BUAN 6341: Applied Machine Learning

Project 2

Aji Somaraj
Axs161031

Data Description

Dataset 1: Adult Income Dataset

Dataset 1 is a subset from the 1994 US Census form UCI data repository , which is used to relate education, heritage and age (among others) against income, in this case, whether income is above or below $50,000 per year. Governments can use this data to determine the most impactful factors for increasing household income. It contains approximately 32000 observations, with 15 variables. The data has less imbalance compared to other classification datasets. The project aims to do SVM, decision tree, xgboost models and study learning curves and the bias variance trade-offs. The structure of dataset is shown below.

```
'data.frame':   32561 obs. of  15 variables:
 $ age           : int  39 50 38 53 28 37 49 52 31 42 ...
 $ workclass     : Factor w/ 9 levels " ?"," Federal-gov",..: 8 7 5 5 5 5 5 7 5 5 ...
 $ fnlwgt        : int  77516 83311 215646 234721 338409 284582 160187 209642 45781 159449
 $ education     : Factor w/ 16 levels " 10th"," 11th",..: 10 10 12 2 10 13 7 12 13 10 ...
 $ education-num : int  13 13 9 7 13 14 5 9 14 13 ...
 $ marital-status: Factor w/ 7 levels " Divorced"," Married-AF-spouse",..: 5 3 1 3 3 3 4 3
...
 $ occupation    : Factor w/ 15 levels " ?"," Adm-clerical",..: 2 5 7 7 11 5 9 5 11 5 ...
 $ relationship  : Factor w/ 6 levels " Husband"," Not-in-family",..: 2 1 2 1 6 6 2 1 2 1 .
 $ race          : Factor w/ 5 levels " Amer-Indian-Eskimo",..: 5 5 5 3 3 5 3 5 5 5 ...
 $ sex           : Factor w/ 2 levels " Female"," Male": 2 2 2 2 1 1 1 2 1 2 ...
 $ capital-gain  : int  2174 0 0 0 0 0 0 0 14084 5178 ...
 $ capital-loss  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ hours-per-week: int  40 13 40 40 40 40 16 45 50 40 ...
 $ native_country: Factor w/ 42 levels " ?"," Cambodia",..: 40 40 40 40 6 40 24 40 40 40 ..
 $ class         : Factor w/ 2 levels " <=50K"," >50K": 1 1 1 1 1 1 1 2 2 2 ...
```

Dataset 2: Bank Marketing Data Set

The data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed. It contains approximately 45000 observations with 12 categorical variables and 7 numerical variables as its features and a binary category as its response variable. The data has high imbalance compared to other classification datasets. SMOTE sampling was used to upscale the class. The project aims to do SVM, decision tree, xgboost models and study learning curves and the bias variance trade-offs. The structure of dataset is shown below.

```
'data.frame':   45211 obs. of  17 variables:
 $ age      : int  58 44 33 47 33 35 28 42 58 43 ...
 $ job      : Factor w/ 12 levels "admin.","blue-collar",..: 5 10 3 2 12 5 5 3 6 10
 $ marital  : Factor w/ 3 levels "divorced","married",..: 2 3 2 2 3 2 3 1 2 3 ...
 $ education: Factor w/ 4 levels "primary","secondary",..: 3 2 2 4 4 3 3 3 1 2 ...
 $ default  : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 2 1 1 ...
 $ balance  : int  2143 29 2 1506 1 231 447 2 121 593 ...
 $ housing  : Factor w/ 2 levels "no","yes": 2 2 2 2 1 2 2 2 2 2 ...
 $ loan     : Factor w/ 2 levels "no","yes": 1 1 2 1 1 1 2 1 1 1 ...
 $ contact  : Factor w/ 3 levels "cellular","telephone",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ day      : int  5 5 5 5 5 5 5 5 5 5 ...
 $ month    : Factor w/ 12 levels "apr","aug","dec",..: 9 9 9 9 9 9 9 9 9 9 ...
 $ duration : int  261 151 76 92 198 139 217 380 50 55 ...
 $ campaign : int  1 1 1 1 1 1 1 1 1 1 ...
 $ pdays    : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
 $ previous : int  0 0 0 0 0 0 0 0 0 0 ...
 $ poutcome : Factor w/ 4 levels "failure","other",..: 4 4 4 4 4 4 4 4 4 4 ...
 $ y        : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
```

Classification Algorithm

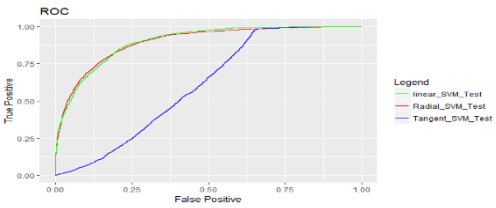
1. Support Vector Machines(SVM)

