Prepared by: Sai Lakshmi Nikitha Akarapu

ID: SXA210112 Date: 07/08/2022

BUAN 6341 Applied Machine Learning

ASSIGNMENT NO 2

Seoul Rental Bike prediction Part II

Executive Summary

- **Sensible feature selection and preparation improve prediction accuracy.**
- **Converted the data set to binary classification using a median of the output parameter.**
- **Experimented with the data set by implementing SVM and optimizing with kernel functions and tolerance.**
- **Optimizing parameters like depth and number of splits improves the Decision Tree classification accuracy (pruning).**
- **Experimented using different K-folds for checking model performance on new data**

Introduction

In this project, the objectives were to convert the data set to a binary classification problem by thresholding the output to a class label and implement Logistic, SVM, and Decision Tree with different hyperparameters for logistic, different kernel functions and parameters for SVM, and performing pruning techniques to prune the decision tree for choosing the best model through experimentation for classifying the data set.

About the Data

The dataset consists of 14 features and 8760 records. The data is the rented bike count captured at each hour and weather conditions (Temperature, Humidity, Windspeed, Visibility, Dewpoint, Solar radiation, Snowfall, Rainfall), along with date information. To remove the serial correlation with time, the hours variable is converted to dummies to predict.

Project Outline

The Project is outlined to have 4 parts:

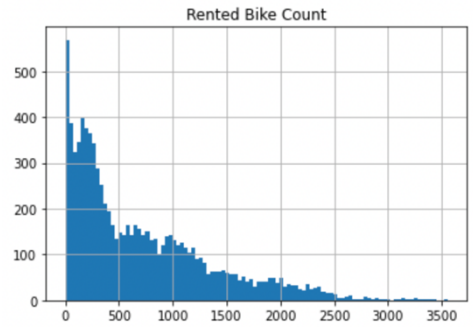
Part 1: Convert the data set from assignment 1 to a binary classification problem by thresholding the output to a class label and implement logistic regression and experiment with different hyperparameters.

Part 2: Implement SVM Classifier and experiment with different hyperparameters such as kernel and with different regularization parameters.

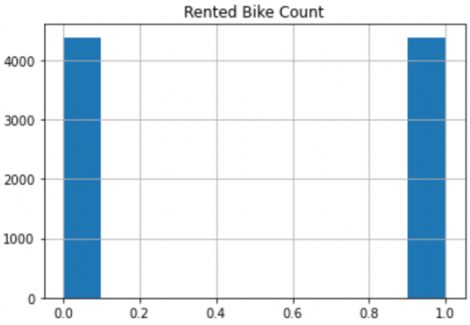
Part 3: Implement a Decision tree and experiment with different hyperparameters in pre-pruning and post-pruning techniques to prune the tree.

Part 4: Use the Cross-validation technique with optimized models of Logistic Regression, SVM Classifier, and Decision using the K-Fold technique to check the accuracies of the models to deploy in production.

Part 1:



From the histogram, we can observe that the Rented Bike Count data is right-skewed. The median is the best technique to separate the data set into two distributions. The median of the output feature is 504.5. The rented bike count is greater than the median value, considered class 1, and less than or equal to the value is class 0.

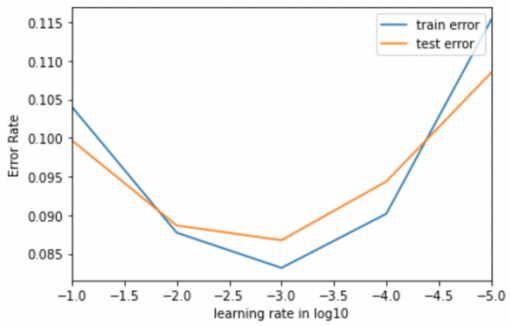


The picture represents the Rented Bike Count after the binary classification. We can observe all the data points under either class 0 or 1.

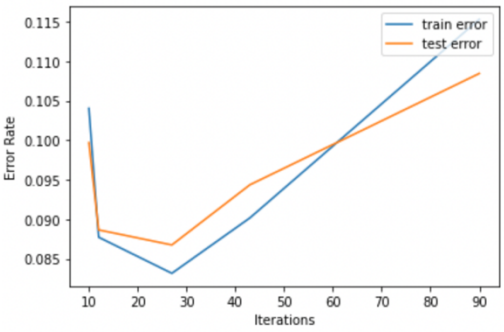
SGD Classifier:

In this project, we will use the SGD classifier as a logistic regressor.

