Rainfall prediction using machine learning technique (SVM): A case study on Cobar and Coffs Harbour, Australia

Mrinmay Date 18BIS0147
SENSE School
Vellore Institute of Technology, Vellore
mrinmay.mrityunjay2018@vitstudent.ac.in

Kartik Agrawal 18BIS0135
SENSE School
Vellore Institute of Technology, Vellore
kartik.agrawal2018@vitstudent.ac.in

Abstract—*Rainfall prediction is important as heavy rainfall can lead to many disasters. The prediction helps people to take preventive measures and moreover, the prediction should be accurate. The main challenge is to build a model for long-term rainfall prediction. Heavy precipitation prediction could be a major drawback for the earth science department because it is closely associated with the economy and lifetime of a human. It's a cause for natural disasters like floods and drought that square measure encountered by individuals across the world each year. The prediction of rainfall will be done using machine learning techniques with SVM.*

Keywords—Rainfall predictions, machine learning, SVM

I. INTRODUCTION

Rainfall prediction is important as heavy rainfall can lead to many disasters. The prediction helps people to take preventive measures and moreover, the prediction should be accurate. The main challenge is to build a model for long-term rainfall prediction. Heavy precipitation prediction could be a major drawback for the earth science department because it is closely associated with the economy and lifetime of a human. It's a cause for natural disasters like floods and drought that square measure encountered by individuals across the world each year. The prediction of rainfall will be done using machine learning techniques with SVM. Australia's weather has changed significantly over the years with an increasing number of heat events and increasing severity of drought conditions due to below-average rainfall.

Among the top ten warmest years on record, eight have occurred since 2005.

In late December 2019 - early January 2020, Australia suffered from the most severe bushfires ever that were reported to have burnt over ten million hectares of land in southern regions.

Bushfires happen to owe to different factors. Strong winds, low humidity, and high temperatures altogether contribute to the higher frequency of fire weather days.

We can apply many techniques like classification, regression according to the requirements, and also we can calculate the error between the actual and prediction and also the accuracy.

Different techniques produce different accuracies so it is important to choose the right algorithm and model it according to the requirements.

II. LITERATURE REVIEW

Thirumalai, Chandrasegar, et al. [1] discuss the amount of rainfall in past years according to the crop seasons and predicts the rainfall for future years. The crop seasons are Rabi, Kharif, and Zaid. The linear regression method is applied for early prediction. Here, Rabi and Kharif were taken as variables if one variable was given then the other can be predicted using linear regression. Standard deviation and Mean were also calculated for future prediction of crop seasons. This implementation will be used for farmers to have an idea of which crop to harvest according to crop seasons.

Parmar, Aakash, Kinjal Mistree, and Mithila Sompura [2] discuss the different methods used for rainfall prediction for weather forecasting with their limitations. Various neural networks algorithm which is used for prediction are discussed with their steps in detail categorizes various approaches and algorithms used for rainfall prediction by various researchers in today's era.

Saroj K. Mishra and Bijaya K. Panigrahi [3] have used artificial intelligence techniques like Artificial Neural Network (ANN), Extreme Learning Machine (ELM), K nearest neighbor (KNN) are applied for the

prediction of the summer monsoon and post-monsoon rainfall.

Singh, Gurpreet, and Deepak Kumar[4] states that there are many machine learning algorithms applied for the prediction of rainfall, and in this, they have used a hybrid approach that is combining two techniques, Random Forest and Gradient boosting with many machine learning techniques like ada boost, K-Nearest Neighbor(KNN), Support vector machine(SVM), and Neural Network(NN).

