Machine Learning Assignment-II

Data Set 1

Data is taken from UCI machine learning repository and is collection of observations of different GPU runt times under different machine configurations

- There are total 14 features, of which first 10 are ordinal while last 4 are binary and total number of observations is 241600
- There are no missing values in data
- There is no need for scaling the data as different features are approximately in same range

Data Preparation for Modelling

- Average run time has been converted into categorical variable based on median value of given runtime thus transforming the problem into classification problem
- Complete dataset is divided into training(80%) and testing dataset(20%).
- Used Cross Validation and GridSearchCV for HyperParameter Tuning in Decision Trees and XGBoost respectively.

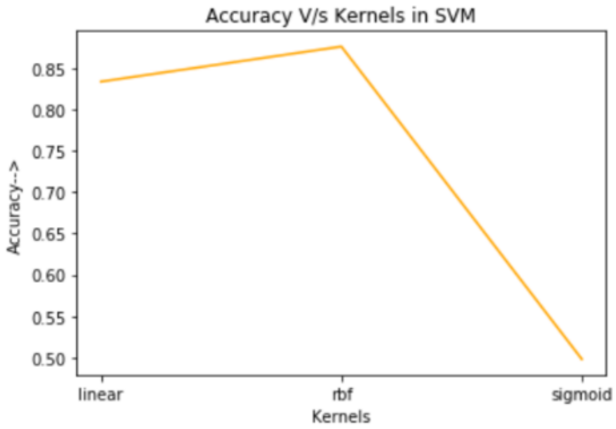
Model 1: Support Vector Machines

Experiment

- Classification of GPUs between high run time or low run time based on different system configuration using support vector classifier

Observations

- Due to Computational constraints, Only randomly selected 150000 datapoints were used to trained the SVM model on different Kernels.
- As support vector classifier supports different kernels, I used linear, rbf and sigmoid kernels to choose best model, below is the test accuracy obtained from models with different kernels:



Kernels	Validation Accuracy
Linear	0.8336
RBF	0.876
Sigmoid	0.4978

Confusion Matrix:

TN:1688	FP:173
FN:324	TP:1565

Conclusions

- As can be seen from above table rbf kernel has highest test accuracy for given data set, thus SVM model with RBF kernel is the final chosen model which has a **test accuracy of 87.6%**

Model 2: Decision Tree Classifier

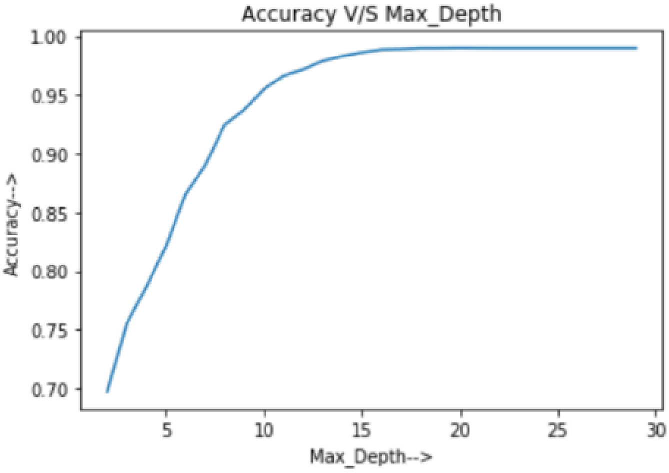
Experiment

- Classification of GPUs between high run time or low run time based on different system configuration using decision tree classifier
- Used Gini Impurity Criterion(Default in SKlearn) over Entropy because it is computationally more efficient as Gini Impurity uses Square function where as Entropy uses Log, which is more faster to calculate as compared to Log functions.
- Used 5-CV to check for Max-Depth as there is a possibility of over-fitting, and using a separate Validation set would reduce the size of training data, and using test set for obtaining hyperparameters would cause the problem of data-leakage. Therefore, to

