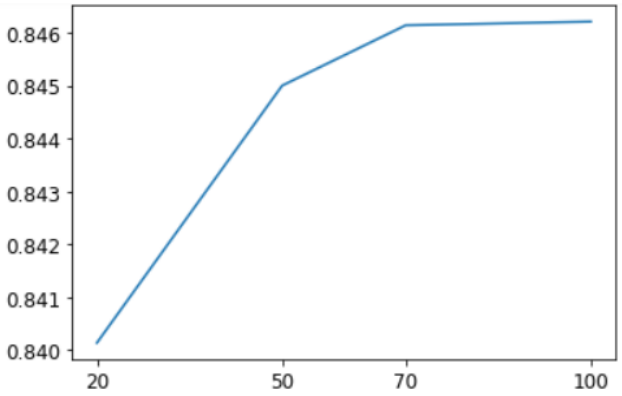
[[4510 203]
[803 484]]
AUC- 0.81558

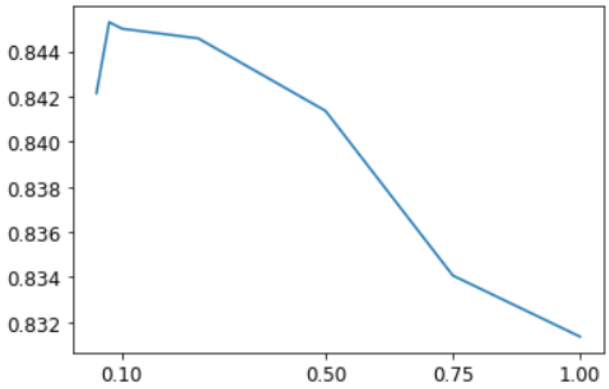
	precision	recall	f1-score	support
0	0.85	0.96	0.90	4713
1	0.70	0.38	0.49	1287
micro avg	0.83	0.83	0.83	6000
macro avg	0.78	0.67	0.70	6000
weighted avg	0.82	0.83	0.81	6000

Algorithm 3: Boosted Decision Tree- Gradient Descent:

For Boosted Decision tree, we choose the number of boosting stages. We experiment with 20, 50, 70 and 100 as the different values and perform cross validation.

We see that n_estimators=50 gives 84.5% and almost reaches a constant after that. Hence, we fix n_estimators=50.

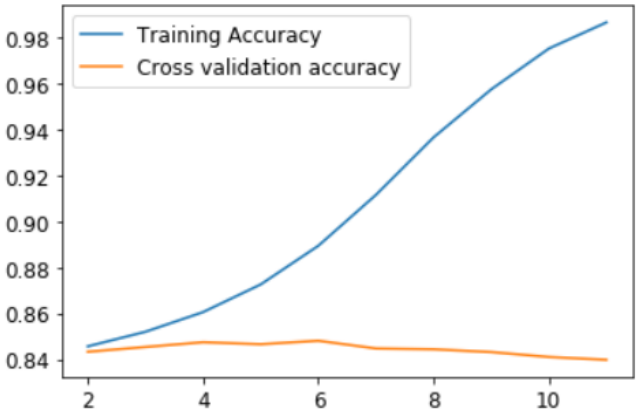




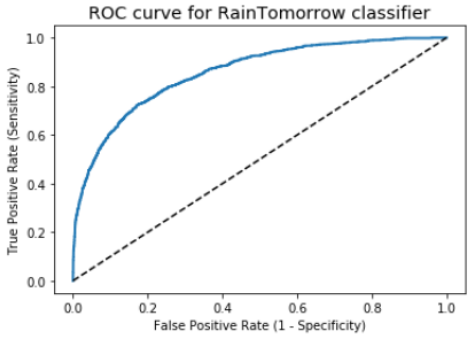
The learning rate is experimented with values of 0.05, 0.075, 0.1, 0.25, 0.5, 0.75, 1 and when alpha=0.075 gives the highest accuracy at 84.53%.

We experiment with various values of depth from 2 to 10. We can see a case of overfitting as after depth =4 the training accuracy starts to overshoot and reaches 100% accuracy whereas the CV accuracy reaches a constant. Hence, we choose 4 as the optimal depth value which gives 84.74%.