III. Methods and Datasets

Rainfall forecasting is very important because heavy and irregular rainfall can have many impacts like the destruction of crops and farms, damage of property so a better forecasting model is essential for an early warning that can minimize risks to life and property and also manage the agricultural farms in a better way. This prediction mainly helps farmers and also water resources can be utilized efficiently. Rainfall prediction is a challenging task and the results should be accurate. There are many hardware devices for predicting rainfall by using the weather conditions like temperature, humidity, pressure. These traditional methods cannot work in an efficient way so by using machine learning techniques we can produce accurate results. We can just do it by having the historical data analysis of rainfall and can predict the rainfall for future seasons. We can apply many techniques like classification, regression according to the requirements, and also we can calculate the error between the actual and prediction and also the accuracy. Different techniques produce different accuracies so it is important to choose the right algorithm and model it according to the requirements.

Here we will implement and compare three learning algorithms -

1) Support Vector Machines (Linear, polynomial, RBF kernels)
2) Decision Trees
3) Boosting

SVM

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

The equation of the separating hyperplane is given in Equation below:

$$w*X + b = 0$$

where Xi s the d-dimensional feature matrix consisting of features of classes to be separated, b is the bias, w is normal to the hyperplane, |b|/ ||w|| is the perpendicular distance from the hyperplane to the origin, and ||w||2 is the Euclidean norm of w.

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 that gives a penalty and tells 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, the 5-Fold Cross-validation method is used.

For linear kernel the equation for prediction for a new input using the dot product between the input (x) and each support vector (xi) is calculated as follows:

$$f(x) = B(0) + sum(ai * (x,xi))$$

This is an equation that involves calculating the inner products of a new input vector (x) with all support vectors in training data. The coefficients B0 and ai (for each input) must be estimated from the training data by the learning algorithm.

Polynomial Kernel

The polynomial kernel is a kernel function commonly used with support vector machines (SVMs) and other kernelized models, that represents the similarity of vectors (training samples) in a feature space over polynomials of the original variables, allowing learning of non-linear models.

The polynomial kernel can be written as

$$K(x,xi) = 1 + sum(x * xi)^d$$

and exponential as

$$K(x,xi) = exp(-gamma * sum((x — xi^2))$$

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.

Boosted Decision Tree – Gradient Descent

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

IV. Results and Analysis

Dataset: Rain in Australia (Kaggle)

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.

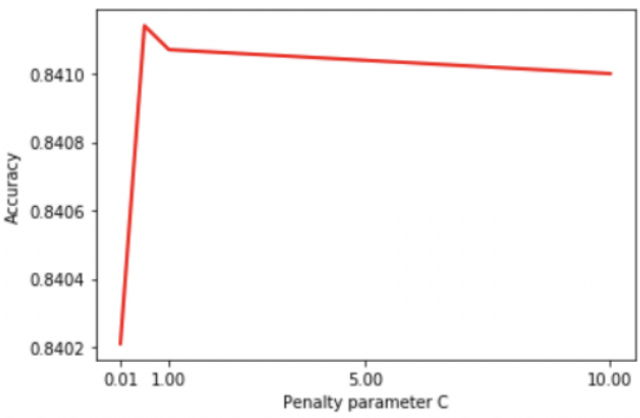
The occurrence of rain is indicated by 1 and nonoccurrence is indicated by 0. I have chosen all the continuous variables for my analysis.

Attribute Information:

Independent Variables: Mintemp, Maxtemp, Rainfall, WindGustSpeed, WindSpeed9am, WindSpeed3pm, Humidity9am, Humidity3pm, Pressure9am, Pressure3pm, Cloud9am, Cloud3pm, Temp9am, Temp3pm some 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 is divided into 70 and 30.

Algorithm 1: Support Vector Machine (SVM): Linear kernel:

