Machine Learning – Project 1 (Supervised learning)

Name: Harsha Mandadi (hmandadai@gatech.edu)

Datset1 –

For my first Dataset, I am considering a balanced classification data – Banknote authentication. The problem statement is to classify the data for the evaluation of authentication of banknotes. The data for features were extracted from images taken from a pool of forged and original banknotes. We use 4 different kind of features. Variance of wavelet transformed image, skewness of wavelet transformed image, curtosis of image, entropy of image. The dataset has 45% classified as 0 and 55% classified as 1. Since this looks like a balanced dataset, I have decided to use accuracy as my performance metric instead of F1 score.

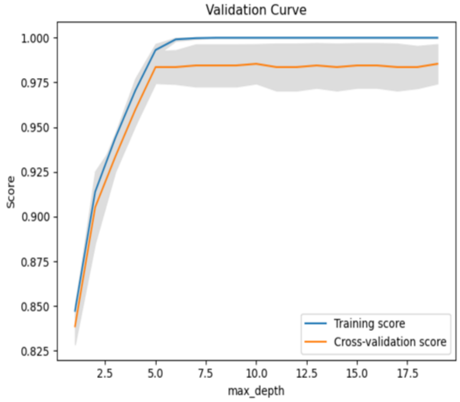
Decision Tree

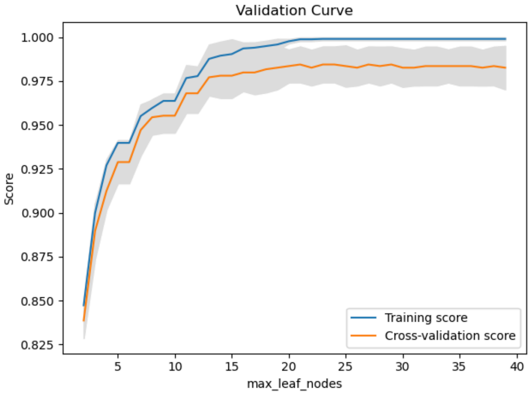
Parameters that can be tuned : max_depth, min_sample_split, min_sample_leaf, max_features,min_impurity_split,min_impurity_decrease,max_leaf_nodes. I have choosen Gini as a metric for splitting the data at each node. Since the number of features is only 4, there will not a problem of overfitting w.r.t features. Hence I did not use max_features for tuning. As the range of min_sample_split increased(1-40), the score increased in both training and test data. Same is the case with min_samples_leaf. Upon careful examination and tuning all pparameters, I have decided to tune max_depth and maximum leaf nodes.

Model complexity curves for each parameter:

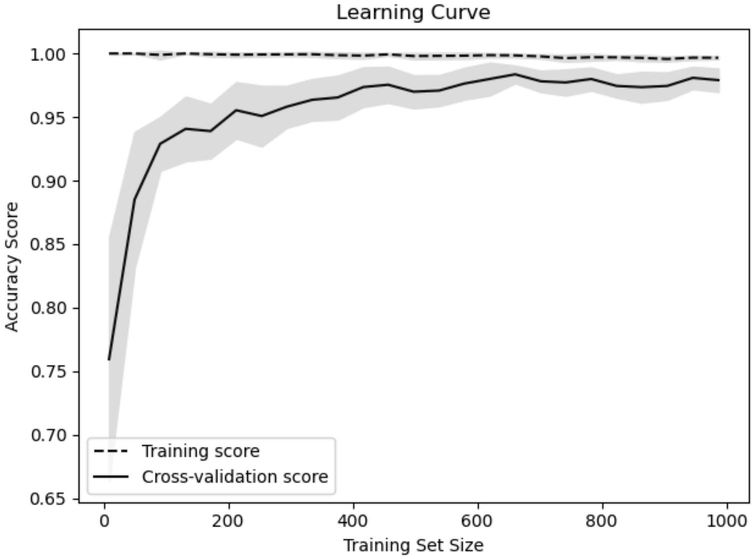
Maximum depth of a tree: As the maximum depth of the tree increased, the depper the tree grows increasing the complexity and trying to fit the training data. After max depth beyond 5, the training and validation score seems to have reached saturation and never converge. The gap between these 2 score implies there is a overfitting problem beyond that point. Also CV score reaches its maximum at 5. At x=4, the accuracy for cv is around 97% which is good. So for further pruning, I will consider the max_depth range between (4 to 6).