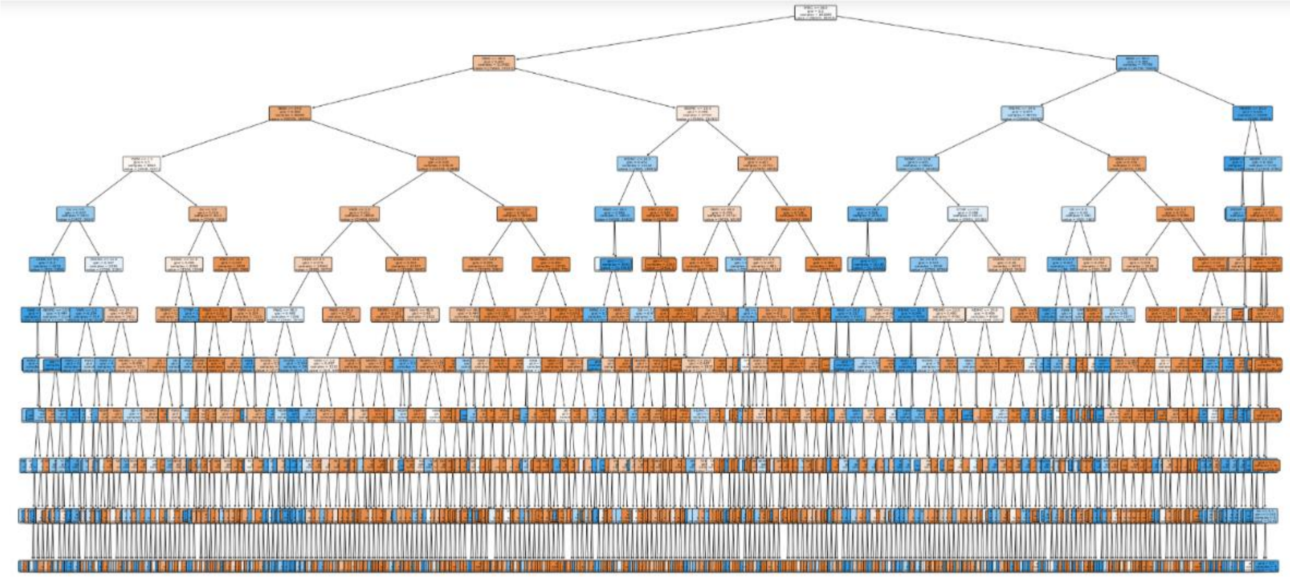
counter this problem,5 fold CV is used to obtain best Max-Depth keeping Accuracy score as performance metric.

Observations

- On training decision tree classifier with max-depth ranging from 2 to 30 following trend is observed, where depth above 11 was having same accuracy. Therefore, 11 was chosen as Max-Depth.

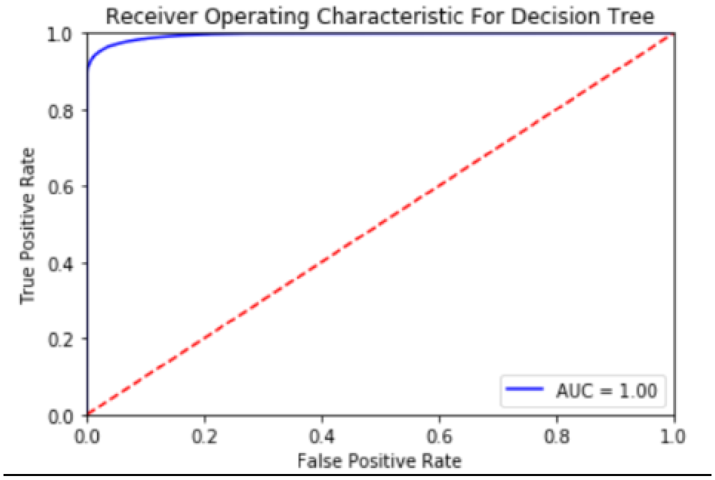


TREE:



ROC CURVE:

The ROC is showing a good lift, which implies that the model is much better than the dumb model(Red Line) and AUC=1 implies that the binary classification model has good measure for seperability.



Confusion Matrix:

TN:23670	FP:546
FN:1126	TP:22978

Conclusions

- Based on above chart we can conclude that max-depth of 11 is optimal for Decision-TreeClassifier for given dataset
- Thus, final Decision Tree Classifier model with max-depth of 11 has **Test accuracy of 0.9653.**