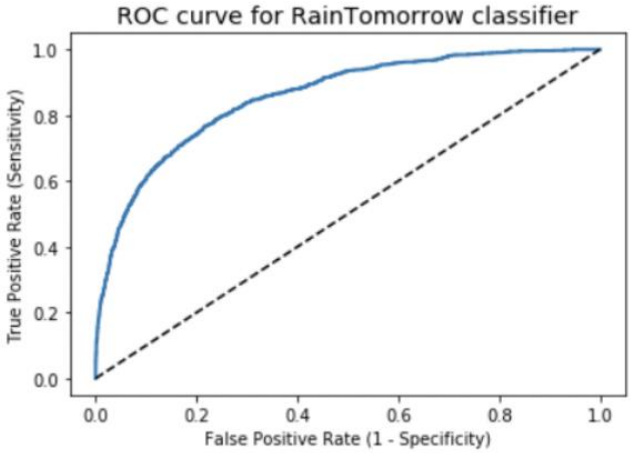
The 5-fold cross-validation to choose a good value of C. I experiment with values of 0.01,0.5,1 and 10. In the graph, we can see that C=0.5 gives the highest accuracy of 84.13%. Thus, C=0.5 is chosen to test the model. We get an accuracy of 84.343% for this model with a linear kernel. Figure 1 and 2 shows the Roc curve and AUC for this model. The confusion matrix when



C=0.5 is - [[4522 191]

[746 541]]

AUC - 0.85673



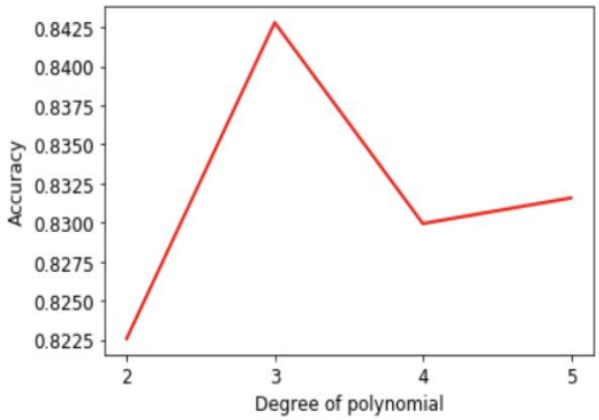
Polynomial kernel:

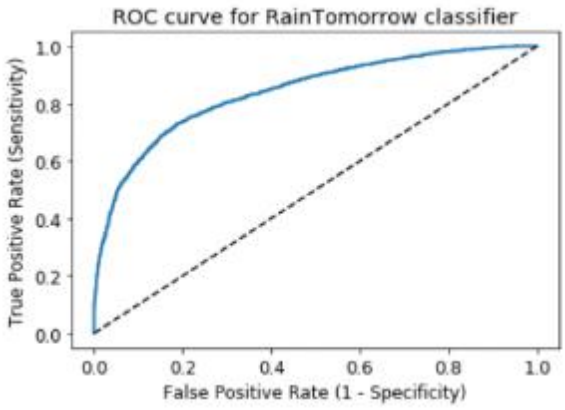
We run the model with different values of degree using 5-fold cross-validation with degrees 2,3,4 and 5. We can clearly see that degree=3 gives the best accuracy with 84.28%. On the right is the confusion matrix when we run the model with degree=3. We get an accuracy of 83.88%.

AUC – 0.84082

[[4592 121]

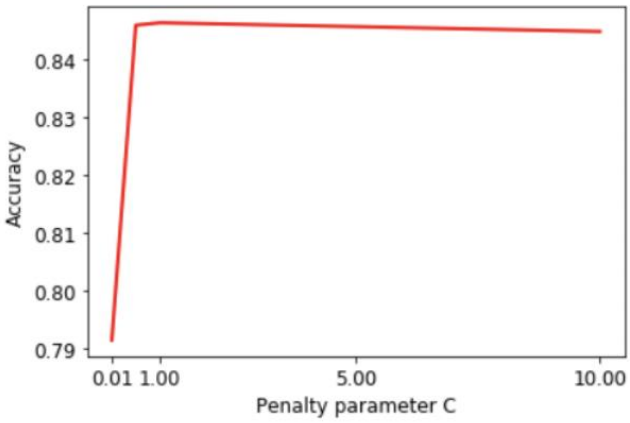
[846 441]]

