CS7641 Assignment 1

Sahil Soni -ssoni41@gatech.edu

CS7641-Assignment 1

CS7641 Machine learning , Learning Curve, Metrics, Model Complexity , Hyper Parameter Tuning

Abstract: CS7641 Machine learning analyzing Decision trees,Neural networks,Boosting,SVM,KNN algorithms .

Index Terms: CS7641 Machine learning,Decision trees,Neural networks,Boosting,SVM,KNN, Bias, Variance,Under-fit , Model-overfit, Imbalanced Data Set,ROC, Precision , Recall , F1 Metrics

1. Introduction

That first assignment is part of Spring 2020 OMSCS CS7641(Machine Learning). In this assignment, five Supervised Machine learning algorithms need to be understanded and explained in detail. We're going to apply machine learning algorithms, Decision Tree, Neural Network, Boosting, Supporting Vector machines and K-Nearest Neighbor. We must apply this on two INTERESTING data sets as part of the task.

2. Data Sets

For this assignment, I finalized Two Data Sets Wine Data Set and Heart Data Set after a lot of thoughts and quest. First set of data is Wine dataset. It has 12 properties, and instances of 4898. We have to estimate the wine's Production Quality Score (0-10) using its 11 input variables.

The second set of data is the set of data for heart disease. This data set initially contains 76 attributes but only 14 of them have been used as part of various online competitions such as Kaggle Platforms and UCI Original Data Source. The reason they provide is because "this database contains 76 attributes, but all the experiments published refer to the use of a subset of 14.". We will estimate the incidence of heart disease 0 (no involvement) to 4(highly present) in this data set. This set of data has about 303 attributes as small as the Wine Data Set.

3. Why Data Sets Interesting

Choosing the Interesting Data Sets is one of the requirements as part of Assignment. I think my two sets of data are very interesting. That's why I think so.

First of all, the Wine Data set, good quality wine is very important because people are paying a lot for good quality wine. Good quality wine has a good taste. Good second-rate wine is key to good health. Nobody wants to drink wine of poor quality and get intoxicated or stomach bad.Even many scientific research papers daily associate a moderate drinking of red wine with good long life. I am also a wine enthusiast who tries different wines every week and helps me to understand what makes a good quality wine by selecting the wine data set. I will buy good quality wine from my local wine shop hopefully using this software.

My second set of data is set of heart data, as we all know how important it is to core. Heart disease has been undetected most of the time and people develop Strokes. That's why they say he's a silent murderer. Previous detection of heart disease can save many lives and help humanity. So I have selected this as my second set of data.

Such two data sets have helped me to understand the principles of machine learning and different algorithms in machine learning. The explanation for both is special, and has distinct features.

Wine data set is unbalanced, although there are no missing values or Outliers. But some of its target values are distributed unequally by Ratio 1:99. It is also a set of mid-size data that helps me to understand the calculation time of different algorithms as data size increases. In addition, how to choose the right hyper parameters Technique(Grid Search / Random and third-party tools e.g. sklearn) as data size increases.It's the data set I've tried to use different ROC, Confusion Matrix, Precision and Recall metrics due to imbalance of data sets.

Heart Data set is a structured set of data, and is small in size. The goal values are 95% distributed equally. So I can play with and understand different algorithms. Also its easier to test most of parameters of various algorithms and understand them well.Even in older days its being said that with less data we can't learn anything from Machine learning models.So I want to demonstrate in days to come, even with less training sets machine learning models can learn.

4. Data Acquisition and Tools used

I obtained my datasets from UCI computer learning platform and Kaggle. I used Python and Sklearn Framework for deployment. Initially I used the Google Colab Notebook as easy to use and I was able to run a lot of experiments on a Ram 12 GB GPU machine. Yet instead, as mentioned by class teachers on Piazza, I have migrated them to jupyter notebook .

5. Metric and Methodology

I treat it as a classification issue for Wine data set so I used accuracy metrics for that. Reason I treat this problem of classification because we have to predict wine quality between classes (1-10). This is a multi-class problem of grouping. It is also precision metric that can tell how accurate our prediction is for wine quality.Although we have used reliable measures, but since this data set is imbalanced on the test it can forecast incorrect values for less attribute classes (Distribution) question. To better understand this, we need to use imbalanced dataset techniques below :-

1) Tried utilizing ROC, Confusion Index, Accuracy and Recall due to imbalanced data sets. 2)We should try re-sampling or Synthetic Samples 3) Increasing weights will penalize Systems like SVM.

For heart data set, I initially tried to treat it as a regression problem, contrary to general belief. So I chose MSE (Mean Squared Error) Metric initially. My goal was to understand and distinguish between two methods. But it makes sense to use it as a grouping measure after many studies.Since we have to differentiate between False Positive and True Negative, which is necessary for such issues. A Roc Curve will help you to better understand the data. So finally opted for accuracy metrics. If we add more data to the data set later and have unbalanced data set, we may not be able to use Classification Metrics.

6. Train/Test/Split and Cross Validation Data Sets

To test our model, we usually rely on more data to test the behavior of our model. Since in our case we don't have more data, we divide data into Train Set and Tet Sets. We train our model on train set and test on test tests its predication. But if our Test Data continues to go back in this way, it will also overfit.We are therefore constructing train, testing and Validation data sets. On train system we exercise our model and continue making predictions on validity stage. Until our model is finished, we never touch our test set.

