Supervised Learning (Assignment 1)

Jacob Seo
CS7641
jseo302@gatech.edu

1. Dataset Introduction

There are two datasets datasets for this classification assignment:

1. UCI (University of California Irvine) Heart Disease Dataset
2. Water Quality Potability Dataset

For the UCI Heart Disease Dataset, it has a mixture of categorical, integer, and real number of values that would fit greatly for the classification. It has 303 instances along with 14 attributes. It has some missing values that would require some either imputing or deleting. The dependent variable that would be used for the machine learning models is 0 and 1. Some attributes consist of continuous values that would also require some encoding as well as scaling.

Water Quality Potability Dataset consists of 3276 instances with 10 attributes. Unlike the previous dataset, it is majorly interpreted in continuous values. The dependent variable that would be used for the machine learning algorithms is 0 and 1. However, the data is highly imbalanced such that Potable water is much less than the water that is not.

2. Dataset Preprocessing

As mentioned above, both datasets have some missing values and UCI HD has some categorical values whereas the Water potability dataset has the fully continuous values. To solve the missing data, I dropped the instances that included NaN or blank values.

For the categorical data, I changed those categories into binary formats by having additional columns.

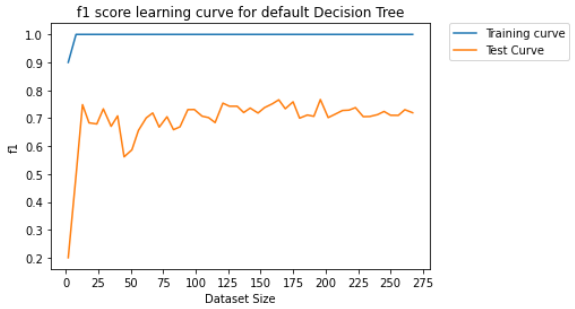
For the continuous values, I used the Standardscaler to normalize the scales. In addition, I used the resampling method to keep the balance between dependent variables for the Water Quality Dataset.

Please note that the measure for UCI Heart Disease is in f1-score due to the imbalance of the independent variable.

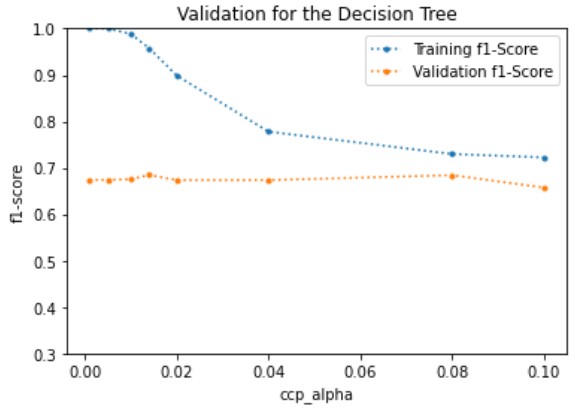
3. Decision Trees

3.1. UCI Heart Disease Dataset

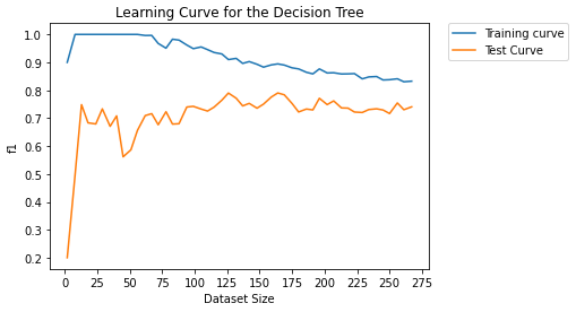
The learning curve for the default decision tree shows the gini index, ccp_alpha values of 0.0. The gap between the training curve and the test curve is considerable, meaning that low bias and high variance are impacting its result and have overfitting issues.



In order to optimize this problem, I used GridSearch to get its optimal hyperparameter. With the 10-fold cross validation, the estimator suggests to use the ccp_alpha value of 0.014 and use the same gini index. With the x range extending from 0.00 to 0.1, the f1-score for the training set gradually decreases and gets closer to the f1-score of validation. With the optimal ccp_alpha values (0.014), the optimal f1-score becomes 0.68.