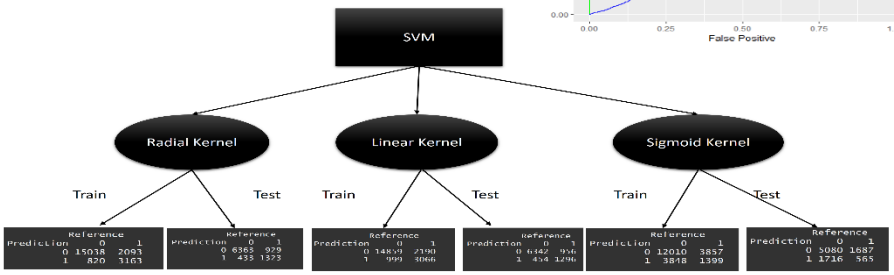
The package kernlab in R was used to implement SVM. The metrics like ROC, AUC, Accuracy, etc. are used to study the performance of the model. The algorithm was used on both the datasets and different types of kernels were used to study how well the model classifies the model. Learning Curves, Error vs train size and Error vs Time was also used to study the algorithm. The learning curves gives an idea of the bias vs variance tradeoff. The project mainly aims to study how well the learning curves tells the performance of the model. Model is not optimum, as all the features in the dataset are included in this study

Dataset 1

The kernels used in adult income was sigmoid, radial and linear. The kernel was chosen based on the performance. The performance metrics, ROC curves and confusion matrices are given below:

		Accuracy	Sensitivity	Specificity	Kappa	AUC
Train	Sigmoid	0.6351	0.7573	0.2662	0.0235	0.6198692
Test		0.6239	0.7475	0.2509	-0.002	0.6071969
Train	Linear	0.849	0.937	0.5833	0.563	0.9035344
Test		0.8442	0.9332	0.5755	0.5496	0.8986451
Train	Radial	0.862	0.9483	0.6018	0.5985	0.9035344
Test		0.8495	0.9363	0.5875	0.5654	0.8986451

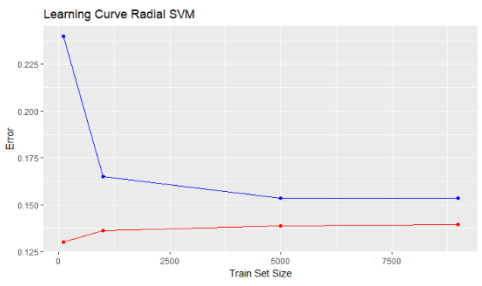


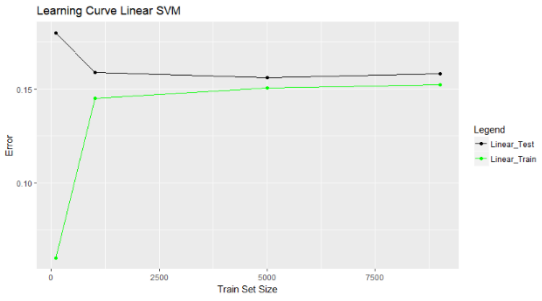


Starting with the sigmoid kernel, the accuracy, AUC and kappa values were very low and hence linear kernel was used which made significant improvement in the metrics. The kappa value was greater than 0.5 which is considered to be good for a classification problem. Radial kernel was then used which gave more improved results compared to the linear and sigmoid kernels. So, among the three kernels radial performed better. The ROC curve tells how well the model performs, we can see that radial and linear has almost equal AUC and sigmoid kernel was performing close to a random model that implies sigmoid kernel is considered the worst in this problem, while linear and radial performs better. Considering the test accuracy, specificity, sensitivity, kappa values radial kernel can be considered as the best among these 3 kernels.

Learning Curves:

Error VS Train Set Size:

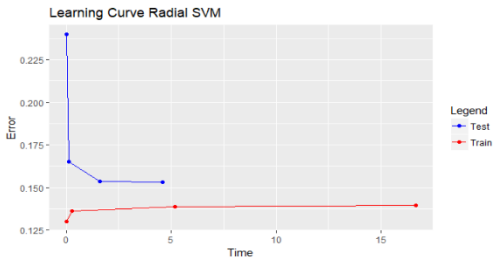


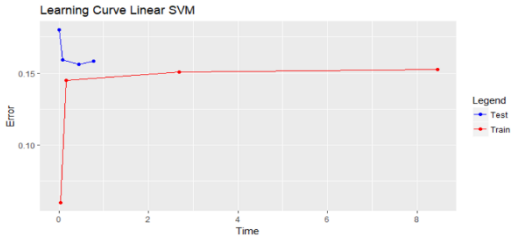


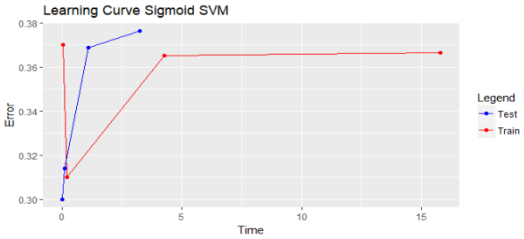
The errors of train and test sets was calculated for various train sizes and plotted. Since the test set size was very low, the study was limited to very few data points (True for all the models below) but it gives us a general picture of the model performance and bias vs variance trade-off. In the radial kernel, initially when the train size is low we can see that the errors are high for test and low for train, as the size increases the train error reach a plateau and the test error also comes down., which is close to an ideal learning curve. In the linear kernel, initially when the train size is low we can see that the errors are high for test and low for train, as the size increases the train error increases and test error does not make any significant decrease in error., the gap between the train and test error are low which implies that it suffers from high bias(underfitting). In the sigmoid which was found to bad model, `we can see that train error is higher for less train size than the test error initially, as the size increases the model is found to be highly biased and indicates underfitting.