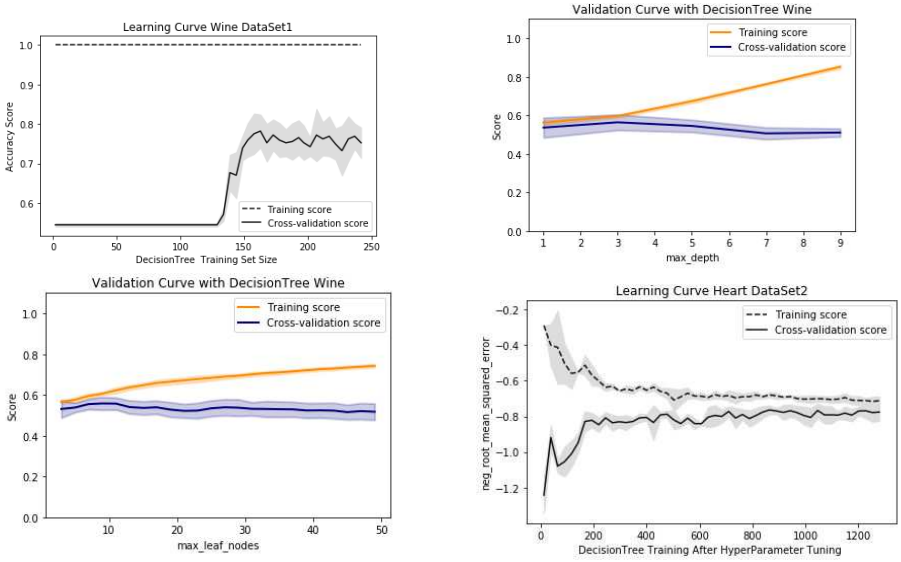

Fig. 1. Decision trees-Wine Data Set1

Even where we can divide our data sets into k folds, we can use cross validation technique. We used k-n system to train our model and rest sets that we were checking with. Notice here whenever we need to randomly allocate our data sets on k-sets otherwise those sets will be overfit.I used k-fold with random distribution from sklearn kit in my assignment on all algorithms.

7. Decision trees(Dts)

Dts are supervised system of learning tools that can be used for data sets of regression and classification. Dts classifier is taught from important Data Sets properties and using If-Else methods such as a Tree Manner. It is one of the most powerful algorithms that can be applied to learn important data set features and can be used on more complex algorithms.We can use it for multi-output problems and numerical and categorical ones.

Dts are shaped as discrete arboreal framework. From the tree's root node(attribute) based on if-else state, it chose the next important Data Set Attribute. Based upon max depth parameter it chose how deep is the Tree. More Deeper the the more complex is to chose the rules to fit the model.Then how we pruned the tree, i.e. where we no longer decide to terminate the tree's leaf node and tree, is also important. Otherwise it will continue to split on the leaf nodes for larger data. Sklearn used the Optimized Cart algorithm by example.

7.1. Analysis -Wine Data set1

We displayed Wine Data Set with learning curve on Fig1 along with validation curves and finally learning curve after hyper parameter tuning . As we can see, on learning Curve Data Set1 we have a flat accuracy learning curve of 99% , which initially implies that our model does not learn anything from this dataset.Large gap between training(accuracy score around 99%) and validation curve (accuracy around 30%) means we have high variance. So our model is over fitting . More training sets can also be added to reduce the variance.But this is not an option in our case. We can try also more complex algorithms like SVM/Neural Network to reduce over-fitting . We can also drop some of features which are not important . We can decreased the regulation to decreasing the variance and increasing the biased.

We have used Pruning in our model to solve the high variance problem . Also as part of complexity of the model, we must adjust various techniques such as Grid Hyper Parameter Tuning,

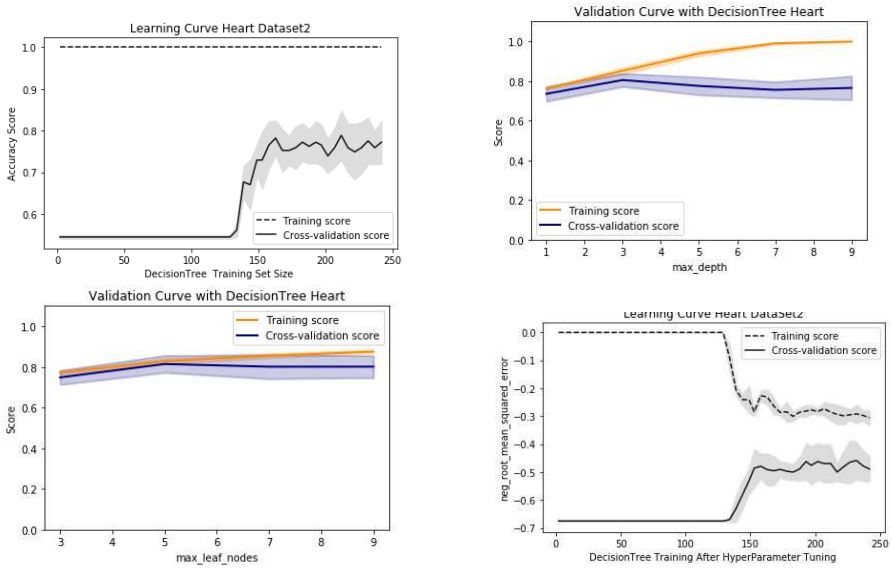

Fig. 2. Decision trees-Heart Data Set2

Random Parameter Tuning and Sklearn Validation Curve to solve high variance problem and to improved score of the model .

From fig 1, we can see on Validation Curve that there is a small gap (low variance) with the parameter ”max leaf nodes” under 4. Although the score for accuracy is about 0.58, it’s fine. We are good so long as we have small variance and our test range will suit.