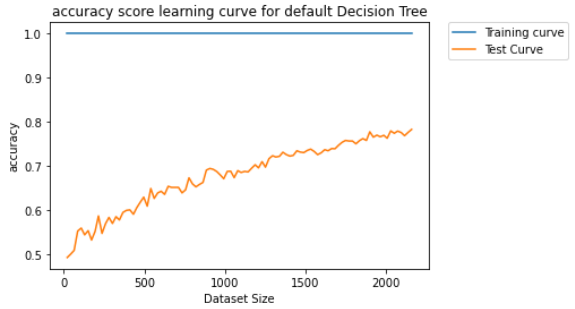


With the optimal hyperparameter another learning curve (below) describes how the decision tree classifier interacts with different sizes of data instances. We can notice that the gap between the testing curve and the training curve is decreasing. If the total dataset instance has been greater, there would be a convergence point. I can extrapolate that the larger dataset for heart disease would help the model to get better f1-scores.

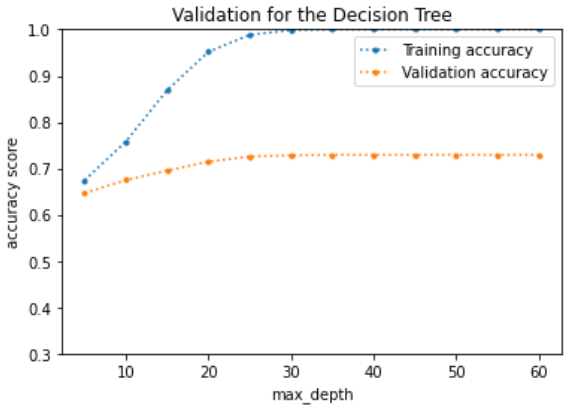


3.2. Kaggle Water Potability Dataset

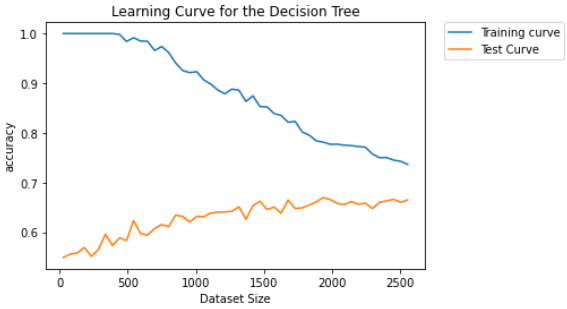
The learning curve for DecisionTreeClassifier (from sklearn library) with its default values shows that there is a high variance and low bias for the testing set, causing the model to overfit the data and not be able to generalize. The Test Curve is increasing to the maximum instance numbers. The accuracy score for this model is 78.29%



With the GridSearch estimator, I got the hyperparameter of max depth=35, and minimum samples leaf=1 and criterion of gini index. The validation graph below also shows that the increase in max depth gets the training accuracy to have a high variance and low bias, causing the overfitting issue. Accuracy for this model increases as I tuned the maximum depth but not as efficiently.



With the optimal hyperparameter including the ccp_alpha of 0.002 (manually tuned), the learning curve below shows that the training curve of this pruned tree has less variance and higher bias. If the dataset size was greater, both Training curve and test curve would have met a crossing. However, it caused decrease in the accuracy (63.75%)

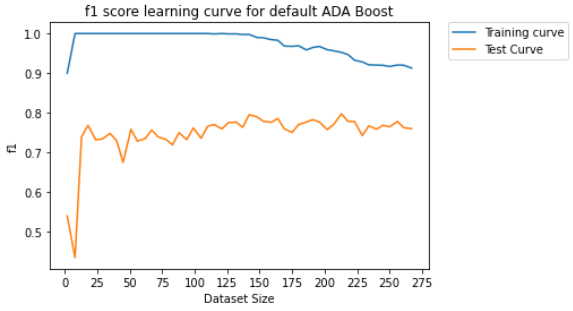


4. Boosting

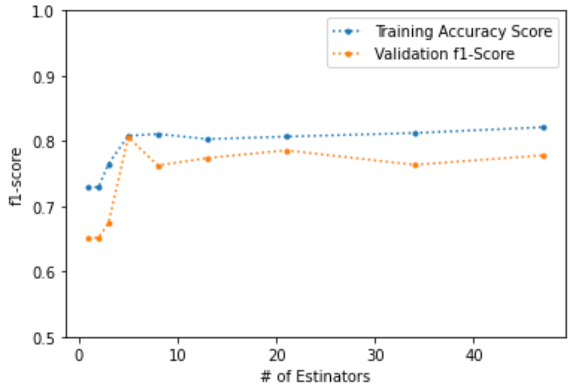
Please note that AdaBoost from sklearn is used for Boosting algorithm for both of the datasets

4.1. UCI Heart Disease Dataset

The learning curve with the default AdaBoost with the 10-fold cross validation shows that the Training curve has the high variance with low bias which would cause the overfitting issues. Its f1-score is 0.79 and its gap between the Test curve and Training curve is also quite considerable. However, the overall trend of its training and testing curve looks very promising such that with more data size the bias and balance seems to balance out for the training curve.

