CS7641 - Assignment 1: Supervised Learning
Jaeyong Kim (jkim3018)

Dataset Introduction

In this assignment, I chose to analyze 2 datasets framed as classification problems to experiment with the different learning algorithms provided. The datasets are water potability provided by Kaggle and white wine quality provided by the University of California Irvine ML Repository.[1,2]

The water potability helps to address the lack of clean water available in some areas of the world. Features used in this dataset include: pH, Hardness, Total Dissolved Solids, Chloramines, Sulfate, Conductivity, Organic Carbon, Trihalomethanes and Turbidity. Often, deeming water as potable can take a lot of time and effort on the collection and testing end of the water. Natural water has many pollutants that must be filtered out before it can be utilized by people so if water can be deemed potable with the help of a few features, it could greatly speed up the process of getting clean water to the people who need it most.

The White Wine dataset includes different features of wine such as acidity, pH, concentration of different chemicals and alcohol amount. What is interesting about judging the quality of wine is that people generally use the 100-point scale to communicate the wine quality. A score is generally given after an enthusiast tastes and smells the wine. They usually taste for acidity, sugar and alcohol and this dataset has all of these features measured in numerical values. It would be interesting to see if the chemical and sensory features that are measurable could be used to classify the quality. The dataset is classified into 11 classes between 0 and 10, but I changed the target so any score under 7 is poor and any score above 7 is good. This helped to balance the dataset a little and on the 100 point system, scores under 70 are considered to be terrible wines.[3]

Experiments

1. Decision Tree

Decision Trees are a very versatile learner because they can work on both classification and regression problems and don't require any standardization or scaling of the features input into the data. For both datasets, the gini index was used to split the features of data. The gini index is bound from 0 to 1 and is a measure of impurity based on the probability of a feature being incorrectly classified. Validation and Learning Curves are shown for the learner in the next figure. The best params were found using the GridSearchCV method available in Scikit-Learn package.