Error vs Time:

These plots doesnot give much insights like the error vs size as the time to test the model takes less processing time compared to the train time. In the radial kernel we can see a similar curve as the error vs train size radial graph initially error is high for test and low for train and as the size increases the train error reach a plateau and the test error also comes down. But due to the low testing processing time the graph doesnot give much information. In linear kernel, initially when the train size is low we can see that the errors are high for test and low for train, as the size increases the train error increases and test error does not make any significant decrease in error. In sigmoid kernel, the graph doesn't give much information.



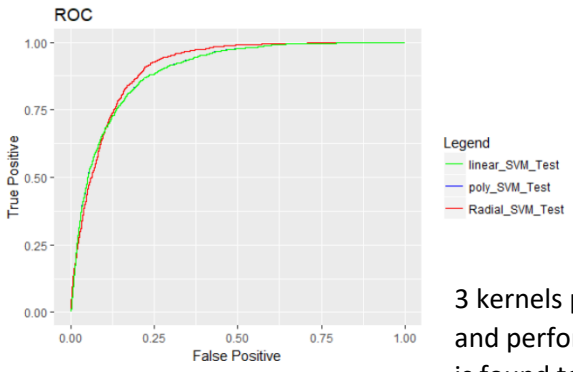


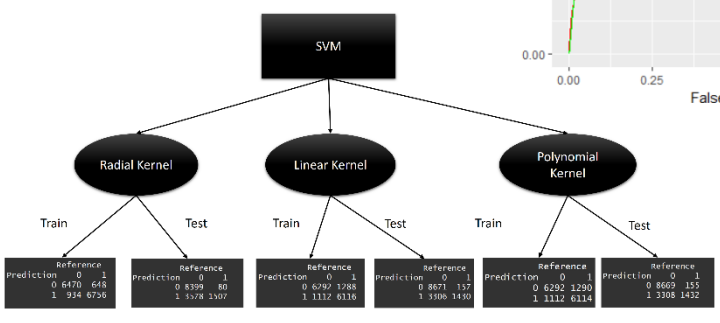


Dataset 2: Bank

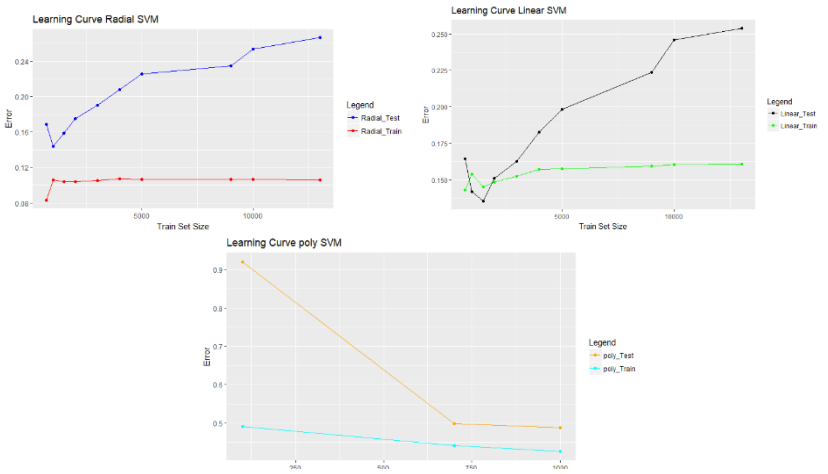
The kernels used in Bank dataset are polynomial, Linear, Radial. The kernel was chosen based on the performance. The data being sampled to compensate the imbalance makes the data overfitting The performance metrics, ROC curves and confusion matrices are given below:

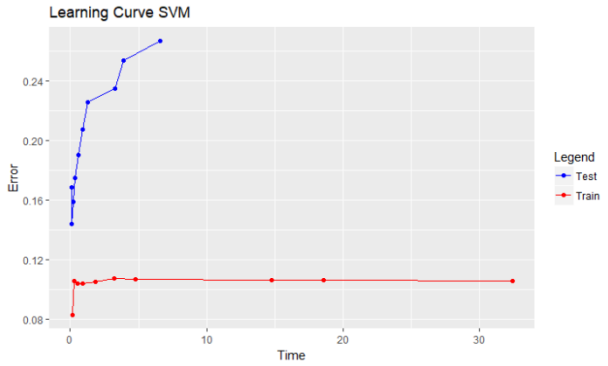
		Accuracy	Sensitivity	Specificity	Kappa	AUC
Train	Polynomial	0.8378	0.8498	0.8258	0.6756	0.9125617
Test		0.7447	0.7238	0.9023	0.3363	0.8959333
Train	Linear	0.8379	0.8498	0.826	0.6759	0.9125573
Test		0.7447	0.724	0.9011	0.3359	0.8959326
Train	Radial	0.8932	0.8739	0.9125	0.7863	0.957139
Test		0.7303	0.7013	0.9496	0.3327	0.9058452





3 kernels performed well based on the ROC, and performance metrics but the radial SVM is found to be the best model for bank data among kernel SVM. The radial function could classify the data better compared to the other 2.