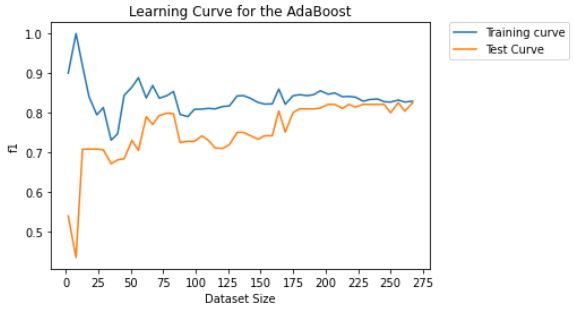


To get better scores I ran the gridsearch estimators to get its optimal hyperparameter. The estimator suggests using the learning rate of 0.15 with the number of estimators of 5 for the f1-score's optimal value. As you can see from the graph below, the f1-score gets less accurate if we add more estimators than 5.



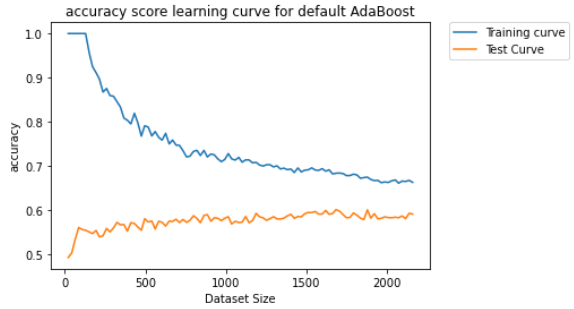
With the optimal hyperparameter for the AdaBoost, you can see that the model fits the shape of the data well from the graph below. As the number of the data increases, both training and testing curve balances out its bias and variance (increase in bias and decrease in variance for training curve vs increase in variance and decrease in bias for testing curve). Eventually they both

converge their scores at the dataset size close to 275.

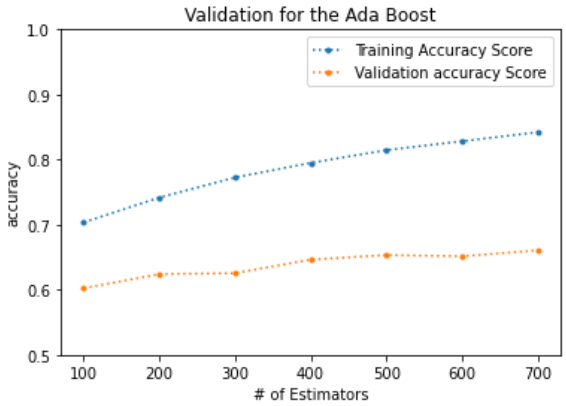


4.2. Kaggle Water Potability Dataset

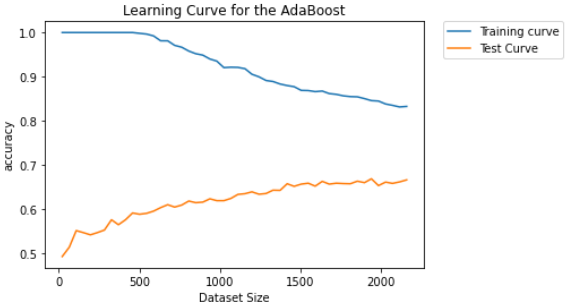
The learning curve with the default AdaBoost with the 10-fold cross validation shows that there is a rapid drop in the accuracy with the increase in the dataset size. This means that the training set has the lower variance and higher bias. The accuracy score for this model is 60.20%



With the GridSearch estimator, I got the hyperparameter of learning rate=0.95, and n_estimators=700. The validation graph below also shows that the optimal hidden layer gets the accuracy score of 65.04%. However, the validation curve shows that it would overfit the data as the number of estimators increases.