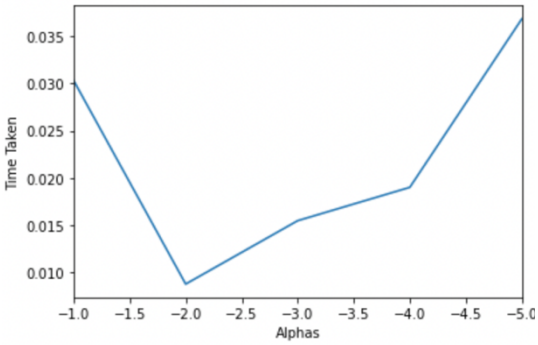
1. Experimented with learning rate and stopping criterion tolerance, the SGD model is optimized by keeping the threshold at 1e-3 and with learning rate alpha, below are the observations:



The experimentation shows that both train error and test error travel in the same direction through the learning rate range. The error rates for train and test decreased till 0.001 learning rate, and after that, the test error rate increased. From this, we can conclude that the train and test error rates are lower and better converging at an alpha value of 0.001.

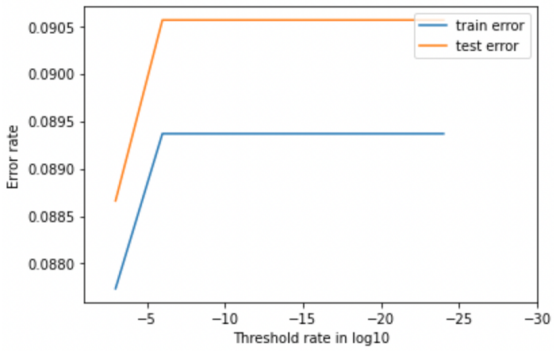


We can observe that the train error rate travels from ideal for underfitting or biased with an increase of iterations. In contrast, the test error rate travels in ideal situations. At around 25 iterations, we can observe that train and test error rates seem lower, ideal, and converged.

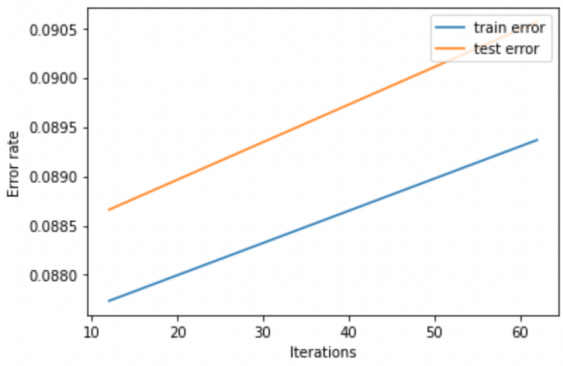


As the learning rate decreases, we can observe that time taken for convergence is increased except at alpha = 0.01. We can say that model at alpha = 0.001 has an ideal time to converge.

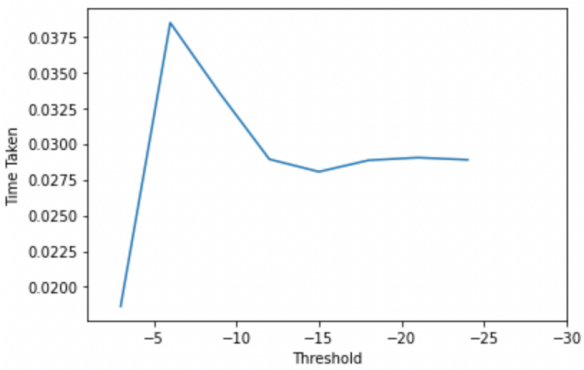
2) Experimented with the model by keeping the learning rate at 0.001 and with a threshold, below are the observations:



We can observe that there is no difference in error rates after the threshold value of 1e-6. Thus, the model converges best at a threshold value of 1e-6.



We can observe that the train and test error rates travel in the same directions with an increase in the number of iterations.



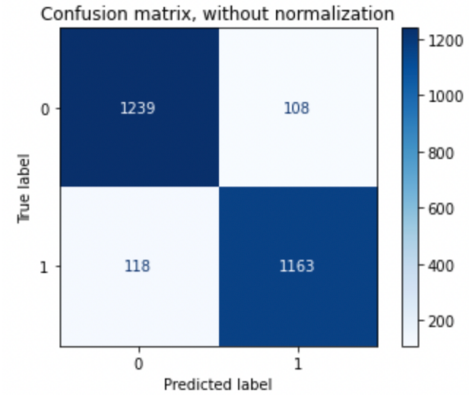
As the threshold decreases, we can observe that time taken for convergence also increased except at threshold = 1e-6.

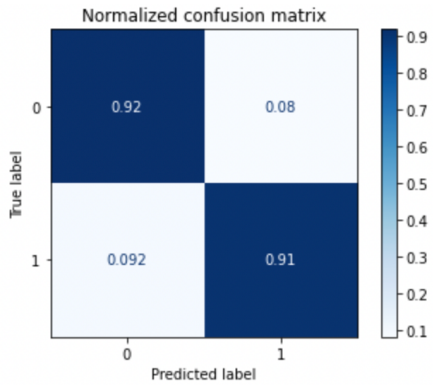
Conclusion:

1) From the SGD classifier experiment, we can conclude that the model has better accuracies for train and test data set at **alpha = 0.001 and tol = 1e-6**.
2) Accuracies on train and test data sets of optimized SGD classifier.