Learning Curves

Error Vs Train Size:

Radial svm is having high variance when the size of train data increases which implies that the model is overfitted. Linear svm is also found to be overfitted. Polynomial SVM is showing a different trend, its show a curve similar to an ideal curve. The overfitting problem can be removed by adding more data to train or by removing features.

Error vs Time:

Doesn't give much information we can see the train error plateaus when the time increases but the test error was found to show a curve which does not show any insights. The test error increases at a high level as time and size increases. Seeing the plot its highly overfitted, here only the radial svm is shown

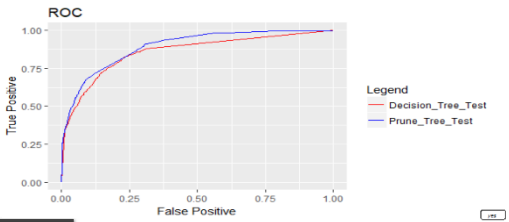
2. Decision Tree Classifier

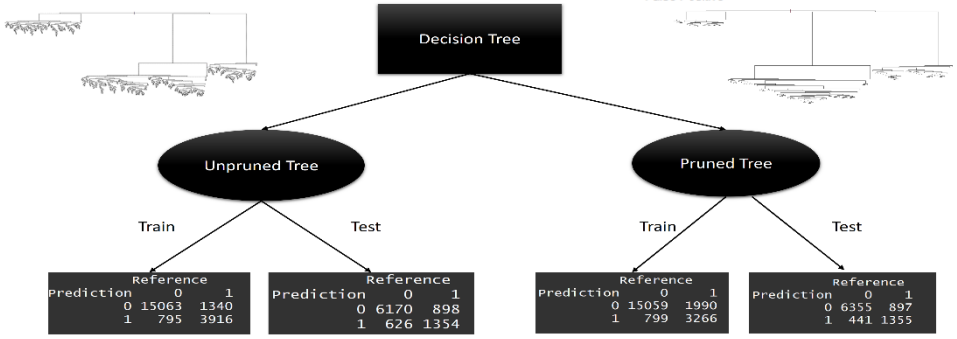
The package rpart in R was used to implement decision tree. The metrics like ROC, AUC, Accuracy, etc. are used to study the performance of the model. The algorithm was used on both the datasets and the tree was pruned. Learning Curves, Error vs train size and Error vs Time was also used to study the algorithm. The learning curves gives an idea of the bias vs variance tradeoff. Model is not optimum, as all the features in the dataset are included in this study, the project mainly aims to study how well the learning curves tells the performance of the model. Information gain was used for split in decision tree.

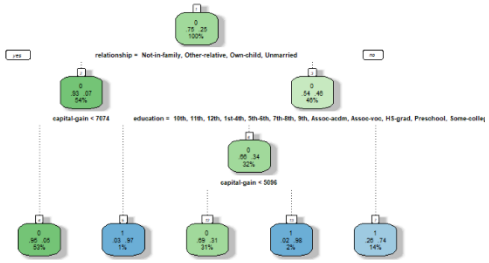
Dataset 1:

The performance metrics, ROC curves and confusion matrices are given below:

		Accuracy	Sensitivity	Specificity	Kappa	AUC
Train	Unpruned Tree	0.8989	0.9499	0.7451	0.7199	0.9634
Test		0.8316	0.9079	0.6012	0.5306	0.8566
Train	Pruned Tree	0.8679	0.9496	0.6214	0.6178	0.8749
Test		0.8521	0.9351	0.6017	0.5758	0.8912



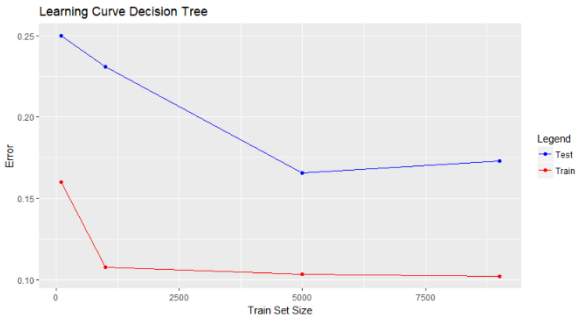


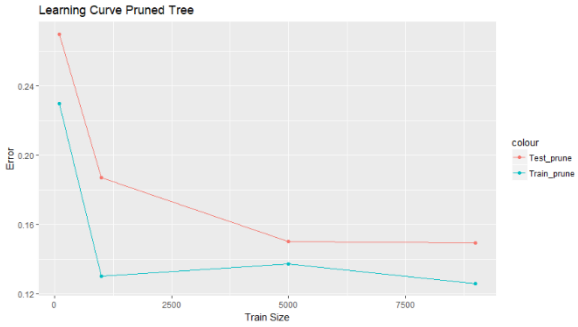


Initially the adult data is modelled using rpart decision with default parameters the default tree shown to right. The decision tree obtained from default parameter is highly overfitted and small and it can't be pruned further. To grow the tree further parameter cp was changed to -1 to fully grow the tree. Then the tree was pruned based on the minimum value of cp. The pruned tree performed better compared to the default and fully-grown tree. The performance metrics also tells us that the pruned is better. The ROC curve is also showing better results for the pruned tree. The trees, confusion matrix is also shown above.