With the hyperparameter, the learning curve shows that the training curve would show low variance in comparison to the bias; however, the model with the default values has a closer gap of testing and training data so overall, the parameter increased its variance and decreased the bias for the testing set. The accuracy, however, is increased to 69.58%

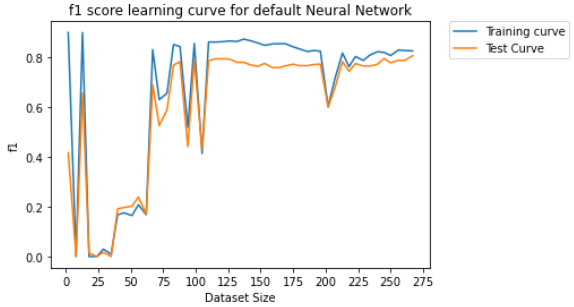


5. Neural Networks

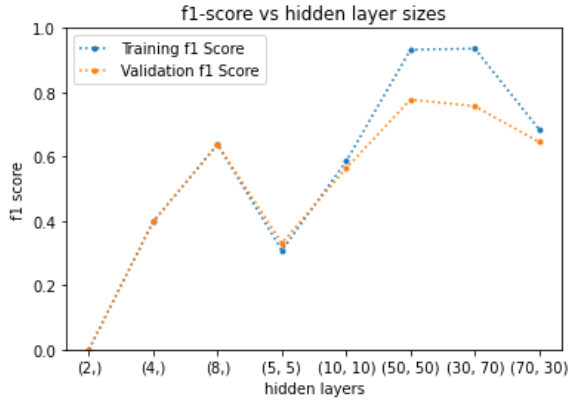
5.1. UCI Heart Disease Dataset

Since the number of instances and features of the heart disease dataset is quite small, there is less opportunity for the neural network model to learn. Not using batch and single instances to adjust models makes the model more sensitive to outliers. The f1 learning score (figure below) for the default neural network model shows that trends in Testing and Training curve are very similar. Meaning that there has been a good balance

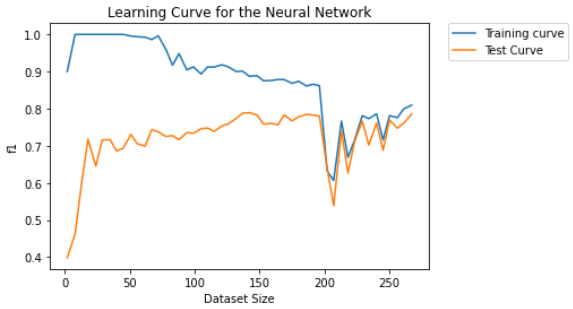
bias and variance across the testing and training dataset. Having a bigger data size for this model would have helped to get a higher score.



With the GridSearch estimator, I got the hyperparameter of alpha being 0.2, and hidden layer sizes of (50,50). The validation graph below also shows that the optimal hidden layer gets the f1 score of 0.77.

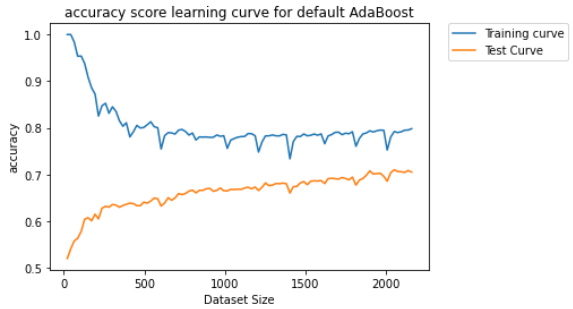


With the optimal hyperparameters, the learning curve for this model shows the relationship of data size and its Training curve's performance such that the increase in data size up to 200 would result in decrease in variance and increase in bias for the Training curve (and vice versa for the test curve).

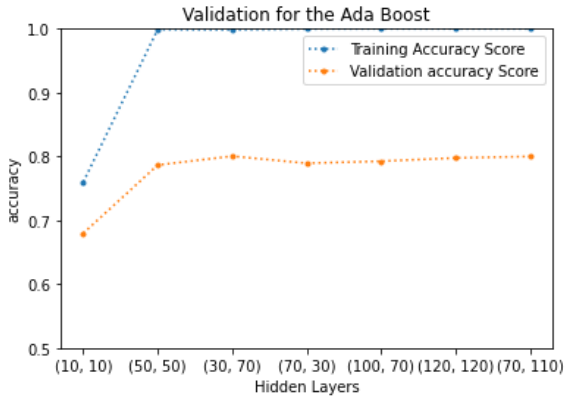


5.2. Kaggle Water Potability Dataset

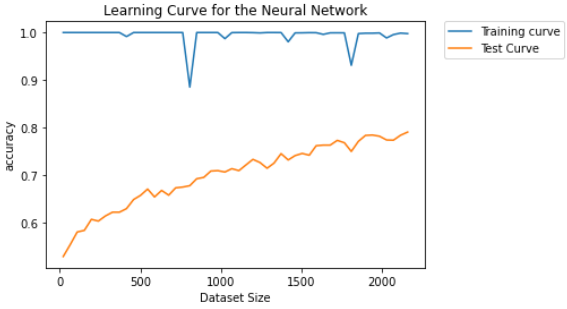
The learning curve for the MLPClassifier (from sklearn library) with the default hidden layer size (100,) and alpha of 0.0001 shows the Training and Test curve with opposite bias-variance trade offs. Such an increase in dataset causes increase in variance and decrease in bias for Training set (vice versa for Testing set). The accuracy score for the neural network with the default parameters is 0.71.