a. Water Potability

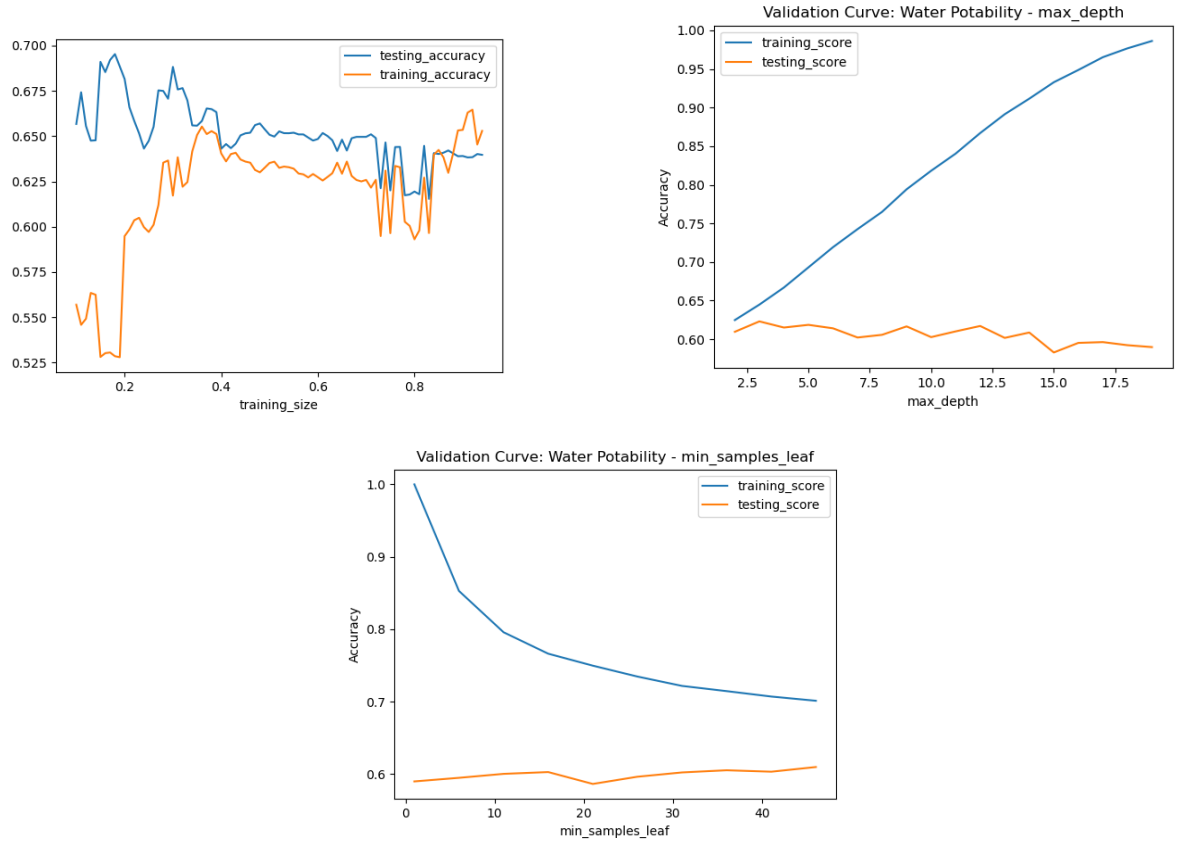
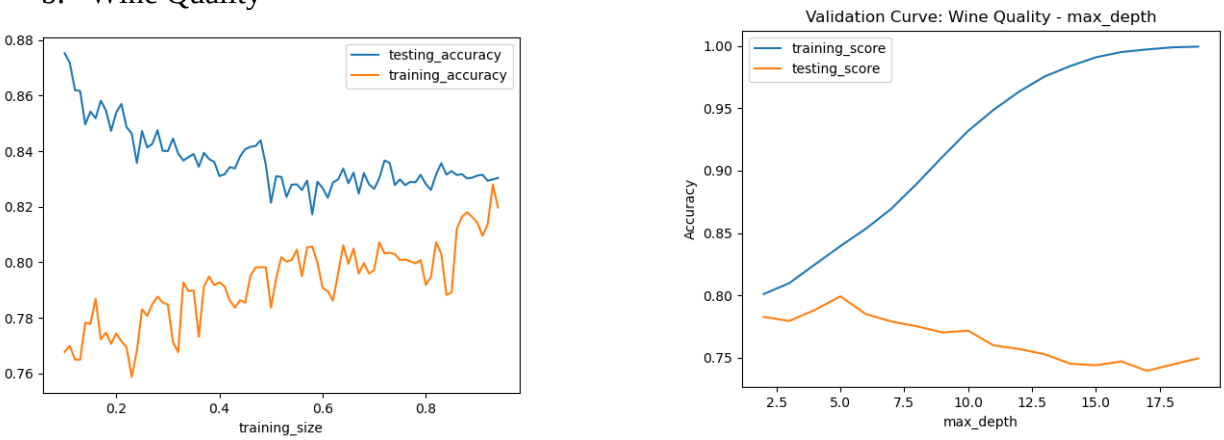


Figure 1: Learning and Validation Curves of Decision Tree on Water Potability Dataset

b. Wine Quality



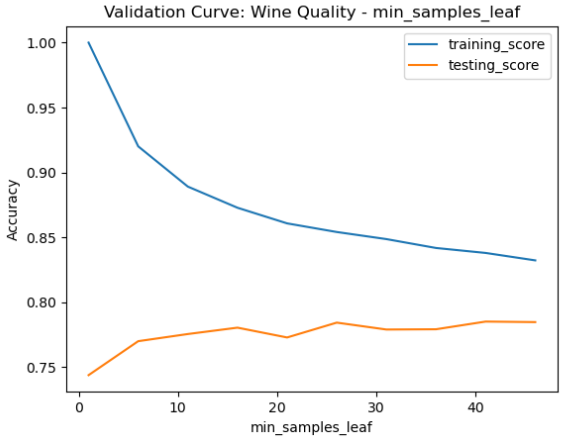


Figure 2: Learning and Validation Curves of Decision Tree on Wine Quality Dataset

2. Neural Network

For this learner, a MLP Classifier was utilized. Multi-Layer Perceptron is a feed forward neural network which uses layers and activation functions to calculate weights and bias coming into each layer. The activation function used is ReLu which stands for the Rectified Linear Unit. Some hyperparameters tested for this experiment were number of hidden layers, learning rate, and max iterations.