Max_leaf_nodes: The graph tends to overfit the model after max leaf nodes reaches beyond 20 as training score reaches 1. Till nodes=5, the test and train score seems to converge but they have a very low score. This implying underfitting. The ideal range for this parameter would be from 6-20, although the data looks very noisy in the graphs.



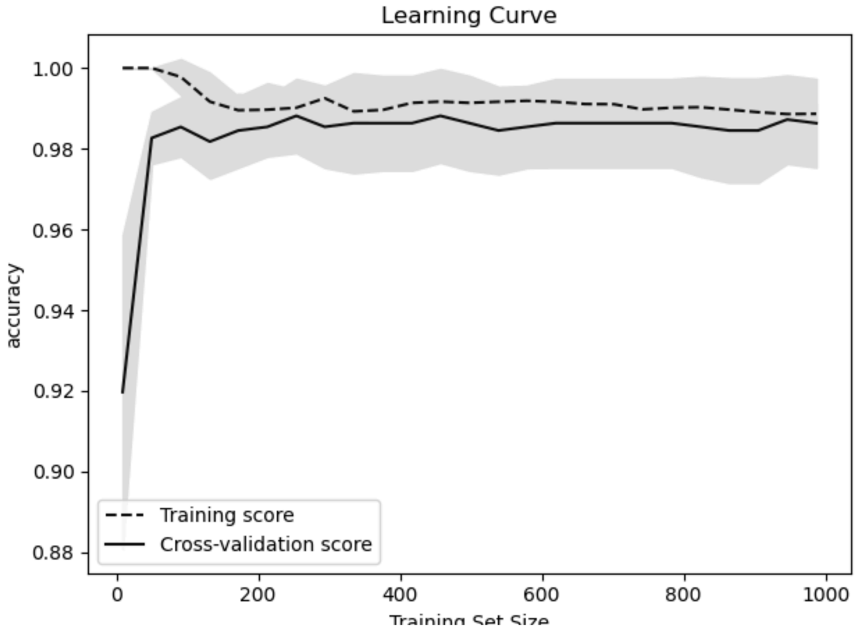


Learning curves for the parameters: max_depth = 6 and max_leaf_nodes = 20



Support Vector Machine:

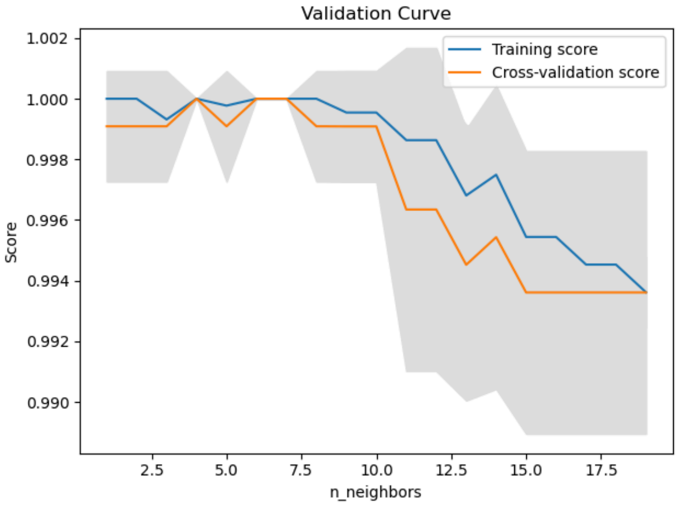
Linear kernel: I have used Linear kernel for this dataset and looks like the data converges both the training and test data on linear kernel. Hence it would be enough to have a linear kernel instead of a polynomial kernel to classify the data. When the training data was very low, we found that the data is underfitting but as we supply more data, the kernel was able to correctly eliminate the unwanted datapoints and only use necessary datapoints properly



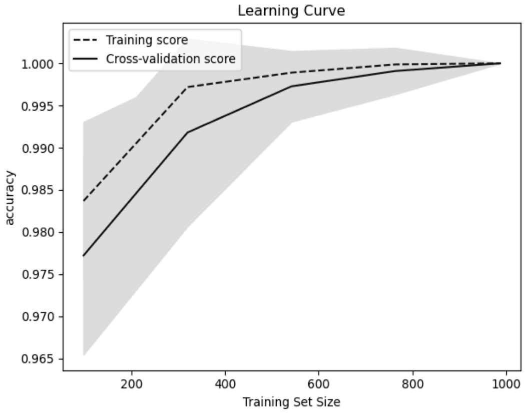
KNN – Since we only have 5 features, I assume that all features will be given equal importance and hence KNN should work well

Parameters to be tuned: k,leaf_size

Model complexity curves for each parameter k (6-7) . When we have K around 6 to 7, the algorithms seems to have learned well. Any value above k, will need more neighbors to adhere to the rule and hence is performing badly on both test and train data. This is causing more bias and hence k needs to be reduced.