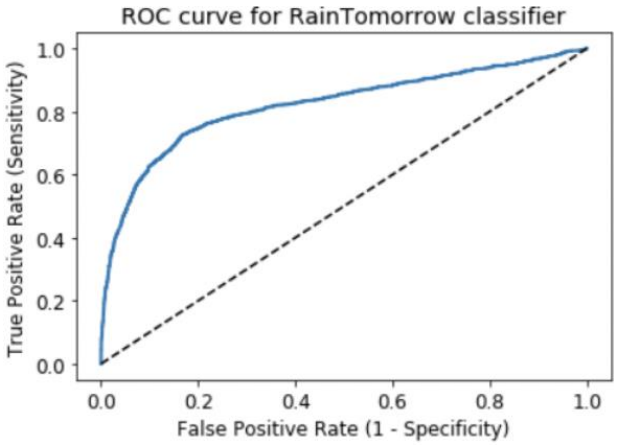




Radial Basis Function kernel:

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 high accuracy of 84.8%.

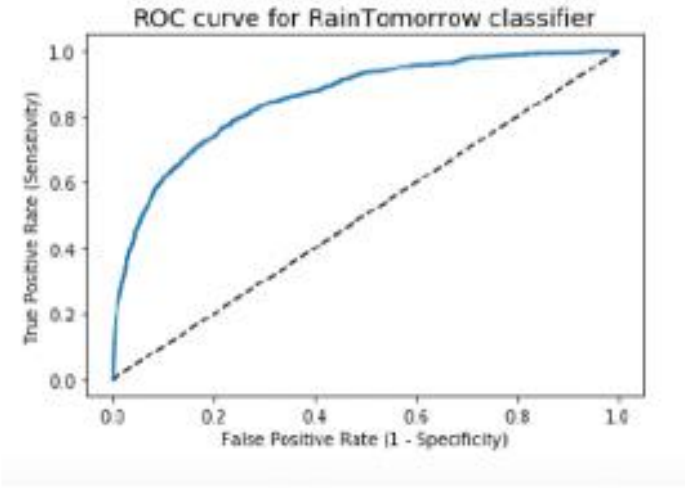
AUC- 0.82043

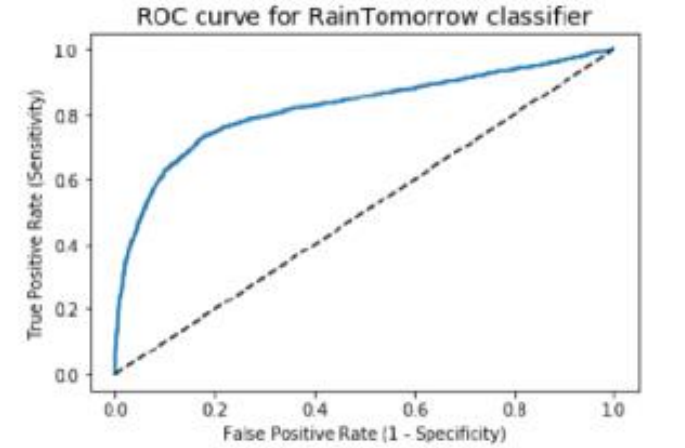
[[4546 167]

[746 541]]

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 are closer to the top left for the linear kernel and have the highest AUC of 0.857. Hence, we choose Linear kernel as the best of all kernels for this dataset





Algorithm 2- Decision Tree:

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

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.

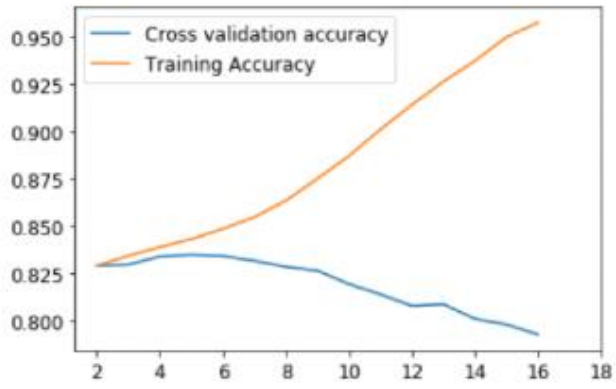
This means the training error is low while the test error is very high. Hence, we finally fix upon depth=5 as a good parameter for pruning.