Model 3: Boosting

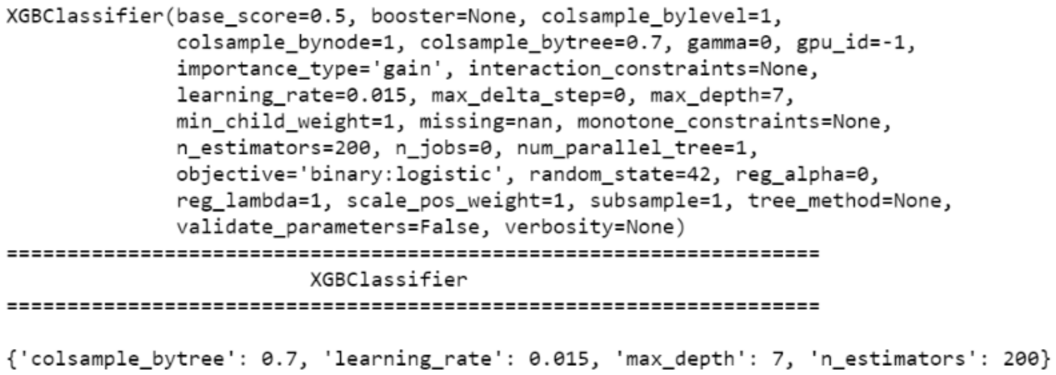
Experiment

- Try to improve accuracy of DecisionTreeClassifier by using boosting algorithm
- I have used XGBoostClassifier with DecisionTreeClassifier as base estimator for this experiment.
- Used Grid SearchCV to obtain ColSample_bytree,N_Estimators,Max_Depth,Learning_Rate.

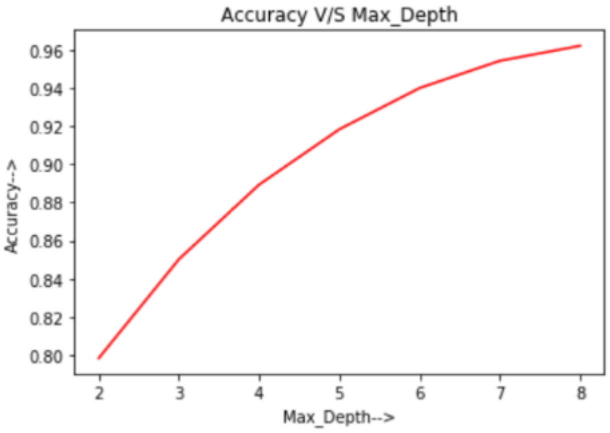
Observations

- Range of Values set for ColSample_ByTree- [0.55,7], N Estimators- [200], Max_Depth- [3,5,6,7], Learning rate-[0.01,0.015].
- Again used Cross Validation to check Max-Depth from range 2-9. The best Max-Depth came out to be 7.

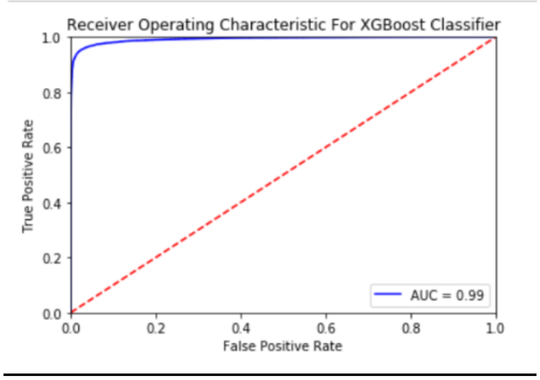
Result Of Grid Search:



```
XGBClassifier(base_score=0.5, booster=None, colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=0.7, gamma=0, gpu_id=-1,
              importance_type='gain', interaction_constraints=None,
              learning_rate=0.015, max_delta_step=0, max_depth=7,
              min_child_weight=1, missing=nan, monotone_constraints=None,
              n_estimators=200, n_jobs=0, num_parallel_tree=1,
              objective='binary:logistic', random_state=42, reg_alpha=0,
              reg_lambda=1, scale_pos_weight=1, subsample=1, tree_method=None,
              validate_parameters=False, verbosity=None)
===================================================================
                         XGBClassifier
===================================================================

{'colsample_bytree': 0.7, 'learning_rate': 0.015, 'max_depth': 7, 'n_estimators': 200}
```



ROC-AUC:



The ROC is showing a good lift, which implies that the model is much better than the dumb model(Red Line) and AUC=0.99 implies that the binary classification model has good measure for seperability.