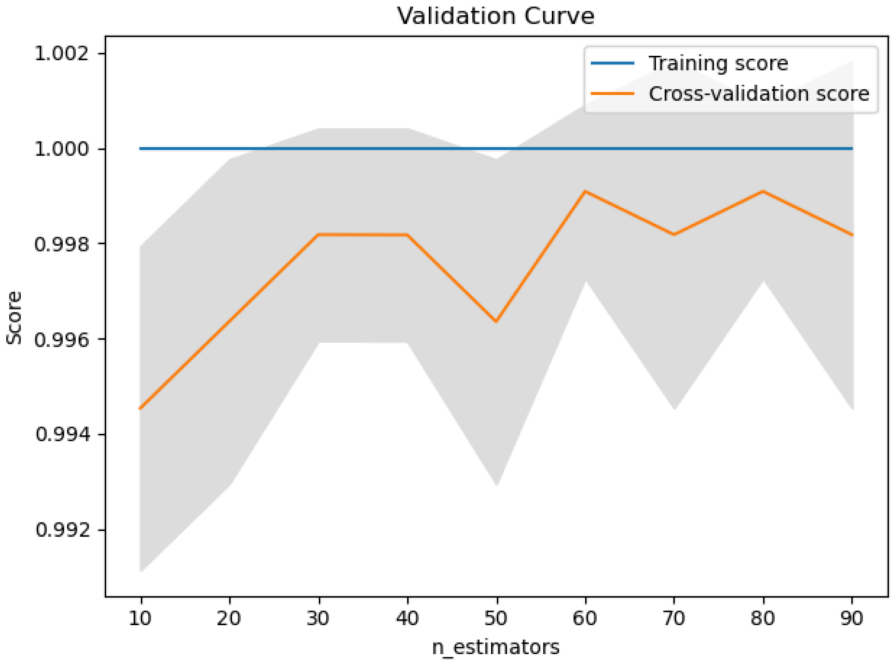
Learning curves for the parameters, k=6

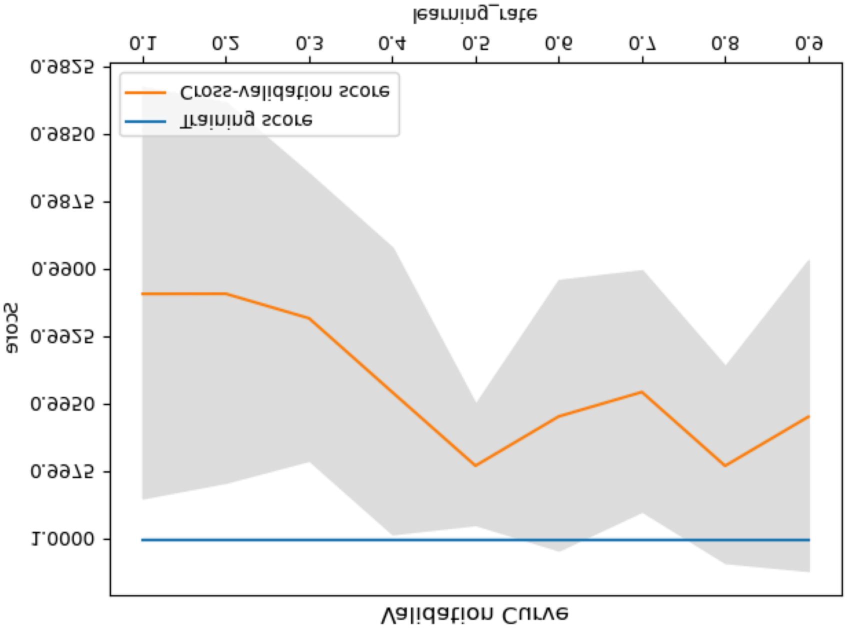


Boosting

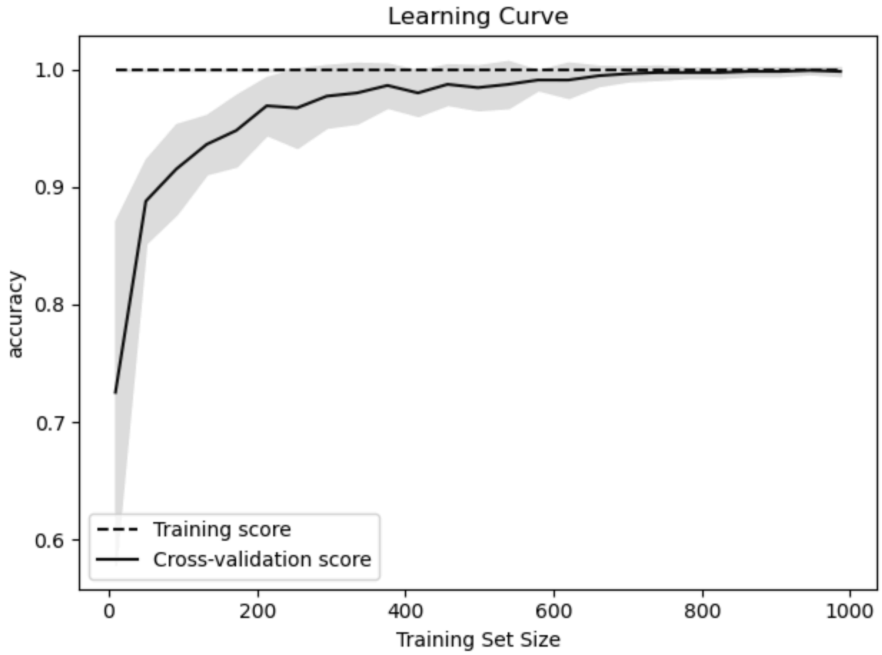
Parameters that can be tuned: max_depth

N_estimators:





Model complexity curves for each parameter:



Dataset2

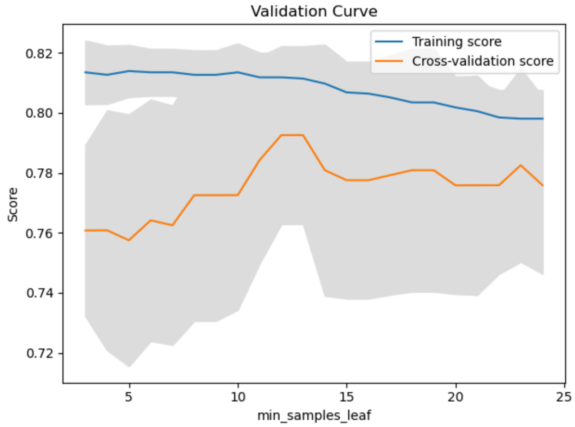
For my second dataset, I am using a dataset which classifies if a person has transfused blood in the past or not based on 5 features. All the features used are continuous variables. The data has 23% classified as

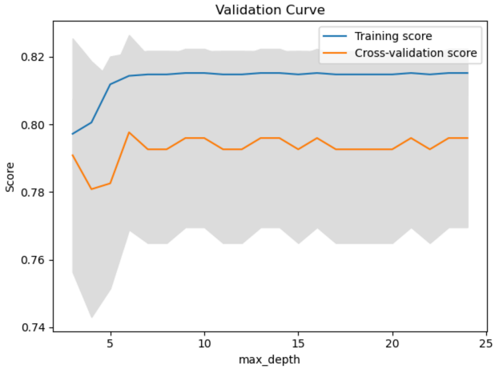
1 and 77% classified as 0. Since this looks like an imbalanced dataset, I have decided to go with F1 score performance metric.

DecisionTree:

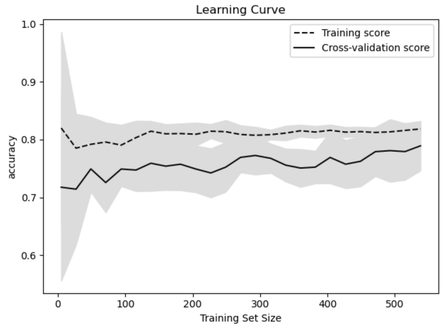
1.min_sample_leaf: By increasing the minimum samples needed at the leaf node for splitting, we are avoiding the leaf nodes to be split into more complex nodes which ca cause the model to overfit to the train data. We see the overfitting in our data after 12. Untill 12, the algorithm tries to learn from the train data with steady increase in f1 score.

2. max_depth: From the graph below, the ideal tree depth is at 6. After that point the score is very noisy.If we observe closely, the validation score decreases from depth 2 to depth 4 instead of increasing as the model tries to learn. This might be because of the underfitting of data upto point where it generalizes properly.