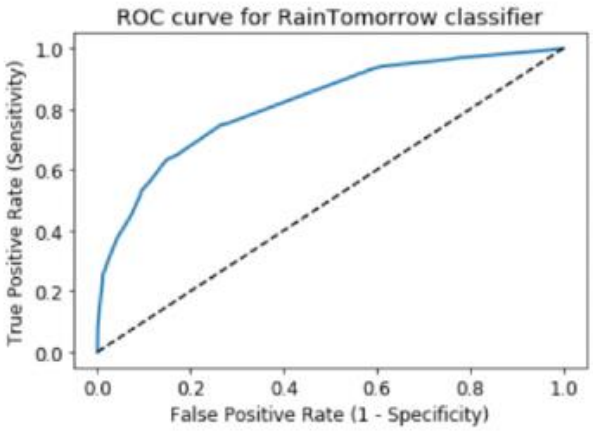
For Depth=5, we get an accuracy of 83.46% as train error and 83.23% as test accuracy, the image shows the confusion matrix for the model followed by the classification metrics and the ROC curve is

[[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.

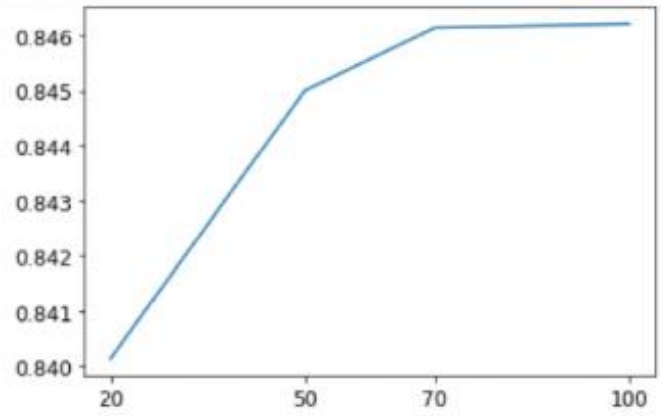
Hence, we fix n_estimators=50. The learning rate is experimented with values of 0.05, 0.075, 0.1, 0.25, 0.5, 0.75, 1 and when alpha=0.075 gives the highest accuracy at 84.53%.

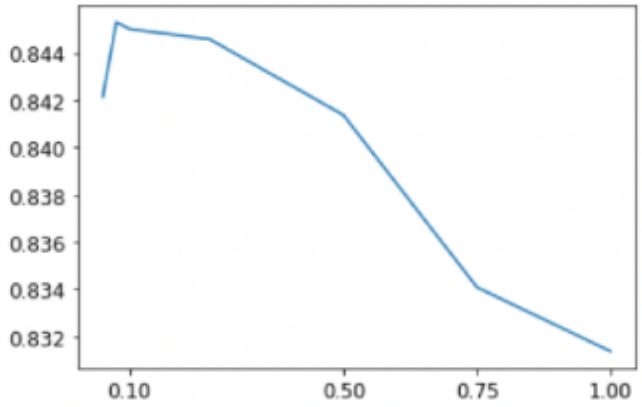
We experiment with various values of depth from 2 to 10. We can see a case of overfitting as after depth =4 the training accuracy starts to overshoot and reaches 100% accuracy whereas the CV accuracy reaches a constant.