Confusion Matrix:

TN:23701	FP:326
FN:1522	TP:22771

Conclusions

- Above trend suggests that max_dept of base estimator i.e. DecisionTreeClassifier beyond 7 is leading to overfitting
- Thus, final boosting model has number of estimators set to 200 while depth is set to 7, which has **test accuracy of 0.9617**

Final Conclusions for data set 1:

- Below is the test accuracy of final models of support vector machines, decision trees and ensemble methods

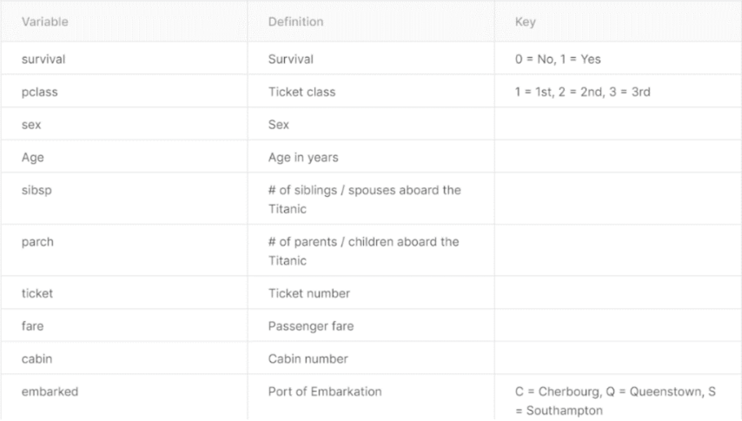
Models	Test Accuracy
Support Vector Machines	0.876
Decision Tree Classifier	0.9653
AdaBoost Classifier	0.9617

- Thus, looking at above table we can conclude, ensemble learning (XGBoost classification) and Decision Tree Classifier gives best results for dataset 1
- Also, as we trained support vector machine model on very small portion of data, the performance is still in comparison of other algorithms, thus suggesting it could be a better model when limited observation is available.

Data Set 2

The famous Titanic Dataset from Kaggle is used to classify Survival of the passengers who boarded the famous Titanic ship.

Features and relevant Information about the Dataset:



Variable	Definition	Key
survival	Survival	0 = No, 1 = Yes
pclass	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
sex	Sex	
Age	Age in years	
sibsp	# of siblings / spouses aboard the Titanic	
parch	# of parents / children aboard the Titanic	
ticket	Ticket number	
fare	Passenger fare	
cabin	Cabin number	
embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

Data Preparation for Modelling

- Checked for the percentage of missing data points and dropped "Cabin" as it has 77% missing values, and then imputed Age and Embarked with mean and mode values respectively.
- Feature Engineered a new feature by combining Parch and SibSp into a feature named Family Size.
- Dropped SibSP,Parch,Name features.
- One-Hot Encoded Embarked and Sex features and dropped the existing Embarked and Sex columns to avoid the exact collinearity problem in the dataset.
- Complete dataset is divided into training(70%) and testing(30%) dataset
- Cross validation has been done for selection of hyperparameter while test dataset would be used for checking the accuracy of final model.

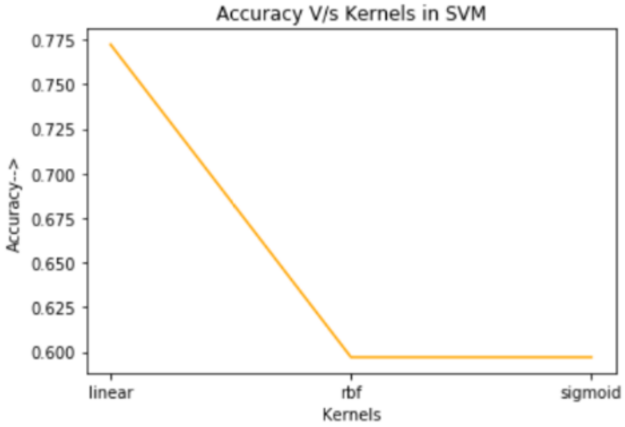
Model 1: Support Vector Machines

<u>Experiment</u>

TO calculate train and test accuracy while classifying the survival of the passengers.