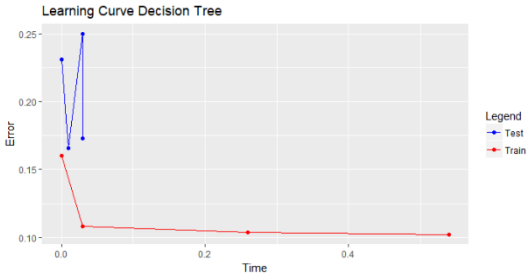
Learning Curve

Error Vs Train size





Error vs time



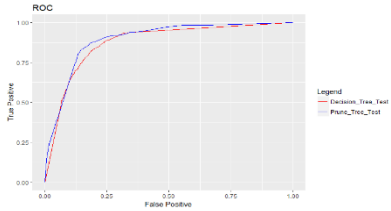
The learning curve of both pruned and fully-grown tree is shown. The prune tree is showing an similar pattern to an ideal learning curve but the fully grown tree is having high overfitting or high variance.

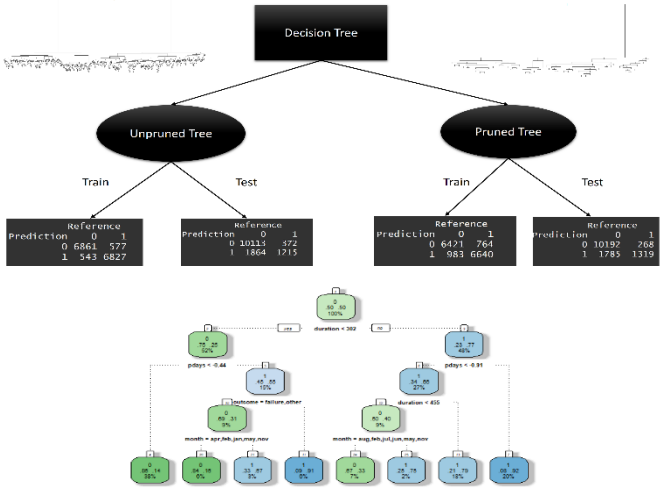
The time learning graph doesn't provide much insights about the test error, train error increases with time, the graph shows the model is highly overfitted. The lack of train size is major drawback in this study which is true for all the models and the learning curves in this project

Dataset 2:

The performance metrics, ROC curves and confusion matrices are given below:

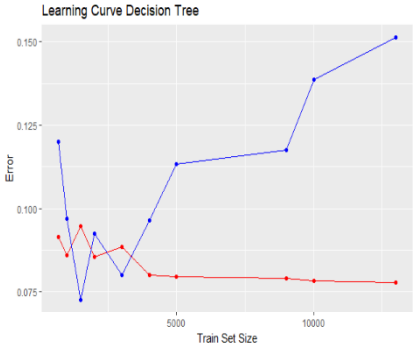
		Accuracy	Sensitivity	Specificity	Kappa	AUC
Train	Unpruned Tree	0.9244	0.9267	0.9221	0.8487	0.9845
Test		0.8352	0.8444	0.7656	0.4333	0.8743
Train	Pruned Tree	0.882	0.8672	0.8968	0.764	0.9343
Test		0.8486	0.851	0.8311	0.4822	0.8928





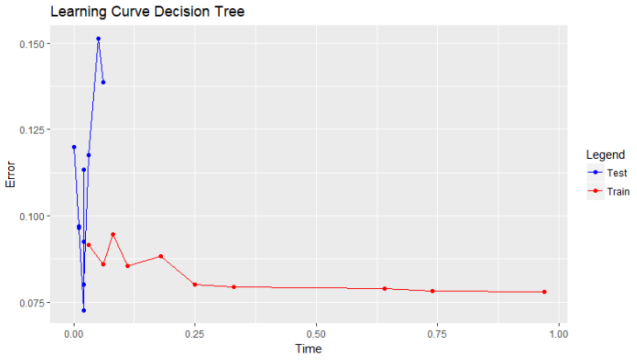
Initially the bank data is modelled using rpart decision with default parameters the default tree shown to right. The decision tree obtained from default parameter is highly overfitted and small and it can't be pruned further. To grow the tree further parameter cp was changed to -1 to fully grow the tree. Then the tree was pruned based on the minimum value of cp. The pruned tree performed better compared to the default and fully-grown tree. The performance metrics also tells us that the pruned is better. The ROC curve is also showing better results for the pruned tree. The trees, confusion matrix is also shown above.

Learning Curves



Error vs Train size

The data being sampled the training set is very less and it suffers from overfitting, getting more balanced data will add more accuracy to the model. Now the model is highly overfitted for both the pruned and unpruned data. The learning curves for both pruned and unpruned is shown and both has overfitted



Error vs time

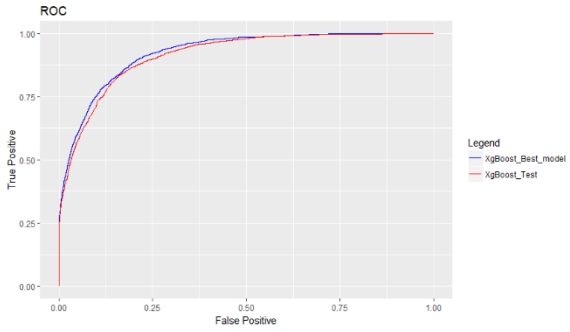
The error vs the time doesn't give much insights as the time to test is very low, but as time increase the train error plateaus to a lower error value. As the test curve performance is low we can't make any conclusion. Seeing the curves its highly overfitted.