a. Water Potability

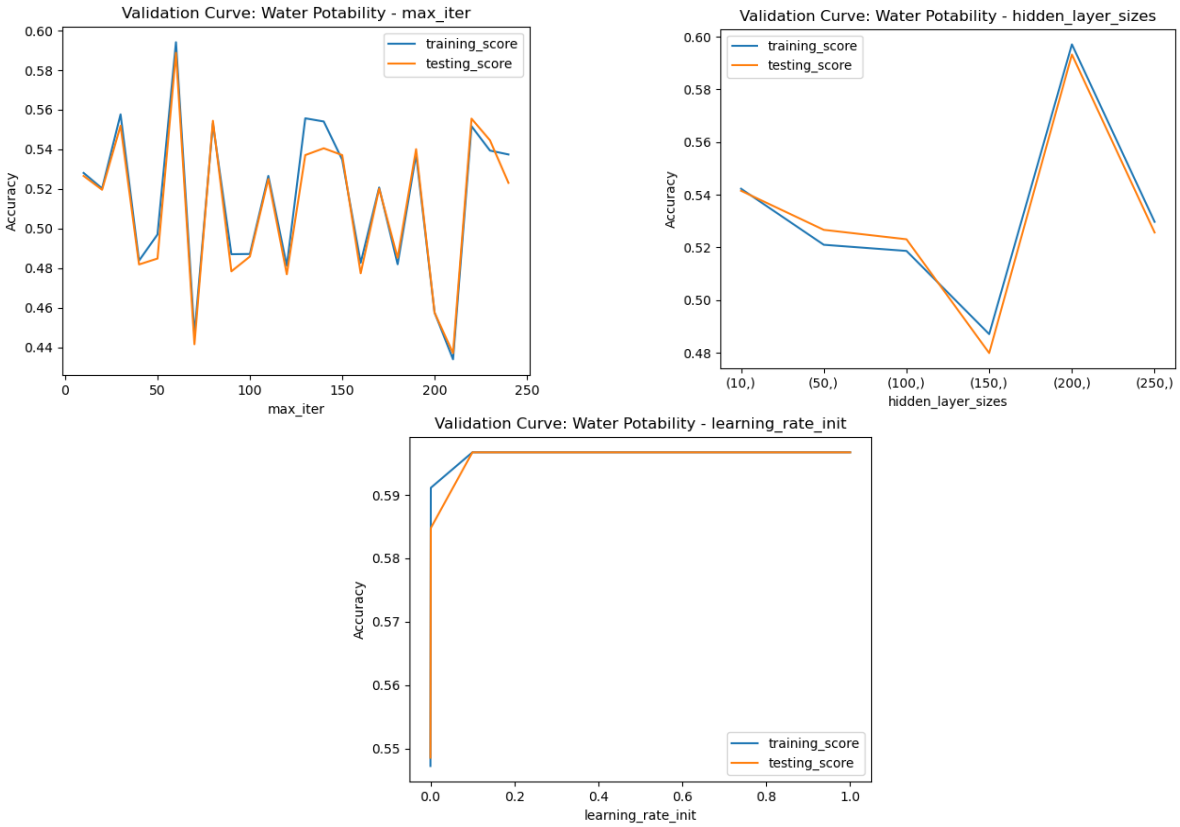


Figure 3: Learning and Validation Curves of MLP on Water Potability Dataset

b. Wine Quality

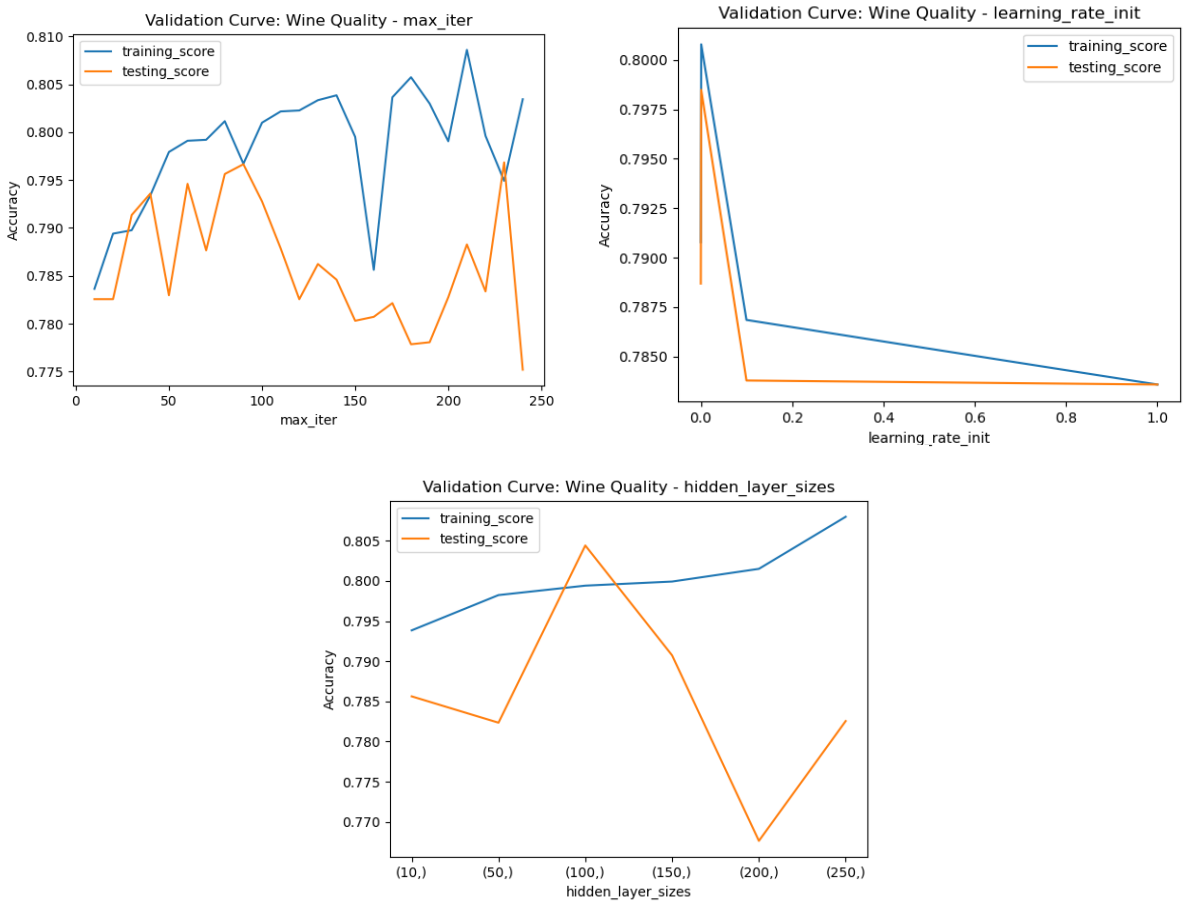


Figure 4: Learning and Validation Curves of MLP on Wine Quality Dataset

3. Boosted Tree

While decision trees are versatile, they are often considered weak learners and techniques such as boosting or bagging can be useful in making them better. In this experiment I used AdaBoost from sklearn to see how that would improve the accuracy of the dataset. Boosting is when multiple weak learners are combined to improve accuracy as a single learner. The parameters that I varied is the learning rate and the number of estimators used.