The GridSearch estimator suggests the values of alpha=0.5, and hidden layer size=(30,70). With those parameters, hidden layers with (30, 70) gets the accuracy score of 80%



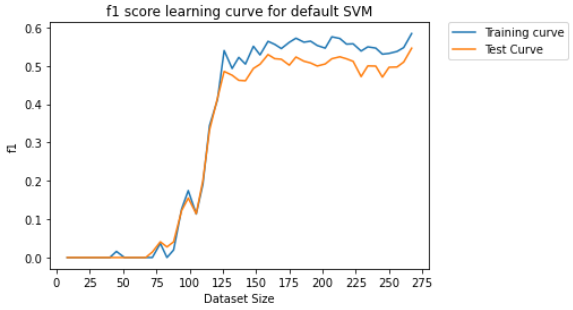
The use of hyperparameters for this model increased the accuracy, it has compromised the bias-variance tradeoff such that Training Curve shows that there has been a high variance and low bias (values are close to the 1s). Since the test curve keeps on increasing, this model would have had better fit to the data if the data set size were greater.



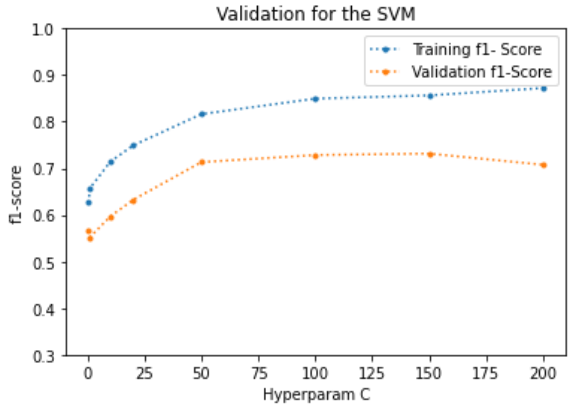
6. SVM

6.1. UCI Heart Disease Dataset

The graph below is the learning curve for SVC (sklearn) with the 'rbf' kernel and C=1. Both Training and Test curves seem to have matching trends such that after 110 data set size the f1-score increases to 0.45. The model does not seem to overfit or underfit the data so it has the balanced variance and bias ratio. However, the overall score for this model is still very low. With the curve tilting upwards at the end, I can speculate that with more dataset, it can have better scores.

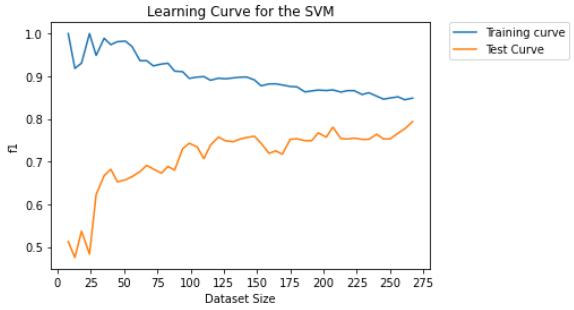


The GridSearch estimator suggests to use C=150 and gamma=0.0001 with the kernel used for rbf. The relationship between C parameters suggests that the f1-score for validation score actually decreases when the parameter C goes beyond 100 and training score starts to increase once it hits 150. With more C values, the bias decreases and the variance increases for the training curve.