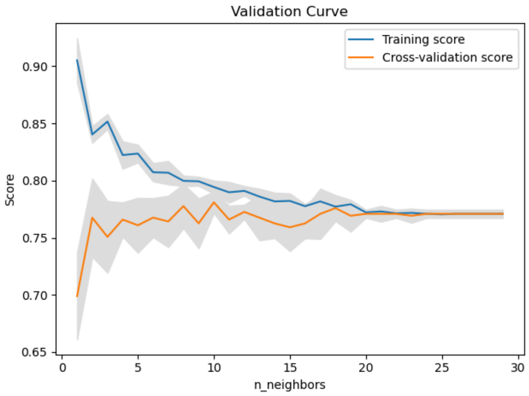
3. Learning curve for parameters max_depth = 6, min_samples_leaf = 12. The learning curve below doenot converge. Although the training and testing scores are not above 90%, as the training size increases, model performs doesn't look like increasing. For this data to work well for decision tree algorithm, I assume we need more data to be collected. The decision tree algorithm model lacks from being too generic(high bias)



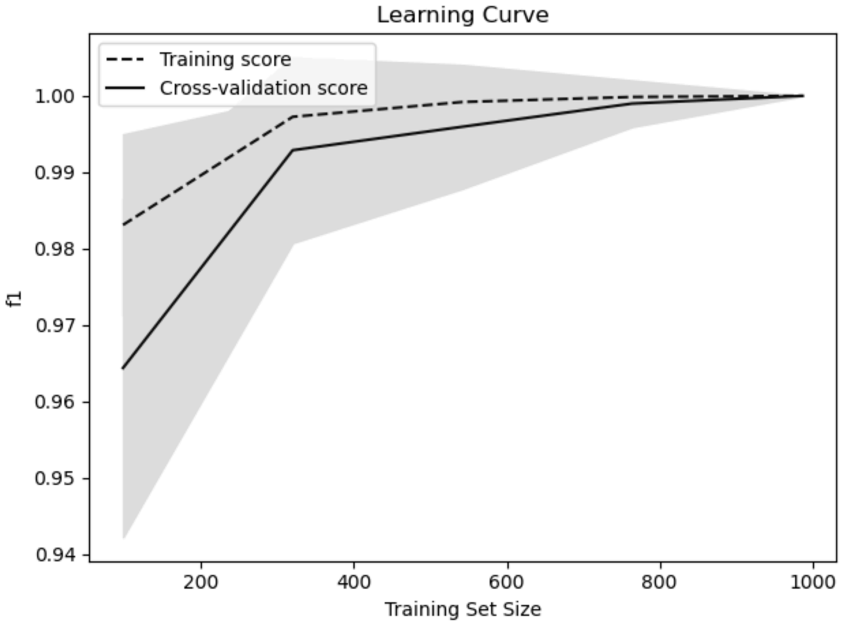
KNN: KNN works best for this sample of data among all other algorithms

Parameters: k,leaf_size

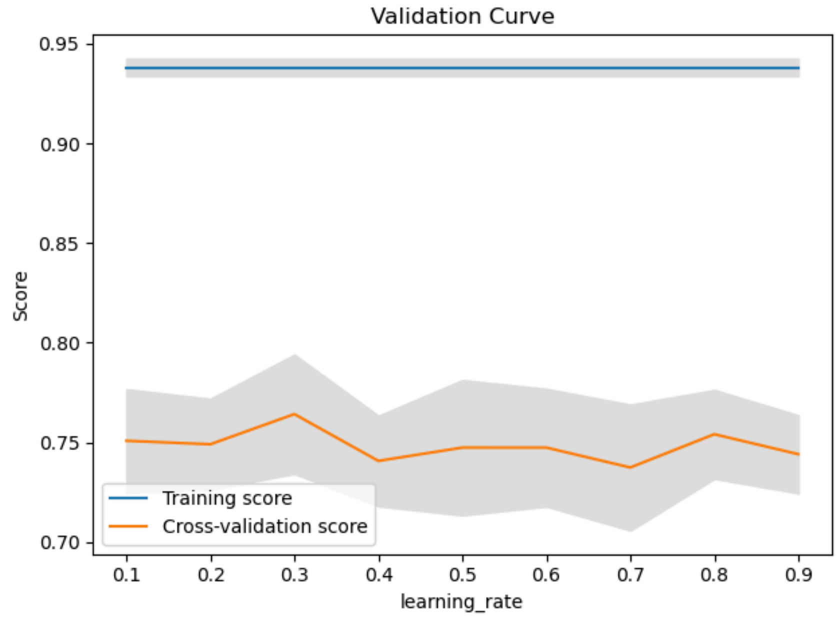
Model complexity curves for each parameter