Model is trained with n_estimators=50, alpha=0.075 and depth=4.
Accuracy - **84.82%.**
The confusion matrix and classification metrics and the ROC curve are as follows:



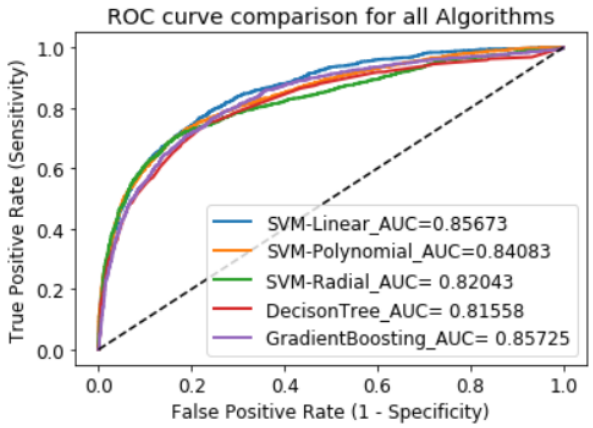
[[4525 188]
[723 564]] **AUC- 0.85725**



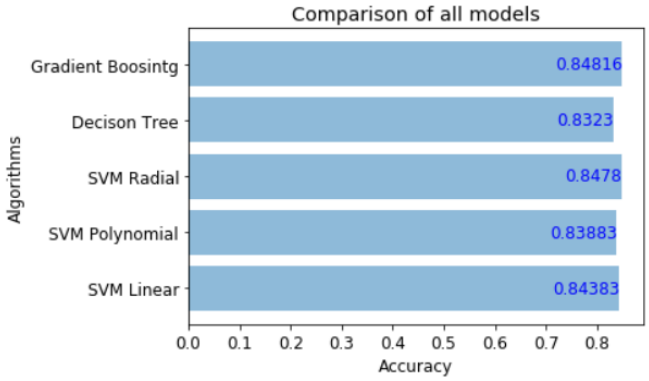
	precision	recall	f1-score	support
0	0.86	0.96	0.91	4713
1	0.75	0.44	0.55	1287
micro avg	0.85	0.85	0.85	6000
macro avg	0.81	0.70	0.73	6000
weighted avg	0.84	0.85	0.83	6000

Comparison across Algorithms:

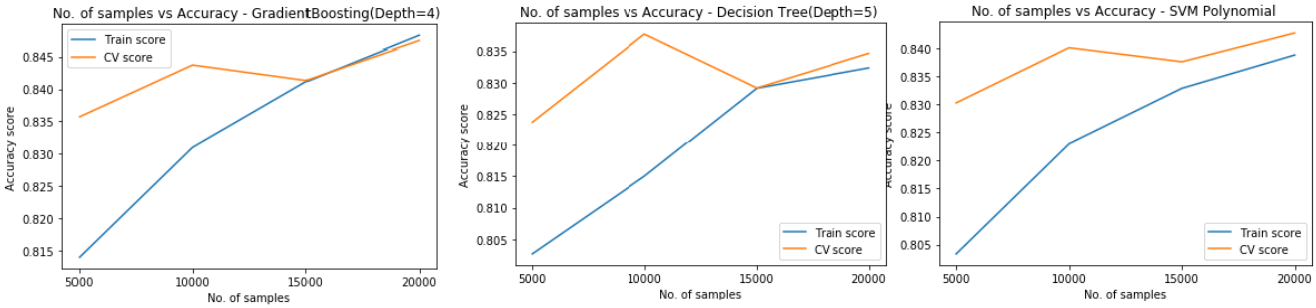
For this dataset, I have plotted the test accuracies and ROC curves for all the models below. We can see that almost all the algorithms perform well on this dataset. SVM Linear gives the best among SVM kernels. Decision Tree though has a good accuracy rate has the lowest AUC score of all.
Gradient Boosting with 84.82% and AUC score of 0.85725 performs best on this dataset. The reason is that this algorithm learns from the weak classifiers and improvises each stage and hence gives a better accuracy.



Decision Tree performs worst for this data.



Learning curve as a function of Train size:



From the above graphs we can see that as the training size increases, the training accuracy keeps increasing while the CV accuracy increases and then drops again before increasing again. Only when the size is at least 15000, the training and CV accuracy have a lesser difference and almost reach same levels.
Hence, we conclude that at least 15000 points would be good for this data.

Conclusion:
We conclude that **Gradient Boosting** does best on this dataset with 84.82% accuracy.

THINGS THAT COULD HAVE BEEN DONE:

- Feature selection could have been done by using forward and backward selection. Regularization could have been done to avoid overfitting.
- GridSearchCV in Scikit-learn package could have been used to do hyperparameter tuning to select optimum values of C and gamma.

EFFECT OF CROSS VALIDATION:
Cross validation was really helpful to avoid overfitting and underfitting. For this assignment, I used CV for choosing an optimal value of C, degree of polynomial, for pruning (choosing depth) and for choosing hyperparameters of Gradient Boosting. This really helped because training the model with only training set and not using validation leads to models with high variance or high bias. Cross validation rectified this issue and as a result I obtained good models with a good bias-variance trade-off.