Model	Train Accuracy	Test Accuracy
SGD Classifier	91.94	91.40

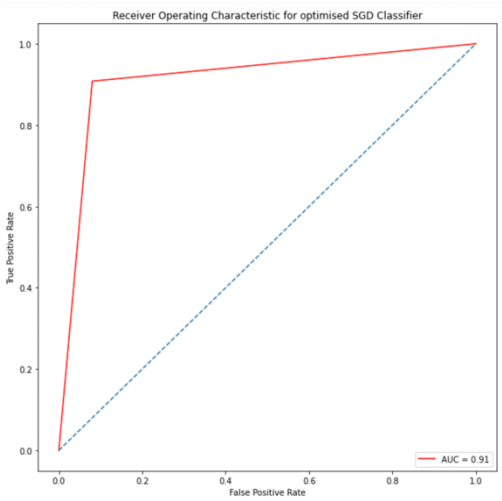
3) Confusion matrix of optimized SGD classifier on test data set.





After optimizing the model with the hyperparameters of alpha = 0.001 and tol = 1e-6, the model has better convergence and classification. There are 108 Type I Errors and 118 Type II Errors, we look for minimizing the Type I errors as we are going to predict the count of rental bikes.

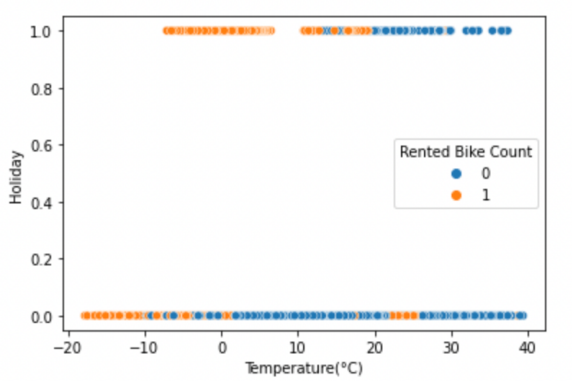
4) ROC and AUC curve of optimized SGD model

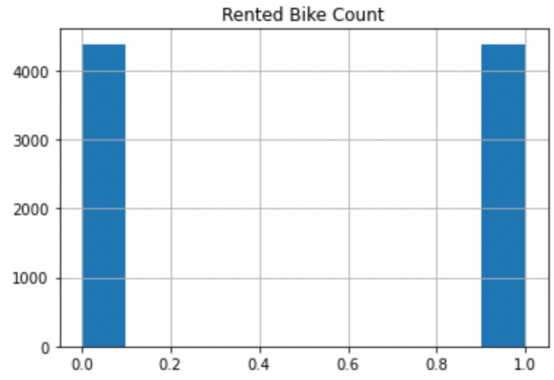


The optimized model with the SGD classifier has an AUC of 0.914. The model with the SGD classifier is good at predicting the classes.

Part 2:

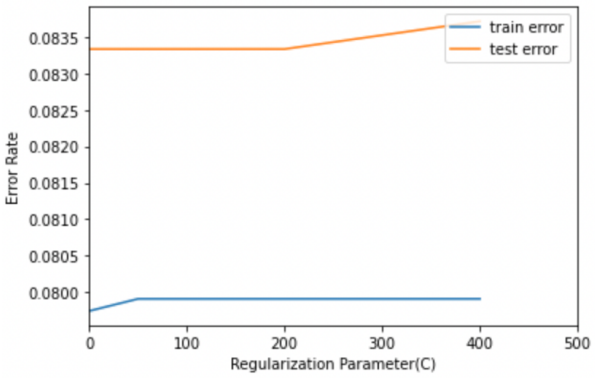
We will use the SVM SVC classifier from a scikit-learn package for the experimentation.



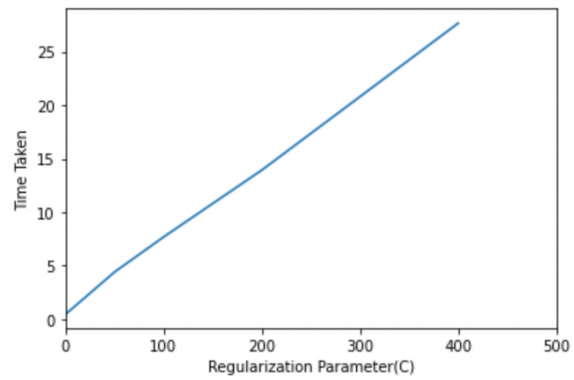


From the plots, we can conclude that there is no linear separable between the classes and need to go for the kernel method to convert to a plan and make a separation of classes, but still, we will keep a linear kernel to experiment with regularization parameter.