Next , In Fig 1 on Validation Curve to demonstrate Tree pruning and to change another ”max depth” hyper parameter. This parameter will determine where the tree should be stopped or how deeper it is. We have low biased, as we can see with about max depth 3 and after that gap between Training and Cross validation curve increased. After that, that indicates he has a strong prejudice.

We used Grid Search hyper parameter tuning technique for tuning parameters (criterion, ccp alpha, max leaf nodes, min samples leaf, max depth, min samples
split) We find after best hyper parameter tuning, around 100 data points training score initially high around 80% and validation score around 45% , which is expected as our model has not learnt anything with 100 data points. But after 100 data points training score start decreasing and validation score start increasing until 700 training sets. Now less gap between training and validation curve throughout means we have find our ideal model. After that, accuracy is almost constant and there is low variance . This is the final curve that we want with a low score with skewed and reasonable precision. We used below hyper parameter for this :- criterion=’gini’,min__=50,max__=5,max_=1,min__=100

Finally, what it means is that the addition of new data will not help in precision after 600 training sets. If we need to improve accuracy, more features need to be added but also taken in picture it should not over fit .Its also proved that that decision tree fits well with imbalanced data, because otherwise even the less distributed class logic fits well.Also they are tool to find features among data sets later to be used on other models .

7.2. Analysis -Heart Data set2

From fig2 from the learning curve, we can see that likely to wine dataset heart dataset did not learn anything at first and it has a flat training curve of 99% precision. Because of the wider gap between training and validity curve, it also suffers from over fitting issue as it has large variance

problem.However,similar to wine data set with DT Pruning and after hyper parameter tuning we see model has improved. Around 120 dataset model start learning and accuracy decreases with training curve moving towards validation curve . Now about 220 data points validation curve also move upward to training curve until both curves stabilized .Following 220 data sets, training and validation curves have less gap between them . so finally model still suffering from high variance and over fitting . So, this demonstrates that data models can learn and forecast well even with less training sets .

Finally we have used below hyper parameter for this:- criterion='gini',min__=1,max__=17,max_=5,min__=11

8. Analysis -Neural Network(NN)

The sklearn Neural Network (MLP) that we introduce here is a supervised learning algorithm that can be used in machine learning models for both classification and regression tasks. In this, given a X number of features (instances), non-linear functions can be learned to predict the Target (y) output. It has one more non-linear layers called hidden layers compared to logistic regression.Input layer consists of a set of functions, called neurons. They feed into hidden layers based on weights, followed by non-linear activation function. Later production obtained values from the last layer which finally gives the Target value y.

Analysis NN -Wine Data set1 Note neural network is a very heavy model compared to the Decision tree and its recommended to run on GPU for larger data sets. From fig 3 , learning curve we can see that as expected(Because model did not learn anything with less data initially) initially larger gap between training and validation curve but later gap start reducing between them as more data feed into model . Around 1200 data sets both curved almost merged . This less gap between training and validation curve means it has high biased and our model is under fitting . Compared to Dts now our problem has been shifted from over fitting to under fitting .

So adding more training sets will not help in this case as model stop learning after 1200 data sets. We can decreased the weight or reduce the regularization or reduce loss functions to solve under-fitting in this case.We have solved under fitting by increasing hidden layers in our case . As part of model complexity we have also tuned two hyper parameters to improved accuracy and reducing high biased after 1200 data points.

First parameter we chose to tune is alpha . This parameter is L2 penalty(regularization) in our case.L2 regularization will add a cost with regards to the squared value of the parameters.Because we have to reduce weights to improved high biased . By default alpha on sklearn is 0.0001. From figure 3 , validation curve we can see by changing to other values has less impact on the biased (Gap is already low between training and validation curve). We tried with different random parameters but that did not help much to improved high biased problem .

Second parameter we optimized is solver(weight optimization) . We have tried different weight optimization techniques (lbfgs,sgd,adam) to see if they help in reduce biased. As we can see on fig 3 validation curve , "adam" optimization helped to reduced biased as gap between training and validation curve increased towards Adam . So we chose Adam as our optimization parameter.

As part of final hyper parameter , we raised the hidden layers from 50 to 100(more complicated network to solve under fitting issue) , used "adam" weight optimization . As we can see after tuning the hyper parameter our model is no longer under fit and we have a nice balance between variance and bias.We have finally used below hyper parameter for this:- activation='logistic', alpha=0.0001, hidden__=(100, 100, 100), learning_='adaptive', solver='lbfgs'

8.1. Analysis NN -Heart Data Set2

In Heart data set , we can see in figure 4 even without any parameter tuning, NN networks start learning and have less variation after 220 data sets for heart data set in Neural network. Similar to the wine data set for NN, the heart data set for NN also suffers from under fitting problem after 200 data sets. Here also also to overcome the under fitting problem we added more hidden layers to our network .With hyper parameter tuning, after 150 data sets, we can see model start