<u>Observations</u>

- As support vector classifier supports different kernels, I used linear, rbf and sigmoid kernels to choose best model, below is the validation accuracy obtained from models with different kernels:



Kernels	Accuracy Score
RBF	0.7723
Linear	0.5970
Sigmoid	0.5970

<u>Conclusions</u>

- As can be seen from above table linear kernel has highest training accuracy for given data set, thus SVM model with linear kernel is the final chosen model which has a **<u>test accuracy of 0.7723</u>**

<u>Model 2: Decision Tree Classifier</u>

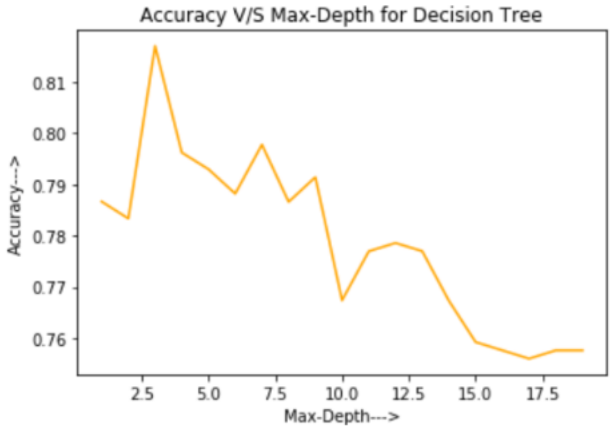
<u>Experiment</u>

- Used Gini Impurity Criterion(Default in SKlearn) over Entropy because it is computationally more efficient as Gini Impurity uses Square function where as Entropy uses Log, which is more faster to calculate as compared to Log functions.
- Used 10-CV to check for Max-Depth as there is a possibility of over-fitting, and using a separate Validation set would reduce the size of training data, and using test set for

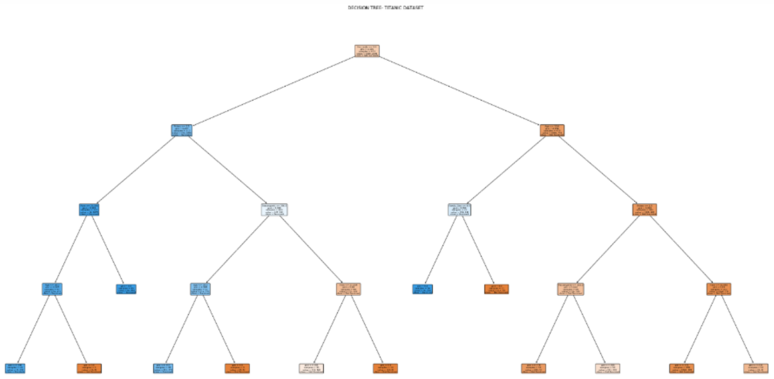
obtaining hyperparameters would cause the problem of data-leakage. Therefore, to counter this problem,10 fold CV is used to obtain best Max-Depth keeping Accuracy score as performance metric.

Observations

- On training decision tree classifier with max_depth ranging from 1 to 20 following trend is observed.

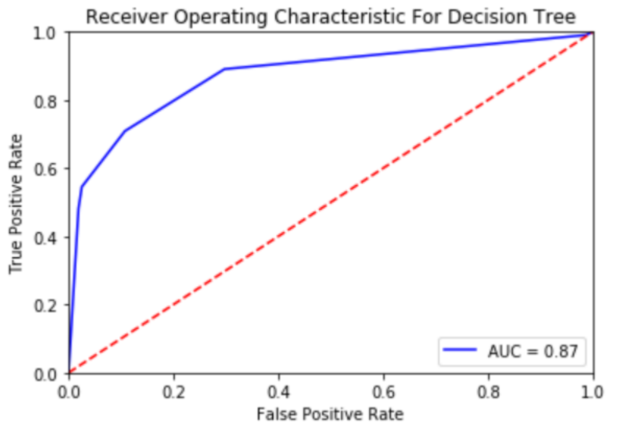


Tree:



ROC-AUC Curve:

The ROC is showing a good lift, which implies that the model is much better than the dumb model(RedLine) and AUC=0.87 implies that the binary classification model has good measure for seperability.



Confusion Matrix:

TN:141	FP:17
FN:32	TP:78

Conclusions

- Based on above chart we can conclude that max_depth of 3 is optimal for Decision TreeClassifier for given dataset
- Thus, final Decision TreeClassifier model with max_depth of 3 has **training accuracy of 83.46% and test accuracy of 81.71%**