1. Experimented with different values of regularization parameter (C) by keeping the kernel function as Linear:

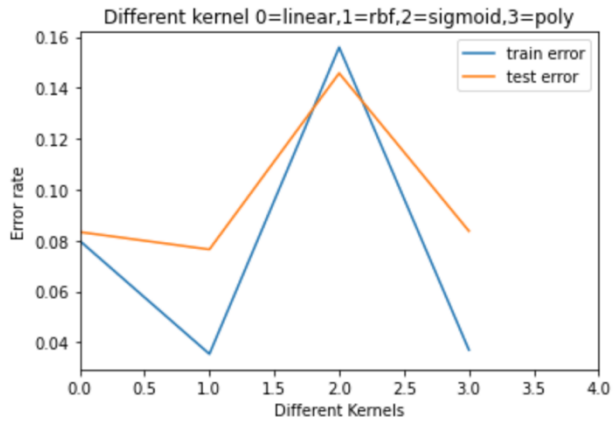


From the experimentation, we can conclude that the model is converging at the regularization parameter value of C= 50.



From the experimentation, we can observe that increase in regularization parameter time taken for convergence also increases.

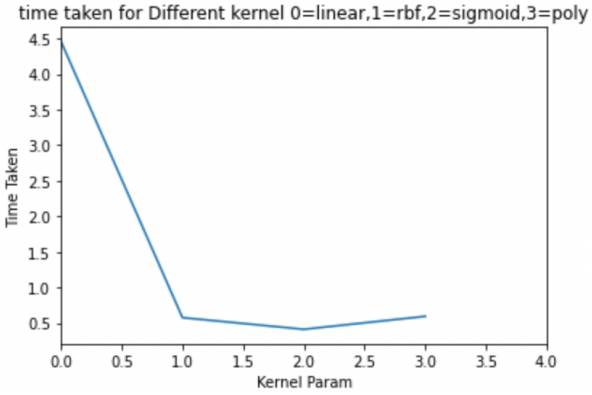
2. Experimenting with different values of kernel methods by keeping C=50:



At C= 50 and using linear as kernel function, the model is good on both train and test sets. From the graph, we can observe that the model built with the rbf kernel function is better than the linear kernel function.

Keeping C=50.0 and kernel = sigmoid model converges but the error rate is high.

Keeping C=50.0 and kernel = poly model converges better than linear but not better than rbf. So, we can conclude that the model with the ‘rbf’ kernel function is the best.



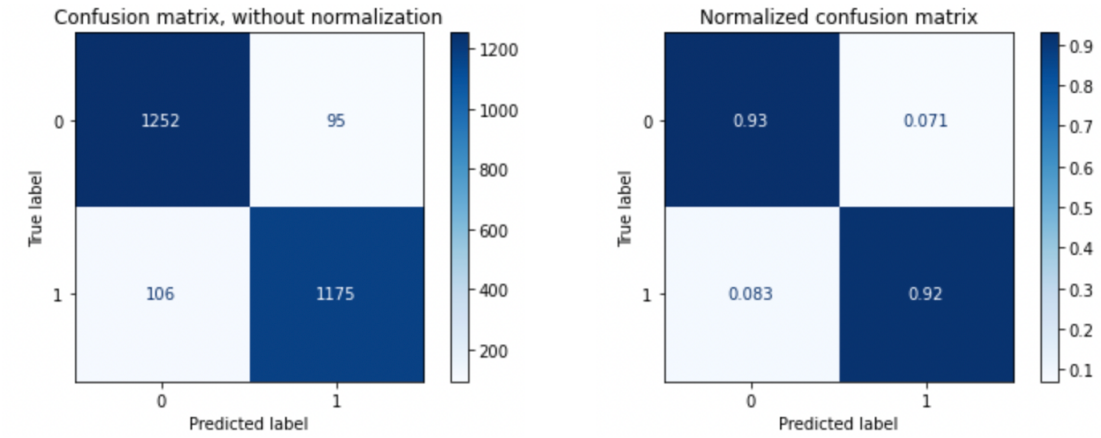
We can conclude that at kernel method = rbf model and sigmoid model have converged faster.

Conclusion:

1) From the SVM classifier experiment, we can conclude that the model has better accuracies for train and test data sets at **C = 50.0 and Gaussian Kernel Radial Basis function**.
2) Accuracies on train and test data sets of optimized SVM classifier.

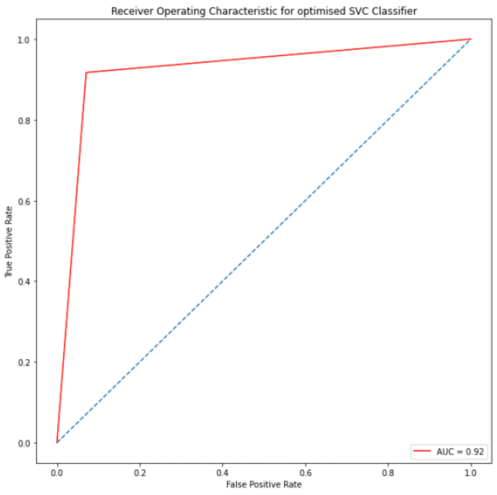
Model	Train Accuracy	Test Accuracy
SVM Classifier	96.46	92.35

3) Confusion matrix of optimized SVM classifier on test data set.