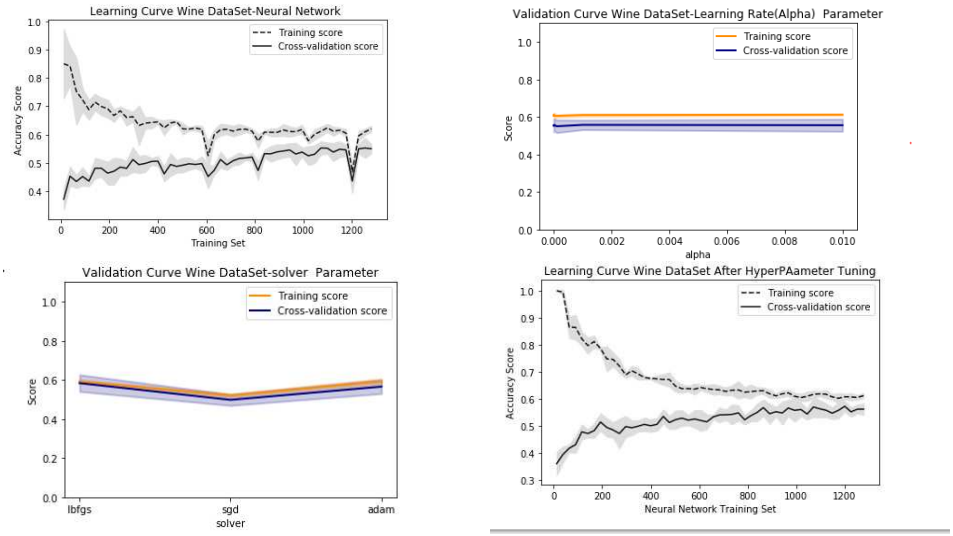

Fig. 3. Neural Network-Wine Data Set1

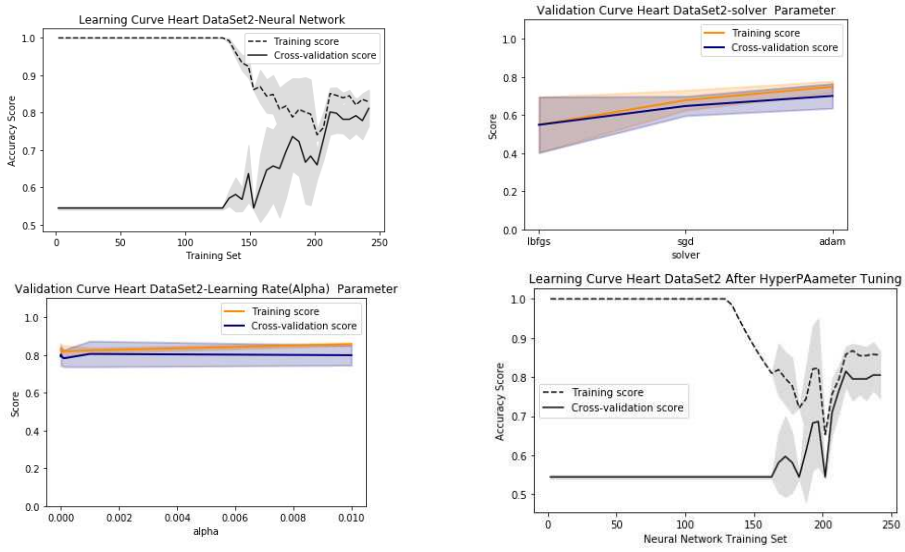

Fig. 4. Neural Network-Heart Data Set2

learning and at last have low bias after 220 data sets.Now training and validation curve are not merging alike before and has less gap between them .

Compute time of Neural netowrk is high with sklearn as they do not used GPU . Its almost as comparable and greater than SVM if more hidden laters are used.

Also to overcome under fitting we try to feed our model 10 data points compared to 50 before . We see interesting results in our data sets and it helped to reduced biased. So if our model is under fitting we can tried to feed it less data . So sometimes, how much you feed your model data and in what sets you can help to reduce bias.Same true for over fitting , if model is over fitting we tried to feed it more data points of 100 and 200 to see if it helped on over fitting. We have used below hyper parameter for this:- activation='logistic', alpha=0.0002, hidden__=(50, 50, 50),learning_='adaptive', solver='adam',learning__=0.002

9. Boosting

For this we used the Adaboost classifier. The main idea of this type of classifier is to combine different model prediction to improve the final target classifier for predication.

9.1. Boosting Wine Data set1

From the Fig(5) similar to neural network we can see initially that Adaboost begins learning well, but later gap between training and validation curve almost merged . After 800 data sets, the question of high variance had moved to high bias concern.

ADABOOST's default estimator is a decision tree.So we will used pruning here to reduce the high biased here .Remember to solved over fitting in Dts we have decreased the max_.Here By default adaboost used max_parameter 1 . So we can simply increased max_parameter to reduced biased as part of pruning in Ada boost. Here after hyper parameter tuning using random search , we have increased the max_value to 3.

Also we tune its learning rate parameter and n . N is identical with DTs.It is "the maximum number of estimators at which boosting is terminated".So we can reduced n from default value from 50 . Second parameter we tuned is Learning rate parameter, tells us how to used estimate of Other classifier. More the learning rate more weights it gives to classifier and less the learning rate less weights it gives to Classifiers. Learning rate parameter tells how Other classifier estimate is used. More the learning rate gives the classifier more weights and less the learning rate gives the classifier less weights. We can reduced the default learning rate from (1.) to reduced biased.

Finally we have plotting validation curve for n and Learning rate. Based upon plots we have used n value of 8 and learning rate of 0.0001 along with max_value to 3 in final model . So finally after tuning hyper parameter we have reduced the biased and able to solved the under fitting problem . This has been shown on figure 5 on final curve of Boosting .We have improved accuracy of model and able to remove its curse of over fitting.We have used below hyper parameter for this:- max_=3,n_=8,learning_=0.0001

9.2. Boosting Heart Data set 2

From figure 6, we can see that the base model of the Heart data set has high variances. Larger gap between training and validation curve. So this is high variance over fitting problem . Contrary to wine data set this is high variance problem .So we will increased the regularization here .

We have plotting validation curves for curves for n and learning rate . Validation curve does not give us much idea here. So we have used random search based upon our knowledge and followed by Grid Search CV. We have with max_3 , n_6 and learning_0.003 we have improved the over all model accuracy and reduce the variance of model . Finally with this we are able to solve over fitting problem of model .We have used below hyper parameter for this:- max_=3,n_=6,learning_=0.003,random_=5

10. SVM

Support Vector Machines (SVM) used for Classification , Regression and Outliners Data . It has higher run time compared to Other algorithms .