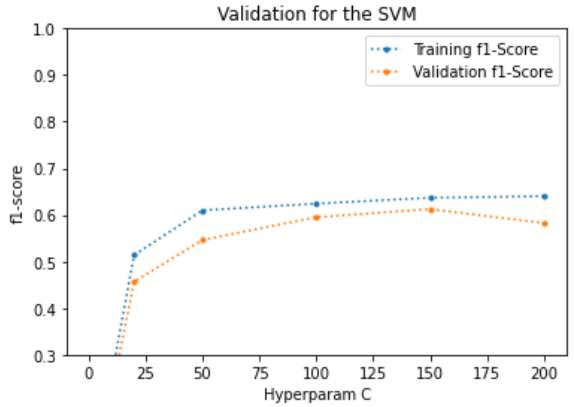


With the optimal values for the C and gamma, the learning curve (below) shows that there is good trade off for bias and variance for the Training set such that more data would decrease the variance and increase bias. For the Test curve, more instances would increase its variance and decrease its bias. The Rbf kernel would

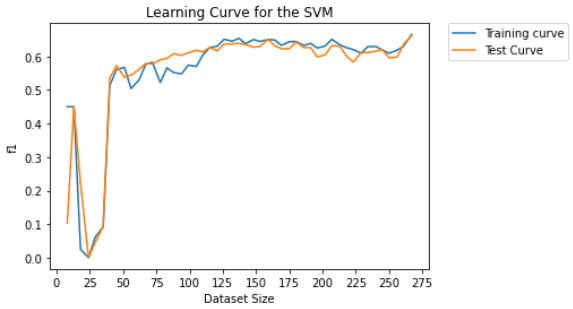
benefit more if the heart disease dataset has a larger dataset.



Another kernel that is used for the SVM is sigmoid. Grid Search estimator suggests to use the parameter of C=150 and gamma=0.000001. However, the overall f1-score is still quite low. From the graph below, the optimal C would still give out the f1-score less than 60%.



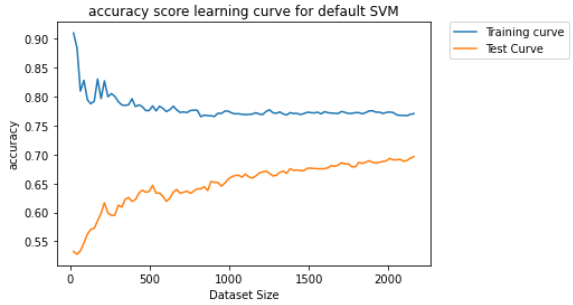
The learning curve for the sigmoid kernel seems similar to that of rbf (with the default parameters). Such that it does not have overfit issues but the overall score of the model itself is low.



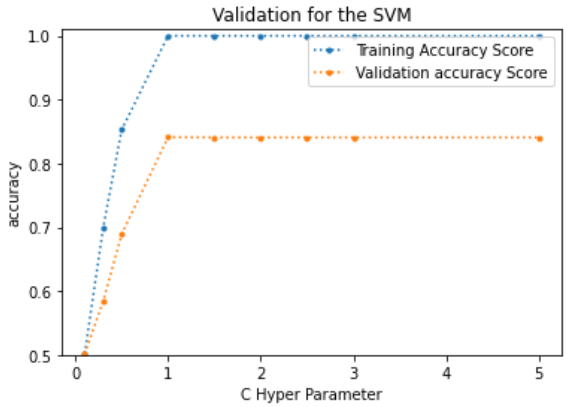
Out of two kernels that are listed, the rbf kernel gives out a better fit for this data set.

6.2. Kaggle Water Potability Dataset

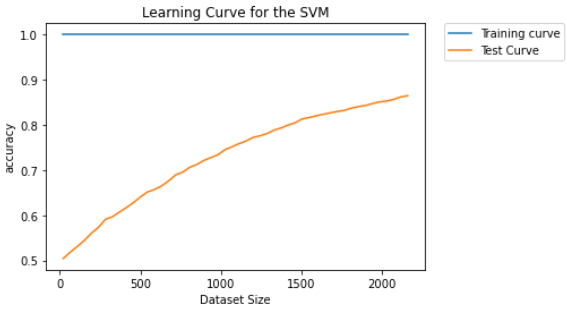
The learning curve for SVC with the default parameters (with rbf kernel) shows that the Training Curve and Test Curve balances out the bias-variance trade off such that with more data set size, the Training Curve has increase in bias and decrease in variance and vice versa for the Test Curve. Accuracy for this model with default parameter is 69.62%



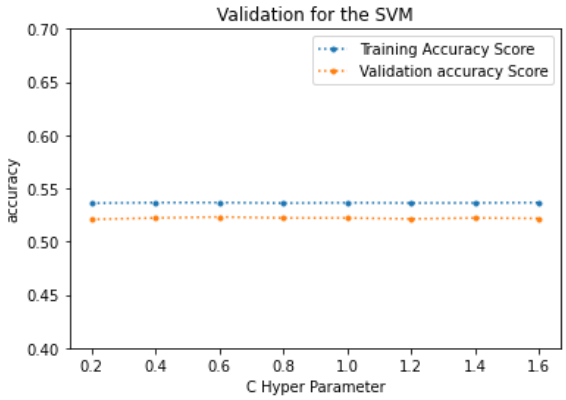
The GridSearch estimator suggests the C=1 and gamma=3 for this model. With the hyperparameter tuning, the validation graph shows that there is an improvement on its accuracy; however, it compromises the bias-variance trade off such that the Training set gets higher variance than bias.