After optimizing the model with the hyperparameters of kernel function = rbf and C=50.0, the model has better convergence and classification. There are 95 Type I Errors and 106 Type II Errors, we look for minimizing the Type I errors as we are going to predict the count of rental bikes.

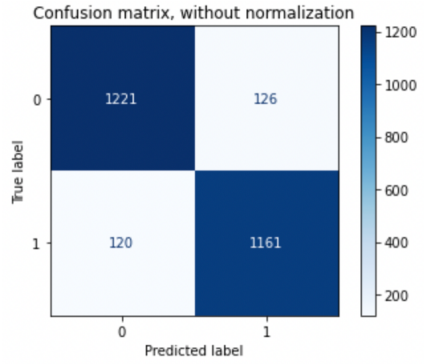
4) ROC and AUC curve of an optimized model with SVM classifier

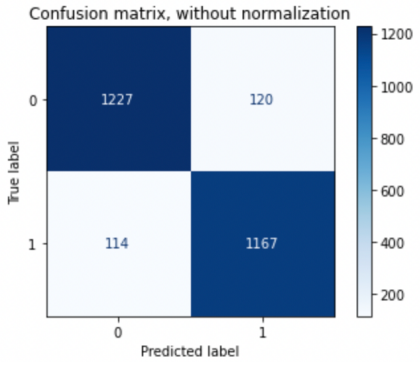


The optimized model using SVC classifier has an AUC of 0.92. We know that the higher the AUC, the better the model is at predicting 0 classes as 0 and 1 classes as 1. The model with the SVC classifier is good at predicting the classes.

Part 3:

The Decision Tree classifier from the scikit-learn package is used for the experimentation. We will perform experimentation for pruning the tree in pre-pruning and post-pruning phases to identify the right set of hyperparameters.



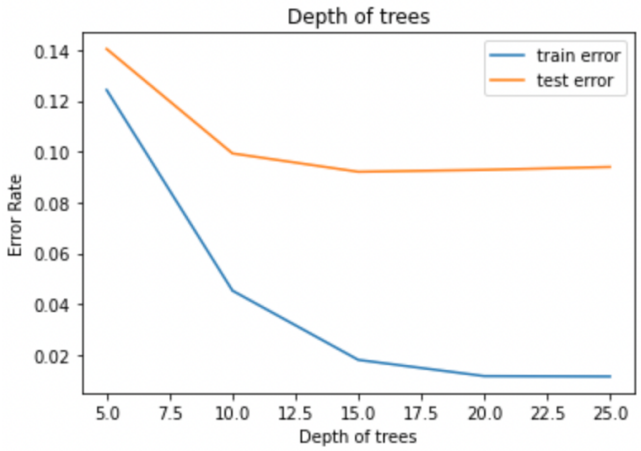


After comparing Entropy and Gini index, we will use the Gini criterion for split information of nodes as we are only dealing with the prediction of classes and not dealing with probabilities of classes. The Gini is much faster whereas the results obtained using Entropy are better. Gini Impurity is better as compared to entropy for selecting and computing the best features after every new splitting.

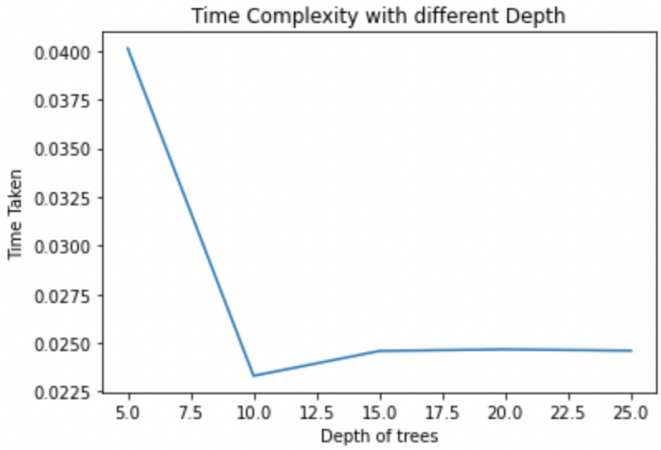
	Train Accuracy	Test Accuracy
GINI	98.5	90.6
ENTROPY	98.4	91.1

Pre-Pruning:

1. Keeping the minimum number of samples required to split at 5 and the minimum number of samples required to be at a leaf node as 1 and varying the maximum depth of the tree.