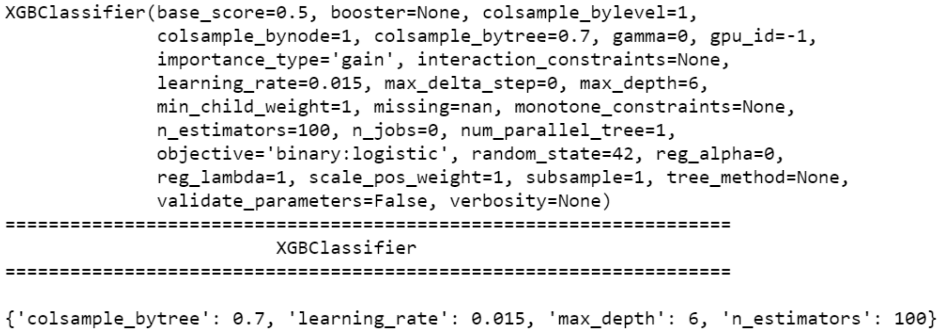
Model 3: Boosting

Experiment

- Trying to improve accuracy of DecisionTreeClassifier by using boosting algorithm
- I have used XGBBoostClassifier with DecisionTreeClassifier as base estimator for this experiment
- Used Grid SearchCV to obtain ColSample_bytree,N_Estimators,Max_Depth,Learning_Rate.
- Range of Values set for ColSample_ByTree- [0.55,0.6,7], N Estimators- [50,100,200], Max_Depth-[3,5,6,7], Learning rate-[0.01,0.015].
- Again used Cross Validation to check Max-Depth from range 1-9. The best Max-Depth came out to be 7.

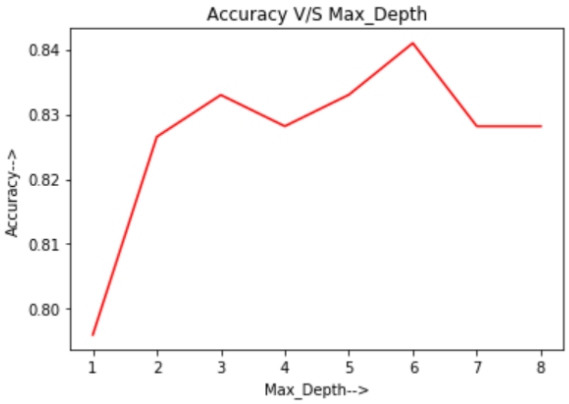
Observations

- **GridSearchCV:**



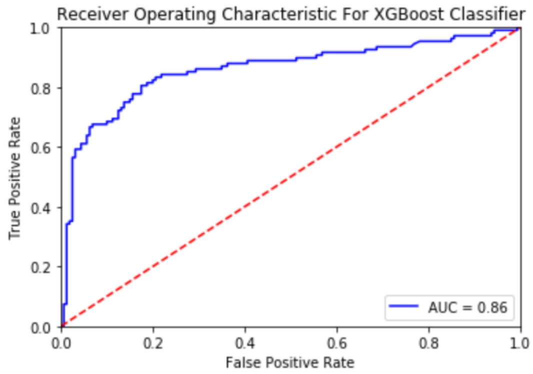
```
XGBClassifier(base_score=0.5, booster=None, colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=0.7, gamma=0, gpu_id=-1,
              importance_type='gain', interaction_constraints=None,
              learning_rate=0.015, max_delta_step=0, max_depth=6,
              min_child_weight=1, missing=nan, monotone_constraints=None,
              n_estimators=100, n_jobs=0, num_parallel_tree=1,
              objective='binary:logistic', random_state=42, reg_alpha=0,
              reg_lambda=1, scale_pos_weight=1, subsample=1, tree_method=None,
              validate_parameters=False, verbosity=None)
===================================================================
                         XGBClassifier
===================================================================

{'colsample_bytree': 0.7, 'learning_rate': 0.015, 'max_depth': 6, 'n_estimators': 100}
```

- **CrossValidation:**



ROC-AUC Curve:

The ROC is showing a good lift, which implies that the model is much better than the dumb model(RedLine) and AUC=0.86 implies that the binary classification model has good measure for seperability.



Confusion Matrix:

TN:144	FP:16
FN:34	TP:74

Conclusions

- Above trend suggests that max_dept of base estimator i.e. DecisionTreeClassifier beyond 6 is leading to overfitting
- Thus, final boosting model has number of estimators set to 40 while depth of DecisionTreeClassifier is set to 6, which has **training accuracy of 90.85% and test accuracy of 81.34%**

Final Conclusions for Data Set 2:

- Below is the test accuracy of final models of support vector machines, decision trees and ensemble methods

Models	Test Accuracy
Support Vector Machines	0.7723
Decision Tree Classifier	0.8171
AdaBoost Classifier	0.8134

- Thus, looking at above table we can conclude, Decision Tree Classifier and ensemble learning (XGBClassifier) gives best results for dataset 2.