SVM Wine Data set1 As seen in figure 7, initially there is a large variance initially between training and validation curves. Why we're not expecting model to know with less data points initially. But after 350 data points we can see later on learning curve there is a low variance between training and validation curve. As training and cross validation accuracy are nearly the same and we have high biased. So this is under fitting problem .

SVM is a complex model so its one of tool to reduce over-fitting if another simpler algorithms like (Decision tree/K neighbour) is suffer from over fitting.But now model is under fitting .

As shown in the figure 7, part of validation curve parameter "C" has not helped much . We have also tried different kernels ('linear', 'poly', 'rbf', 'sigmoid') as part of validation curve .We noticed validation curve of kernel did not offer much help to tell which kernel we used. Now our model

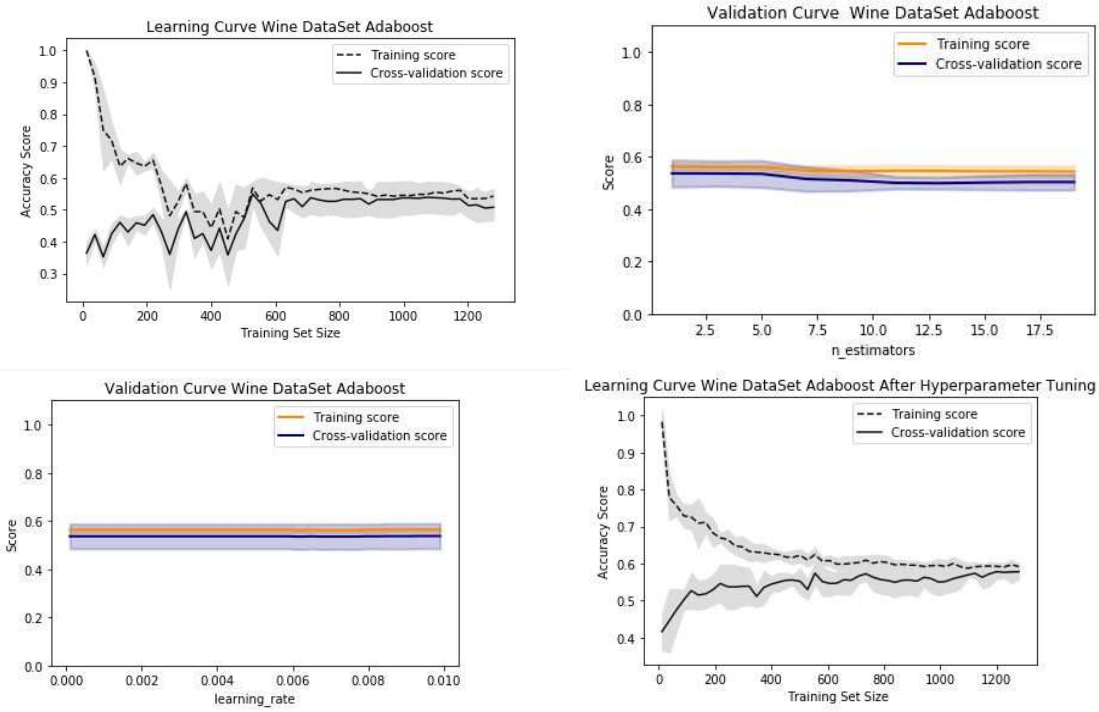

Fig. 5. Boosting-Wine Data set1

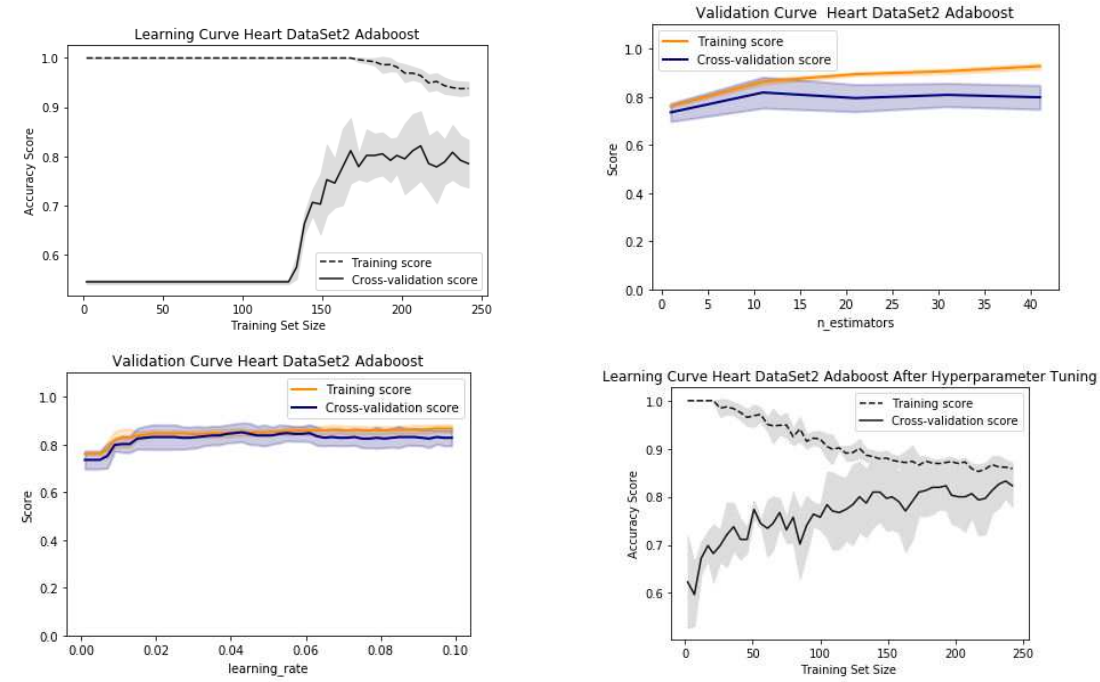

Fig. 6. Boosting-Heart Data Set2

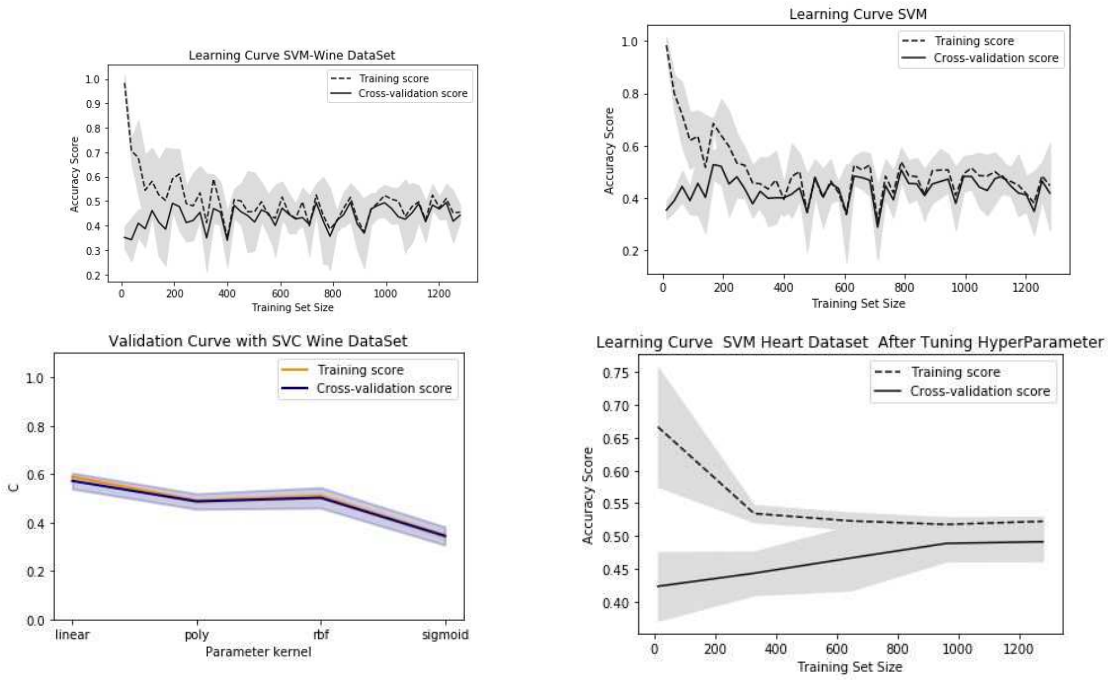

Fig. 7. SVM-Wine Data set1

was still under fitting even with tuned ”C” and ”Loss” and kernel parameter validation curve.

Now We can add more specific features or change the processing type of the feature to reduce bias .So this way our model will learn more. But again that not part of this assignment .