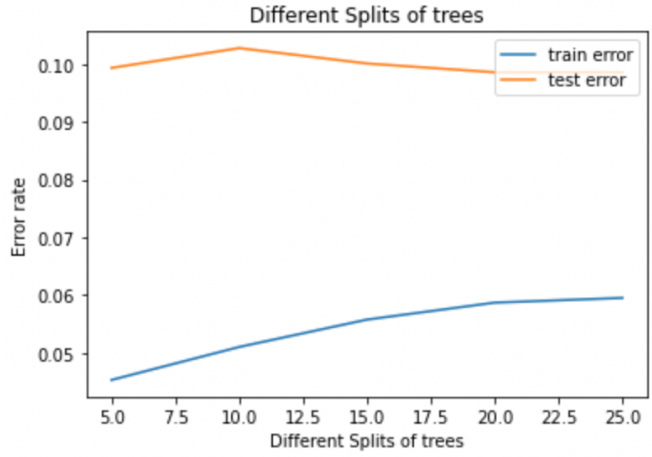


As the depth of the tree increases, the train error rate decreases which tend towards overfitting. The test error rate decreases and increases a bit. Even though error rates are lower at 15 and 20 tree depth, 10 will be the ideal maximum depth of tree for both train and test data sets as train error rate leads to an overfitting situation.

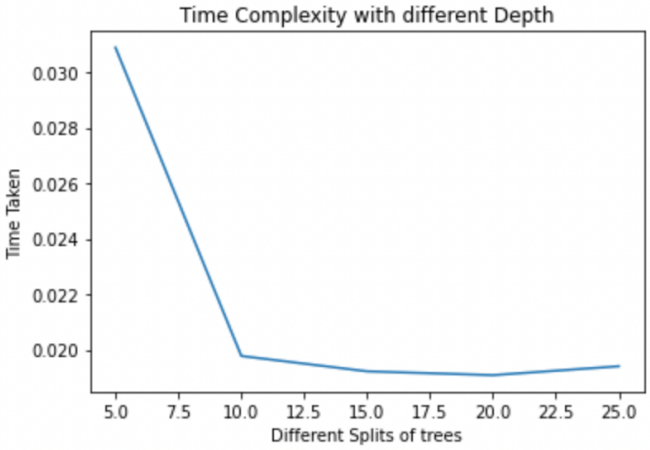


We can observe a decrease and then increase the type of pattern in the time taken for convergence with an increase in the depth of trees. But, at depth = 15.0 model took more time to converge.

2. Keeping the maximum depth of the tree at 10 and the minimum number of samples required to be at a leaf node as 1 and varying the minimum number of samples required to split:

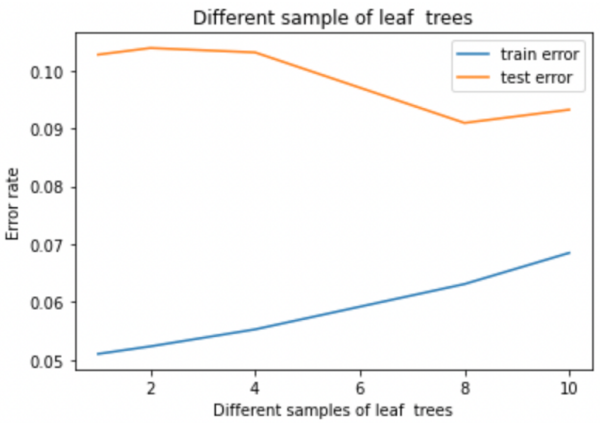


The number of splits didn't impact the model error rates much. But we can conclude min_sample_split = 15.

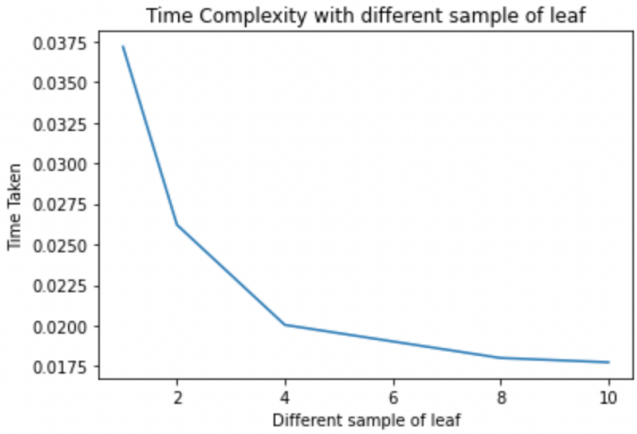


As the number of splits increases, the time taken to convergence decreases. However, min_samples_split has not had much effect on the model time taken for convergence.