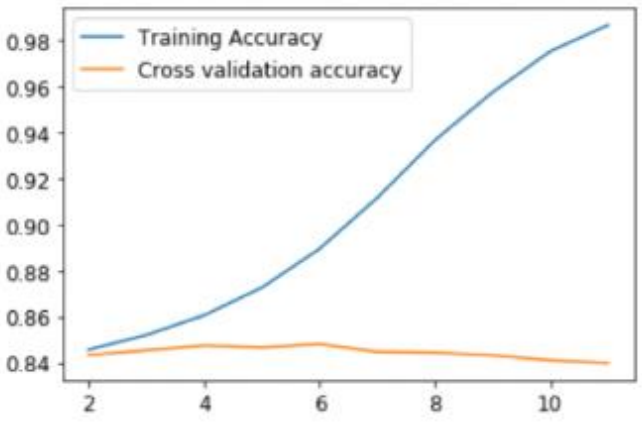
Hence, we choose 4 as the optimal depth value which gives 84.74%. Model is trained with n_estimators=50, alpha=0.075 and depth=4. Accuracy - 84.82%.

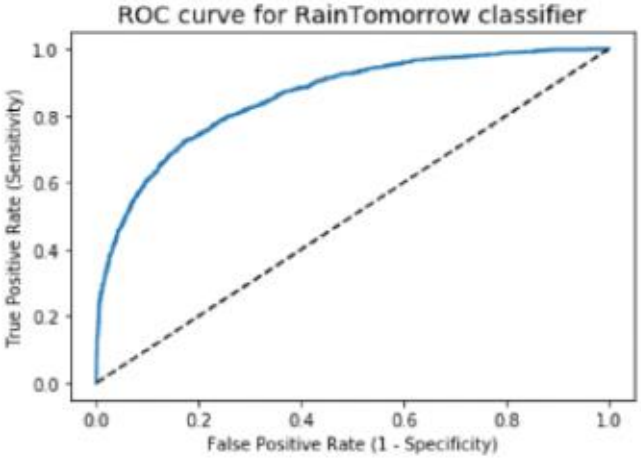
The confusion matrix and classification metrics and the ROC curve are as follows:

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



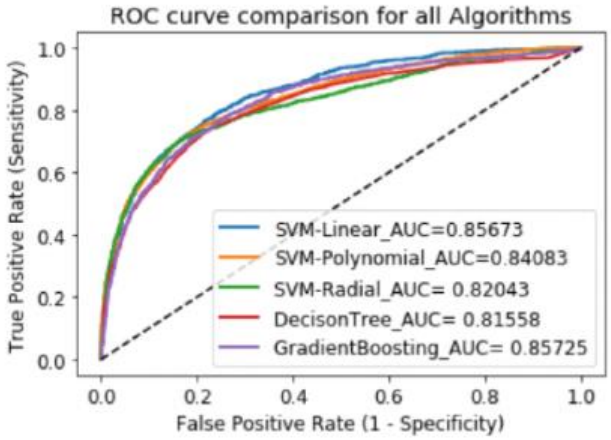




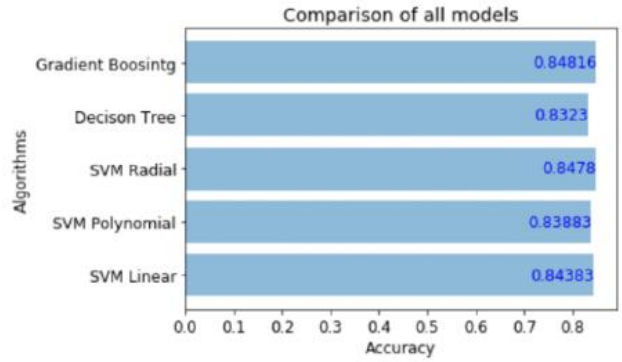


V. Conclusion

For this dataset, we 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 better accuracy.



The decision tree performs worst for this data and Gradient boosting does best with an accuracy of 84.82%.

Cross-validation was really helpful to avoid overfitting and under-fitting. For this assignment, we used CV for choosing an optimal value of C, degree of the polynomial, for pruning (choosing depth), and for choosing hyperparameters of Gradient Boosting.

This really helped because training the model with the 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, obtained are good models with a good bias-variance trade-off.

Future areas of research

- 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.