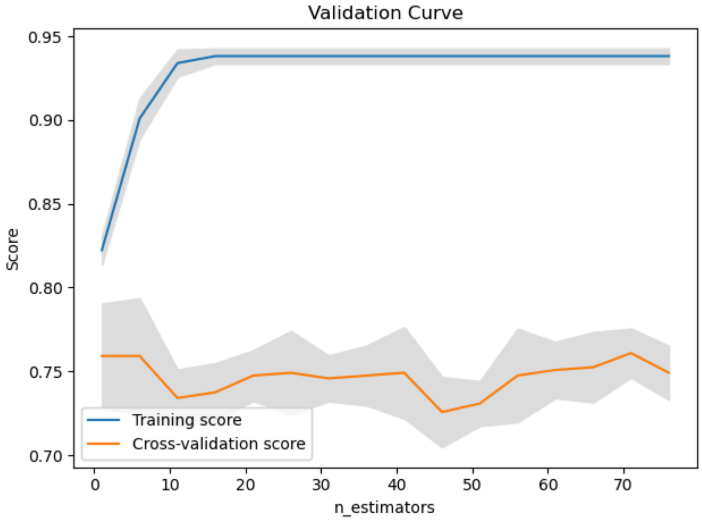


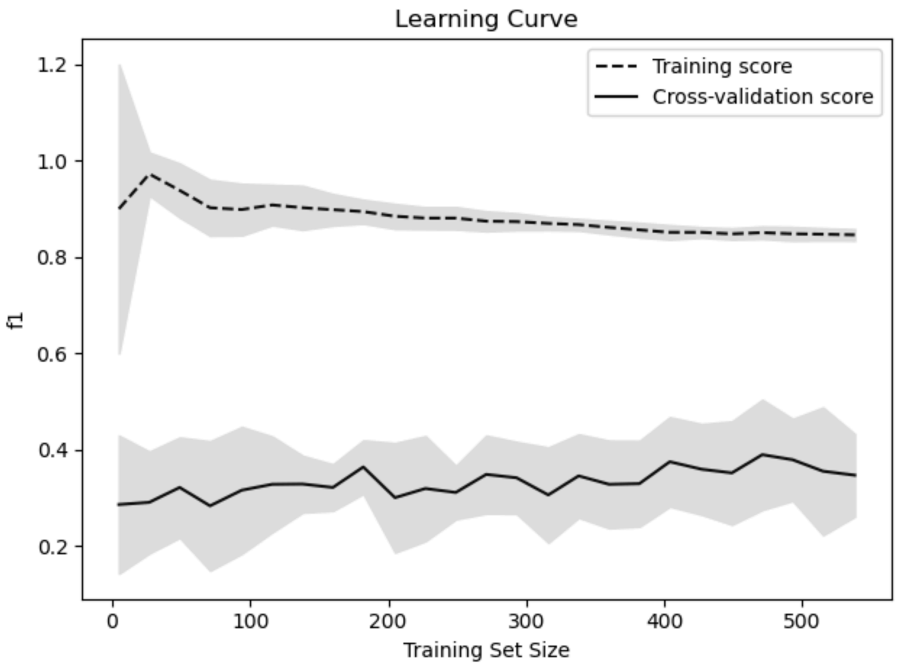
Learning curves for the parameters, k=20,leaf_size=30



3. Boosting: Parameters to be tuned=n_estimators and learning rate. Boosting is a very bad algorithm for this dataset. No amount of tuning on all the hyperparameter made any difference to the learning curve. Boosting algorithm leads to a very high bias model for this data


Validation Curve
Score
0.95
0.90
0.85
0.80
0.75
0.70
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
learning_rate
Training score
Cross-validation score


Validation Curve
Score
0.95
0.90
0.85
0.80
0.75
0.70
0
10
20
30
40
50
60
70
n_estimators
Training score
Cross-validation score



3. References

1. Volker Lohweg. and Helene DÃ¶rksen. (2012). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of Applied Sciences.

2. Yeh, I-Cheng, Yang, King-Jang, and Ting, Tao-Ming, "Knowledge discovery on RFM model using Bernoulli sequence, "Expert Systems with Applications, 2008,