3. Keeping the maximum depth of the tree at 10 and the minimum number of samples required to split as 15 and varying the minimum number of samples required to be at a leaf node:



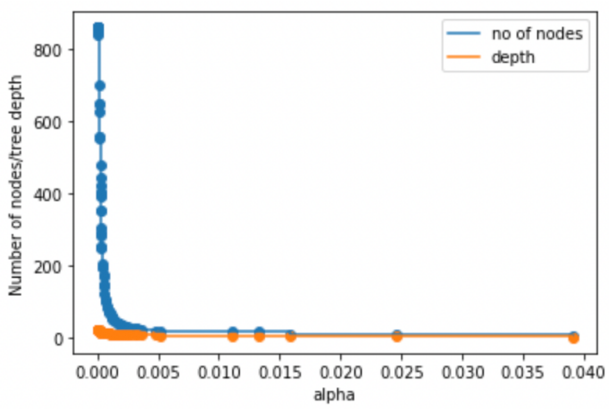
The minimum number of samples at the leaf node hasn't had much effect on the model error rates. But at min_samples_leaf = 8, both train and test error rates are lower than at the other number of leaf nodes to be considered.

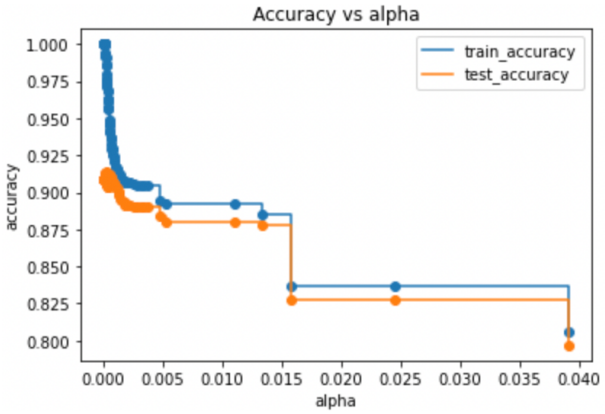


The number of splits increases, and the time taken to convergence decreases. But min_samples_leaf has not much effect on the model time taken for convergence.

Post-Pruning:

In this experiment, we will check cost complexity parameters varying w.r.t depth, no of nodes, and accuracies.





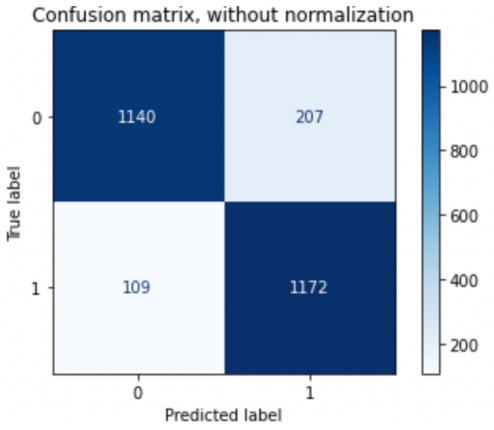
At the complexity parameter value of 0.013, the train and test accuracies are better.

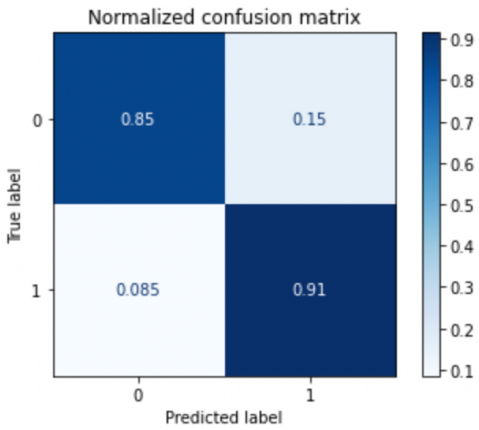
Conclusion:

1. From the Decision Tree classifier experiment, we can conclude that the model test and train accuracies are better at **max_depth=10, min_samples_split=15, min_samples_leaf=8, cp=0.013**
2. Accuracies on train and test data sets of optimized Decision Tree classifier.

Model	Train Accuracy	Test Accuracy
Decision Tree Classifier	88.47	87.74

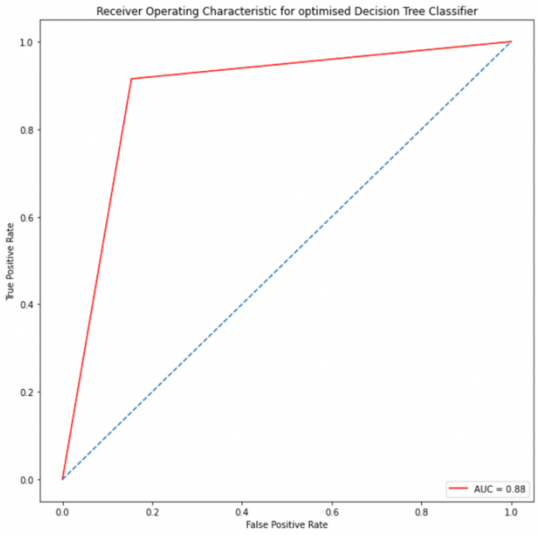
3. Confusion matrix of optimized Decision Tree classifier on test data set.





After optimizing the model with the hyperparameters at max_depth=10, min_samples_split=15, min_samples_leaf=8, cp=0.01, the model has better convergence.