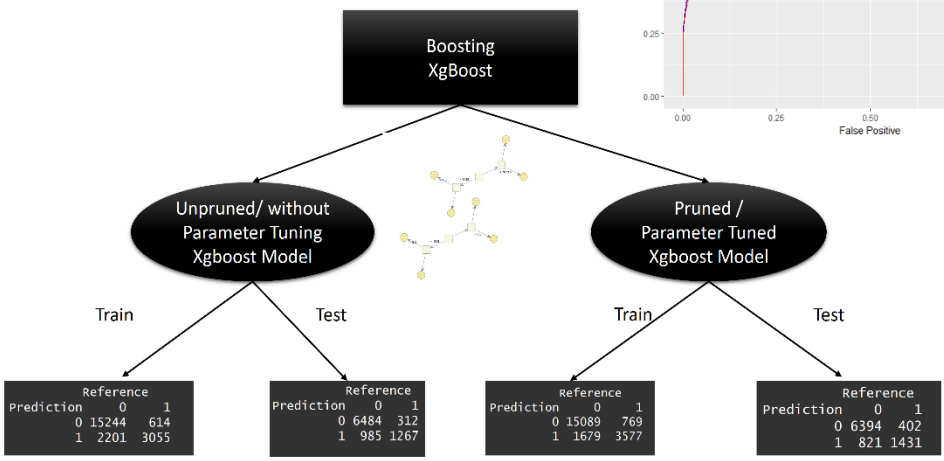
3. Boosting: eXtreme Gradient Boosting algorithm[xgboost]

Dataset 1:

The performance metrics, ROC curves and confusion matrices are given below:

		Accuracy	Sensitivity	Specificity	Kappa	AUC
Train	Unpruned/ without Parameter Tuning	0.8667	0.8738	0.8327	0.6034	0.9282
Test	Xgboost Model	0.8567	0.8681	0.8024	0.5741	0.9144
Train	Pruned/ Parameter Tuned Xgboost	0.8841	0.8999	0.8231	0.6709	0.9421
Test	Model	0.8648	0.8862	0.7807	0.6145	0.9243

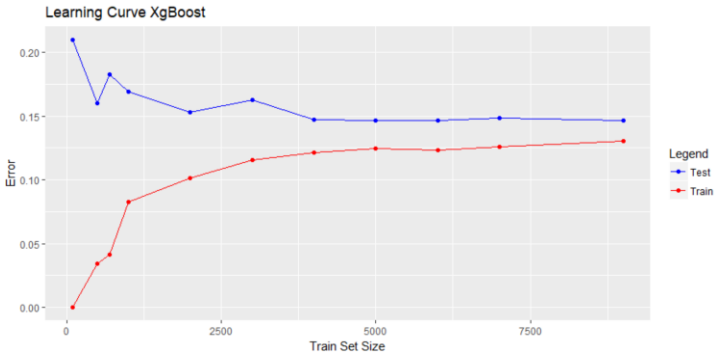




The boosting algorithm used here is xgboost using xgboost package, which gives higher performance than any other model in this dataset. The metrics and confusion matrices are shown above. The prune or parameter tuned xgboost performs better compared to non-parametrized boosting. The parameters like min_child_weight, max depth is used to prevent overfitting. scale_ pos_weight deals with data imbalance. The inbuilt cross validation is used to find the best iteration(nrounds) among the lot. The best model is marked in green which is pruned xgboost model. A default xgboost tree is also shown above

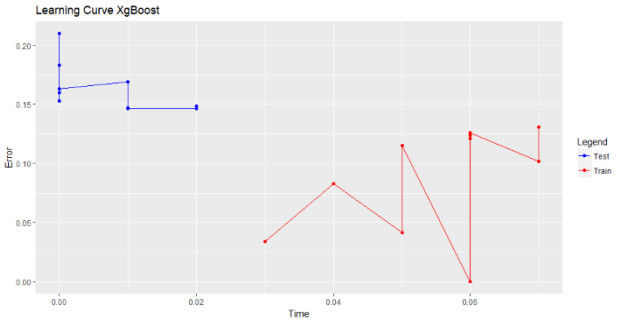
Learning Curves

Error vs Train size



Initially the train error is low and test error is high then train error increases and test decreases with increase in size of data then the train error flattens to a value, the test error goes parallel to the train error decreasing error which implies the model is close to an ideal curve but due to lack of more test data point. We can't completely agree if the model is showing ideal bias variance trade-off. The error values are very low which can be seen in the plot. Here the xgboost is not optimum because the parameters are not set optimally for the points hence the plot.