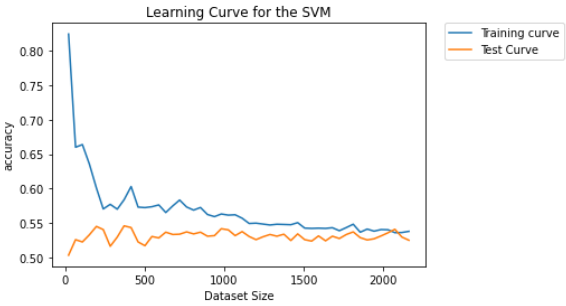
As a result, the hyperparameter also reflects similarly on its learning curve. The accuracy has jumped to 60s to early 80s; however, it causes high variance and low bias which would lead to overfit.



For other kernels, I used linear to see if it fixes the overfitting issue. The GridSearch estimator suggests to use c=0.6, and gamma=0.0001. The validation graph below shows that there has been a decrease in the gap between the Testing Curve and Validation Curve. Due to the minute difference results among the other C parameters, the accuracy stays between 51 and 54 for both curves.



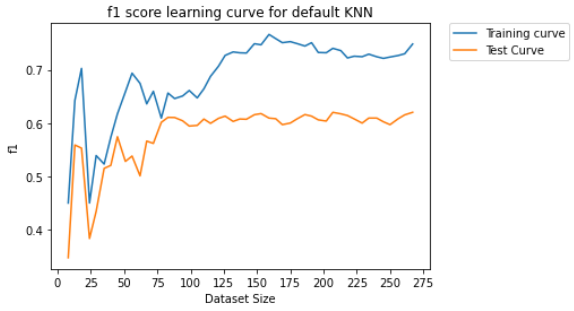
Even though the overfitting issue might have been resolved, the accuracy score for the linear kernel is worse off, ending up with an accuracy score less than 55%.



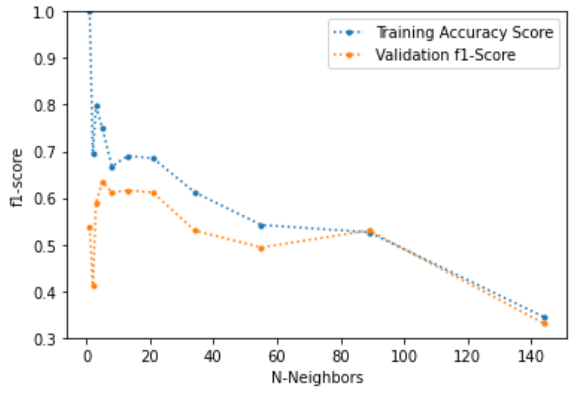
7. KNN

7.1. UCI Heart Disease Dataset

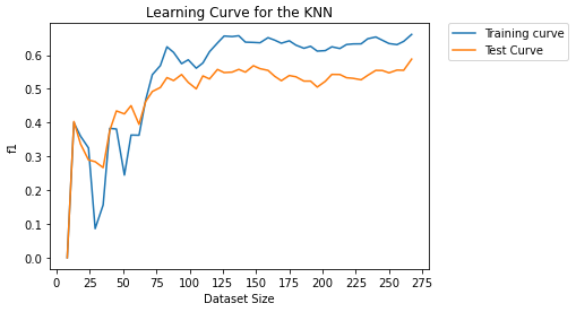
The graph below is the learning curve for KneighborsClassifier (sklearn) with its default parameters. Training and Test Curve trends seem to match their patterns. However, with increase in dataset size, their gap gets wider and Training curve gets closer to 1 at a faster rate. This observation implies that the Training curve's variance is increasing much more than its bias which would cause the overfitting issue. f1-score for this model with the default parameter is 62.03%.



The GridSearch estimator suggests the n_neighbors=5 and uniform weights for this KNN model. The validation graph below shows that an increase in n_neighbors would actually decrease its f1-score. However, increasing the neighbors would definitely help out decreasing the gap between the test and training curves.