So To overcome this high bias problem, we moved to a more complicated algorithm and used the kernel ’poly’ to overcome the problem of under fitting .Poly kernel has parameter degree. This parameter controls complexity of model . In simpler terms more the degree more complex the model and less the bias. So after hyper parameter tuning we have raised the degree from 1 to 10. As we can see on our final figure after hyper parameter training and validation curve have finally low biased and we have approached to ideal model . Last but not least our paradigm has less prejudice and continues to understand well. By increasing the degree of poly kernel we have solved the problem of fitting high bias.Finally hyper parameter used for this are:- C=6,kernel=’poly’,degree=10

SVM Heart Data set 2 similar to wine data sets , SVM learning curve for heart data is begun under fitting . As the difference between training and testing curve is small. Contrary to wine data set , we have used more complex SVM , Liner SVC to reduce the high bias issue

As we can noticed in validation curve ,parameter ”C” and ”loss fucntion ” has much bigger impact on linear SVC to reduced biased. With loss ”squared ” and C parameter value of 7 we have finally solved the under fitting issue heart Data Set. As the model’s measurement period is maximum relative to other ones, it took more time to change hyper parameters of different values. But in this case we finally managed to get good training and validation curve for ideal model.

11. KNN

Nearest Neighbour algorithm is used for learning algorithms that are both supervised and unsupervised. Its basic idea behind the training set is to find the closed training points and predict the final output target.