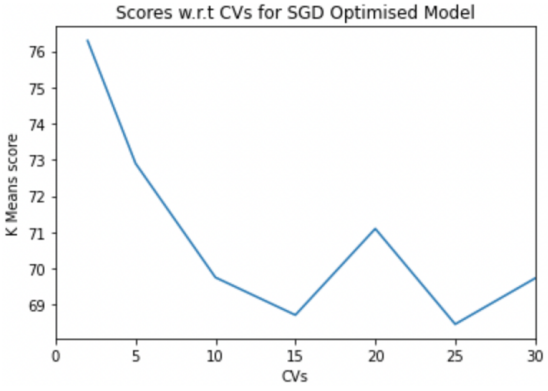
4. ROC and AUC curve of an optimized model with SVM classifier



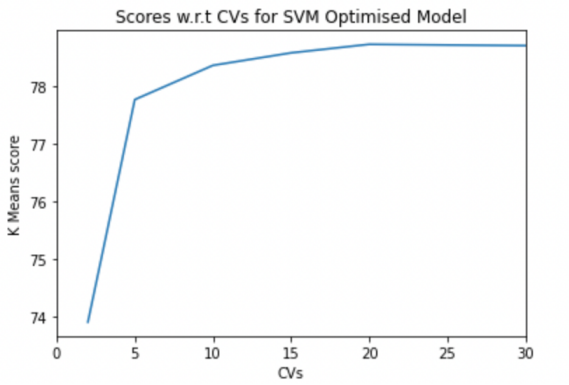
The optimized model using the Decision Tree classifier has an AUC of 0.88. The model with the DT classifier is good at predicting the classes.

Part 4:

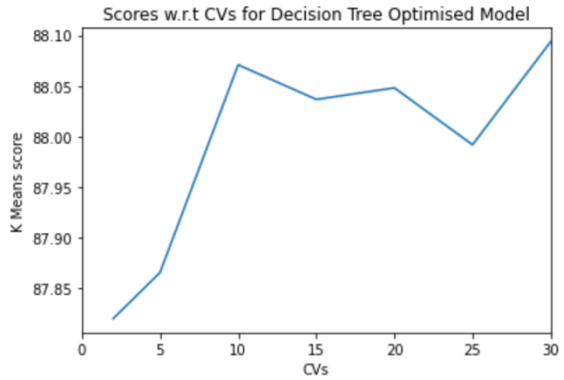
Now we will use K-Fold Cross Validation Techniques to check the model performance for the optimized SGD, SVM, and Decision Tree Classifiers.



The mean value of accuracy of the model decreases from 76% to 70% with an increasing number of folds. But without the cross-validation technique, the accuracies are around 91 for the SGD model. That means the optimized model is not best for classifying the unseen data or new data.



The mean value of accuracy of the SVM model increases from 74% to 79% with an increasing number of folds. But without the cross-validation technique, the accuracies are around 95 for the SVM model. That means the optimized model is not best for classifying the unseen data or new data.



The mean value of accuracy of the Decision Tree model increases from 87.5% to 88% with an increasing number of folds. But without the cross-validation technique, the accuracies are around 88 for the SVM model. That means the optimized model is best for classifying the unseen data or new data.

Results:

Through SGD classifier experimentation, we found that hyperparameters like learning rate and threshold are effective for better classification accuracy. Through SVM classifier experimentation, we found that hyperparameters like kernel function and regularization parameter are effective for better accuracy. Through Decision Tree classifier experimentation, we found that hyperparameters like the GINI index, maximum depth of the tree, and the number of samples required at a leaf node are effective for better accuracy. Therefore, machine learning involves balancing accuracy and computational limitations.

	Train Accuracy	Test Accuracy	AUC	Cross-Validation	Type I Error
Optimized SGD	91.94	91.2	0.91	70-76	108
Optimized SVM	96.46	92.35	0.92	74-79	95
Optimized DT	88.47	87.74	0.88	87.5-88	207

1) From the normal split accuracy table, we can say that all classifiers are good at classifying the data whereas the SVM classifier may lead to overfitting the training dataset or good with the present dataset
2) When we observe the AUC measure of separability, the model with the SVM classifier is highly capable of distinguishing between class 0 and class 1, on the other hand, the AUCs for other classifiers are almost the same.
3) From the cross-validation perspective, we can conclude that the model with the Decision Tree classifier is better at classifying the existing data and new data when compared to the other classifiers as the scores are almost similar in the case of k-fold and normal split.
4) As we focus on predicting classes 0 and 1, we concentrate on Type I error to minimize them as possible. We can say that the SVM classifier has a good class separation capacity but its performance on new data is bad. SGD might be good at classifying the patients with a disease or no disease.

Overall, we can conclude that the model with the Decision Tree classifier is good at predicting classes from the accuracy, AUC, and Cross-Validation point of view even though it is slightly performed lower when compared with SVM and SGD, it is great with unseen or new data.