Error vs Time

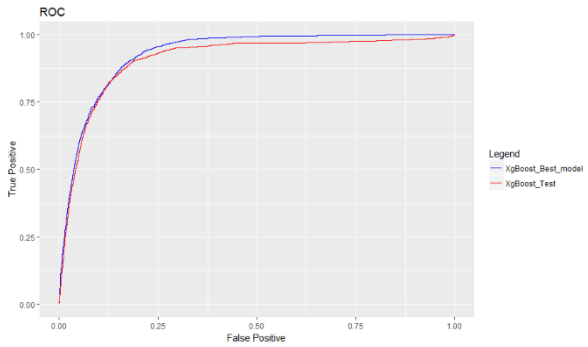


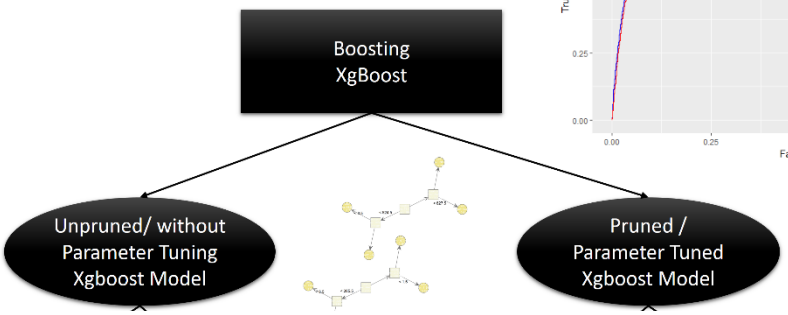
Error vs time does not give much insights as the time to train and test are totally different.

Dataset 2:

The performance metrics, ROC curves and confusion matrices are given below:

		Accuracy	Sensitivity	Specificity	Kappa	AUC
Train	Unpruned/ without Parameter Tuning	0.986	0.9851	0.9933	0.9289	0.9978
Test	Xgboost Model	0.9046	0.9309	0.6255	0.4784	0.9093
Train	Pruned/ Parameter Tuned Xgboost	0.9178	0.9337	0.7271	0.5324	0.9449
Test	Model	0.9073	0.9272	0.6614	0.4686	0.9307

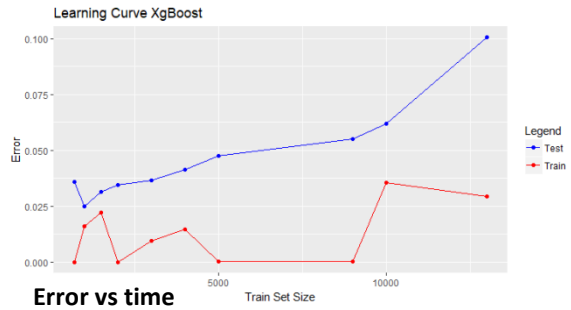




The boosting algorithm used here is xgboost using xgboost package, which gives higher performance than any other model in this dataset. The metrics and confusion matrices are shown above. The prune or parameter tuned xgboost performs better compared to non-parametrized boosting. The parameters like min_child_weight, max depth is used to prevent overfitting. scale_ pos_weight deals with data imbalance. The inbuilt cross validation is used to find the best iteration(nrounds) among the lot. The best model is marked in green. A default xgboost tree is also shown above. The bank data is highly imbalance, the scale_ pos_weight parameter deals with that. The best model is the pruned xgboost with high AUC, accuracy. ROC curve also shows the same.

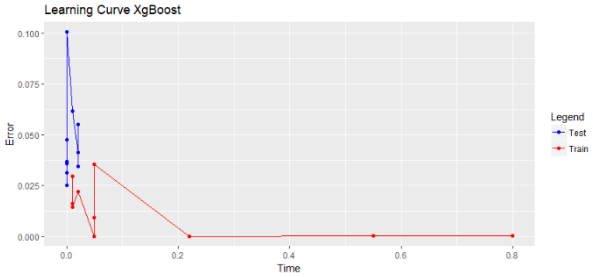
Learning Curves

Error vs Train size



Initially the train error is low and test error is high then train error and test increases with increase in size of data then. Due to lack of data points and with non-parametrized xgboost performs badly as the train data size is increased. Due to lack of feature selection and parameter tuning the xgboost here shows high variance plot. Which can be solved by taking more datapoints for the study and with proper feature selection and tuning parameters we can make a model much better than this.

Error vs time

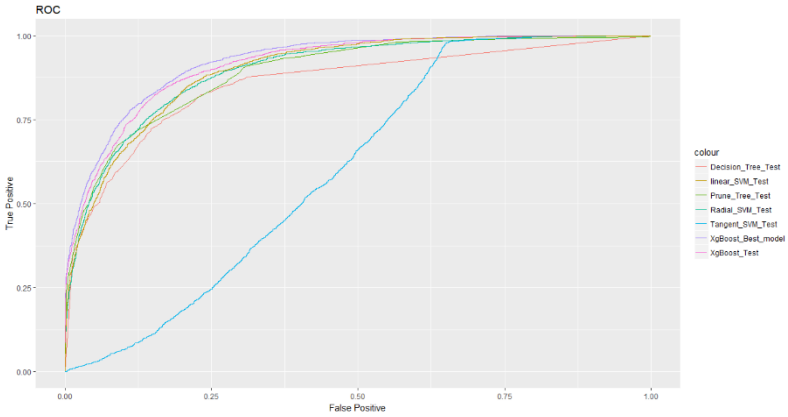


The error vs time doesn't give much insights as the train and test times are totally different. Train error decreases and almost becomes zero, no insights about can made about the test data.