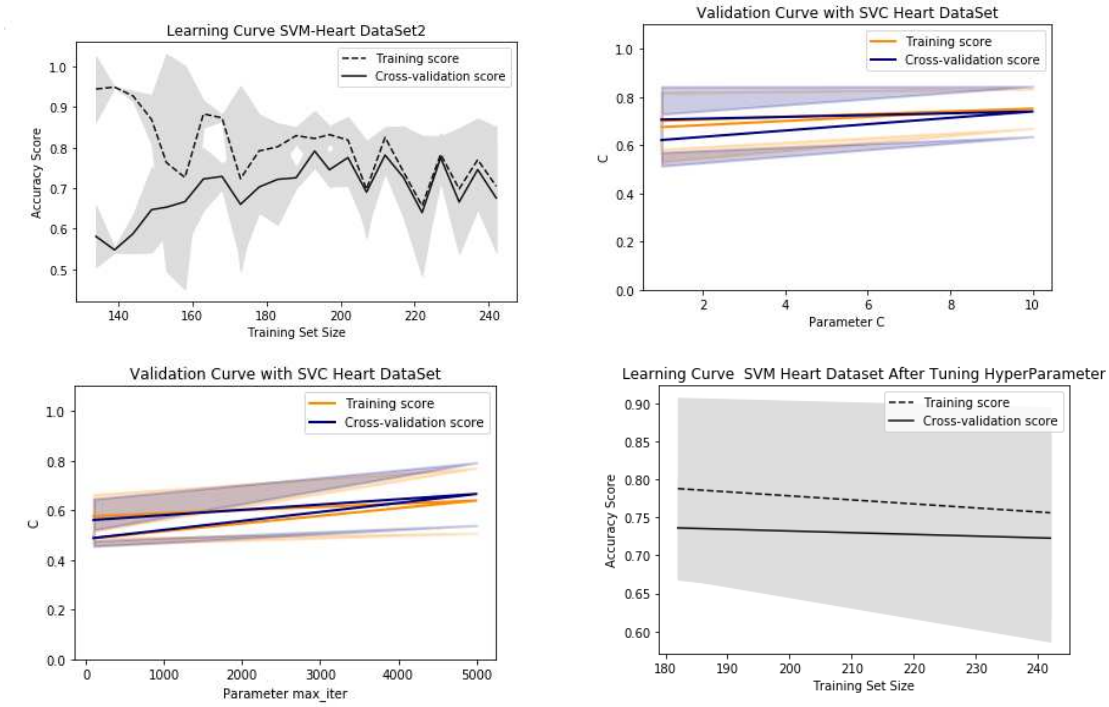

Fig. 8. Svn-Heart Dataset2

Analysis -KNN-Wine Dataset1

As we can see in figure 9, there is a high variance in base classifier for KNN. Cause wide distance between training and testing Curve. Therefore KNN's base model is over fitting. Now as part of the sophistication of the model, we are going to tune the hyper parameters of KNN. The most important parameter for KNN is "N " when it comes to how many neighbors to train Collection.As we can see from Validity Curve, when both validity and training curve starts to improve, we have a pleasant no. somewhere about 10.Please note gap between training curve and validation curve start to almose merged after 10 neighbours. Also we have only 11 attributes in this case. So it does not make sense to used more neighbours than 11 . We need to look at that we have a minimal variance. The second parameter we've tweaked here is whether to use the weight "uniform" and "width."From validation curve it almost make sense to used "uniform" weight when gap between training and validation curve is not bigger than other weights.

Here after tuning hyper parameter model was still over fitting. Like we know, in Knn if there are no features, we need more details like features grow. As explained in the lecture, this is where the curse of dimensional problems falls in picture. So to prevent over-fitting in Knn we can get more data as it helps to reduce variances.Before applying the Knn algorithm, we can apply the Principal Component Analysis, but I believe this is for the next part of the assignment.

So finally we have used data normalization to improved the over fitting in this case. As we can see, we have improved High Variances compared to Original Models in the final figure after hyper parameter tuning and data normalization. We have now used the Standard Scalar to standardize the data and feed the hyper parameter to the final model.It's got good results but it's still over fit. We used below hyper parameter for this:- weights='uniform',n_=11

11.1. Analysis KNN-Heart Data set2

Similar to Wine Data set, Heart Data set Base model also over fitting.We followed same approach as followed on wine data set. Tuned parameters and normalized the data . While we have better

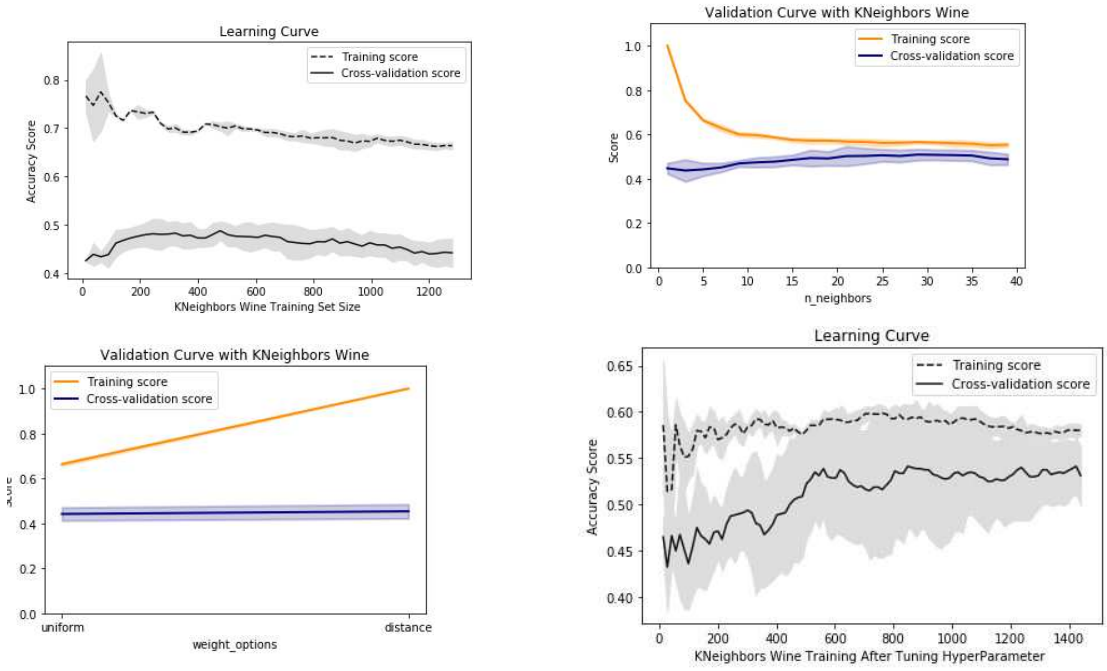

Fig. 9. kn-Wine Dataset1

variance than base model after doing model complexity and normalization on data .But still model is over fitting .