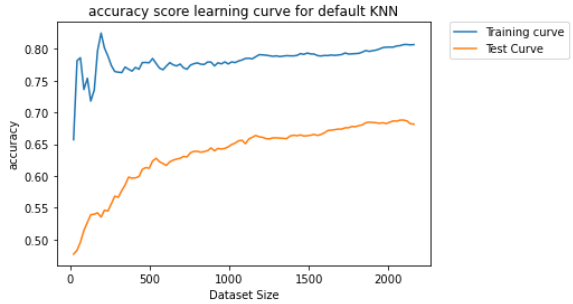


With the optimal value of neighbors, The learning curve shows the decrease in gap between the Testing and Training curve. Between the dataset size of 20 and 60 the f1-score of the testing curve actually surpasses that of Training Score. But as the data set size increases, the Training curve gets higher variance and gets higher values than the Test curve. If the heart disease data was larger, SVM would get a better estimate with a good fit

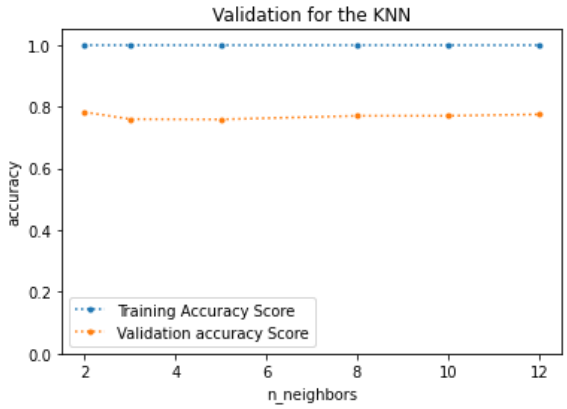


7.2. Kaggle Water Potability Dataset

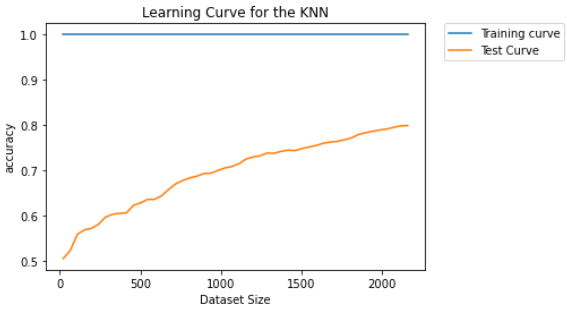
The learning curve for KneighborsClassifier with its default values shows that both training and test curve have similar trends. At the data size of 250, the training curve gets more boost in its accuracy due to its outliers. Overall the training curve has higher variance than bias and an increase in data size would not change its overfitting issue. Accuracy for this model with the default parameter is 68.79%



The GridSearch estimator suggests the n_neighbors=2 with the distance weights. This parameter would increase its accuracy considerably; however, it leads to having its training curve to have more variance and less bias even with the cross validation values of 10.



The learning curve with the hyperparameter also reflects the test and training relationship. Such that While Test curve has increased in accuracy the bias and variance tradeoff is so great that it would cause overfitting.



Conclusion

Here are the results for the two datasets:
Please note that Heart Disease data is the f1-score value whereas the water potability data is in accuracy.

UCI Heart Disease Dataset (f1)
Decision Tree: 79.41%
Neural Network: 77.62%
AdaBoost: 90.67%
SVM: 59.70%
KNN: 58.62%

For the Heart Disease dataset, the Ada boost has the highest f1-score of 90.67% whereas the KNN has the lowest scores of 58.62%.

Kaggle Water Potability Dataset (accuracy)
Decision Tree: 63.75%
Neural Network: 81.67%
AdaBoost: 69.58%
SVM: 48.33%
KNN: 82.50%

For the Kaggle Water Potability Dataset, the KNN has the highest accuracy value. However, due to the high variance and low bias representation of overfitting, I would say the Neural Network would be overall better. However, since the number of instances and features of the water potability dataset is quite small. It would have worked much better if the dataset size is bigger. Not using batch and single instances to adjust models makes the model more sensitive to outliers. Lower features with a large number of nodes makes the low dimensionality.

For Clock Time, neural networks and SVM takes the most time running the process both for training and prediction. The difference becomes obvious when running on the larger dataset. Decision Trees have the least amount of time to run on both test sets.

References

[1] Bouckaert, Remco R. "Heart Disease Datasets." ***UCI Machine Learning REPOSITORY: Heart Disease Data Set*****, https://archive.ics.uci.edu/ml/datasets/heart+Disease.**

[2] Kadiwal, Aditya. “Water Quality.” *Kaggle*, 25 Apr. 2021, https://www.kaggle.com/adityakadiwal/water-potability.