SUMMARY

Dataset 1:

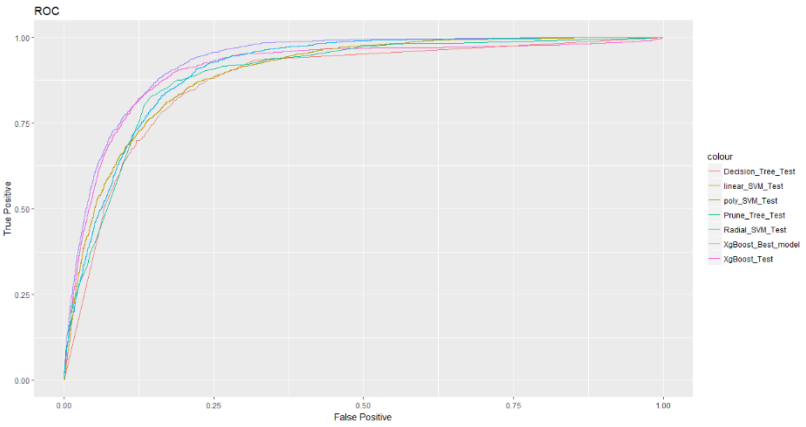
	SVM			Decision Tree		XgBoost	
	Sigmoid	Linear	Radial	Unpruned Tree	Pruned Tree	Unpruned	Pruned
Accuracy	0.6239	0.8442	0.8495	0.8316	0.8521	0.8567	0.8648
Sensitivity	0.7475	0.9332	0.9363	0.9079	0.9351	0.8681	0.8862
Specificity	0.2509	0.5755	0.5875	0.6012	0.6017	0.8024	0.7807
Kappa	-0.002	0.5496	0.5654	0.5306	0.5758	0.5741	0.6145
AUC	0.6072	0.8986	0.8986	0.8566	0.8911514	0.9144327	0.9242987



The algorithms SVM, Decision Tree, and Xgboost was run on the adult income dataset and we found that pruned/ parameter tuned xgboost performs better than any other algorithm with high AUC and high Accuracy. The metric values for every model is shown above. Xgboost is having kappa statistic of more than 0.6, AUC of more than 0.9 and high accuracy as well. Other model's values are also given to compare. Sigmoid kernel(blue) performed close to a random model with relatively low performance metric values. Cross validation could improve the accuracy of SVM and Decision tree the models. It was not used in the study because of its high run time. The k-fold cross validation improved the accuracy of SVM and Decision Tree 0.85166252 and 0.842 respectively. The xgboost had inbuilt cross validation to find best iteration so that was used. The decision trees were made to fully grow to study how pruning improves the model. Pruning in decision tree significantly increased the accuracy and AUC. Some learning curves doesn't give much insights due to lack of test and train size used in the study. The project mainly aims to get insights from the learning curve, figure out the bias and variance tradeoff, which models are underfitting and overfitting. The ROC curve of all the models is shown, xgboost as an edge over other algorithm due to its ensemble learning methods. The information gain was used in the decision tree over gini index as both gave similar results so, the tree was trained based on the information gain.

Dataset 2:

	SVM			Decision Tree		XgBoost	
	Polynomial	Linear	Radial	Unpruned Tree	Pruned Tree	Unpruned	Pruned
Accuracy	0.7447	0.7447	0.7303	0.8352	0.8486	0.9046	0.9073
Sensitivity	0.7238	0.724	0.7013	0.8444	0.851	0.9309	0.9272
Specificity	0.9023	0.9011	0.9496	0.7656	0.8311	0.6255	0.6614
Kappa	0.3363	0.3359	0.3327	0.4333	0.4822	0.4784	0.4686
AUC	0.8959333	0.8959	0.9058	0.8743113	0.8927807	0.9093191	0.9307



The second dataset was bank marketing which is a popular imbalanced class dataset, SMOTE data sampling method was used to correct the imbalance. Algorithm like xgboost, SVM, Decision Tree was used and pruned xgboost performed better than the two which is shown in the table above, with high AUC, accuracy values. Cross validation could improve the accuracy of SVM and Decision tree the models. It was not used in the study because of its high run time. The k-fold cross validation improved the accuracy of SVM and Decision Tree 0.8758102and 0.8390045 respectively. The xgboost had inbuilt cross validation to find best iteration so that was used. The decision trees were made to fully grow to study how pruning improves the model. Pruning in decision tree significantly increased the accuracy and AUC. Some learning curves doesn't give much insights due to lack of test and train size used in the study. The project mainly aims to get insights from the learning curve, figure out the bias and variance tradeoff, which models are underfitting and overfitting. The ROC curve of all the models is shown, xgboost as an edge over other algorithm due to its ensemble learning methods. The information gain was used in the decision tree over gini index as both gave similar results so, the tree was trained based on the information gain. The model could be improved using feature selection, more data preprocessing steps and proper parameter tunings in the