There we also introduce data standardization because of ”Dimensional Curse” and we have less details there.Now we can add more data , add more featured , standardized feature scaling to reduce variance . But that’s not part of this assignment and we might see them in later assignments .Finally Based on our study, it is safe to say that if we have less data and more functionality, we should not use KNN but it is good fit when we have higher measurements and lots of data.We used below hyper parameter for this:- weights=’uniform’,n_=2

12. Lesson Learnt

I did not initially used ShuffleSplit for my Kfold and my final curve start over fitting because of this. Based upon my validation curve and writing from this assignment i noticed my final curves were off with hyper parameters used. Finally on last moment i re-write all my final curve with ShuffleSplit. So lesson learnt minor mistakes on machine learning can caused over fitting and under fitting . But if your’s basic are strong you can easily catch those mistakes.

13. Conclusion

In terms of accuracy and best model with no over fitting and under fitting Ada boost performed well with validation and training accuracy of around 60%Ḟor Wine data set also best model with no over fitting and under fitting Ada boost performed well with validation and training accuracy of around 80% . Knn model performed worst in terms of accuracy and have more variance even after hyper parameter tuning. NN and SVN take more computing time and performed worst or equal to Boosting in accuracy . So we can finally conclude that it is very important to learn curve, validation curve, hyper parameter tuning and model selection. We won’t know whether our model is suffering from excessive prejudice or high variance issues without learning curve. Once we know what issue we have to solve there are lot of parameters for a model to tune which can

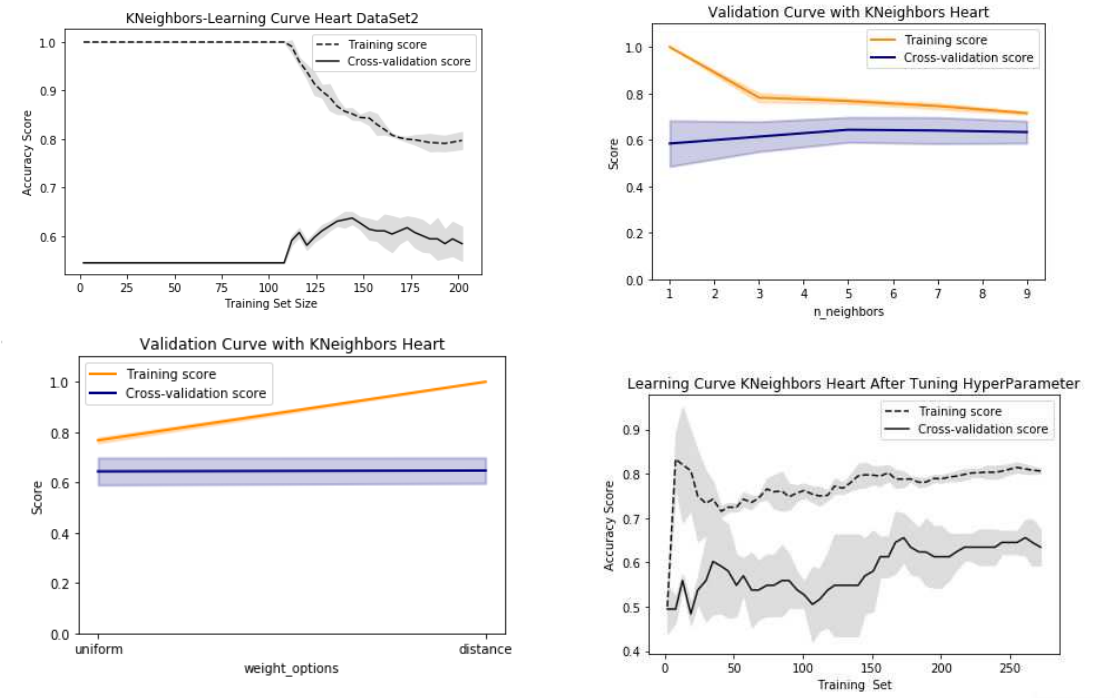

Fig. 10. kn-Heart Data set2

make the model good or bad.Model selection is also very important as we see SVM model also performed similarly to other models but it has more time to calculate. Same time KNN model is suffering from the problem of ”Dimensional Curse.”

From my understanding of machine learning up to now, we can continue with a simpler model such as a decision tree and then we can use more complex model based on more research and the problem we face. As we see in our scenario, in less calculation time and similar or better precision than other types, decision tree and boosting worked well.Although neural network has performed well, we need GPU for that and it doesn't make sense to use GPU for smaller data sets like this.Complexity of learning curve and model are techniques that can be used on those algorithms to solve bias and variance problems.

For them this was not possible thanks to the exclusive Piazza, after hours and Slack for Machine Learning course 7641 Spring 2020.Author also wants to thank Sklearn library and documentation.

References

https://www.dataquest.io/blog/learning-curves-machine-learning/
https://scikit-learn.org/stable/modules/learning_.html
https://www.kaggle.com/abhikaggle8/wine-classification
https://www.kaggle.com/cdabakoglu/heart-disease-classifications-machine-learning
https://keras.rstudio.com/articles/tutorial$_{o}verfit_{u}nderfit.html$
https://medium.com/ml-research-lab/under-fitting-over-fitting-and-its-solution-dc6191e34250
https://towardsdatascience.com/overfitting-vs-underfitting-a-complete-example-d05dd7e19765