a. Water Potability

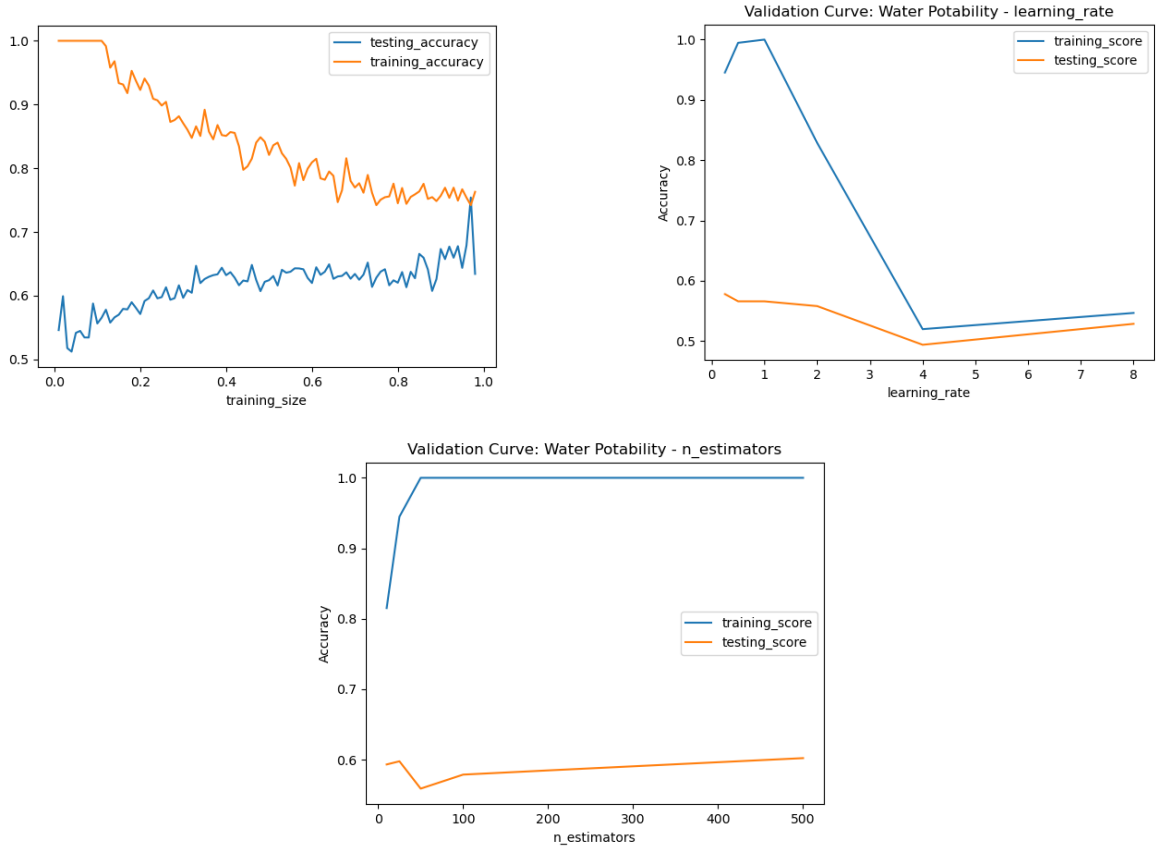
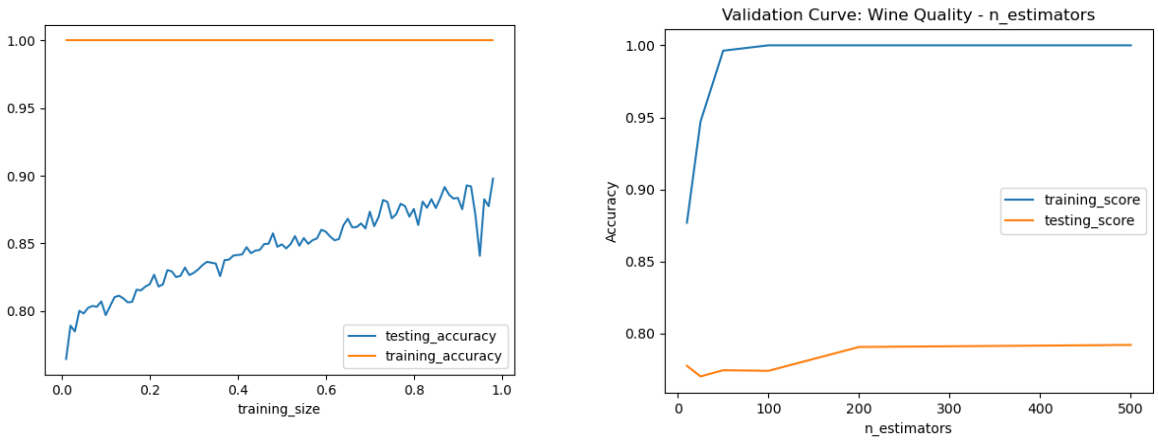


Figure 5: Learning and Validation Curves of AdaBoost DT on Water Potability Dataset

b. Wine Quality



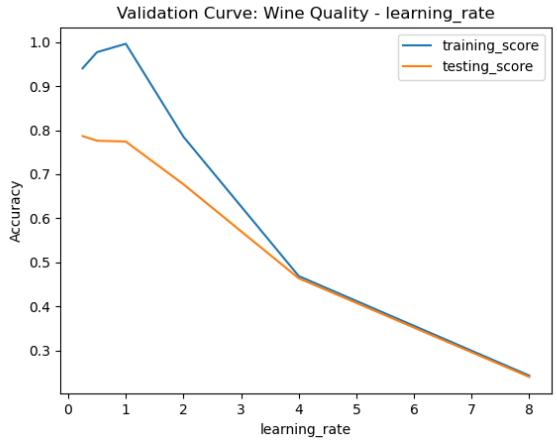


Figure 6: Learning and Validation Curves of AdaBoost DT on Wine Quality Dataset

4. Support Vector Machine

SVM was also implemented with the sklearn package. Some parameters that were varied include C-value which is the regularization parameter and the max iterations for the SVC learner. Different kernels were also used to see what the effects might be. A sigmoid and rbf kernel were used.

a. Water Potability

i. Kernel = 'rbf'

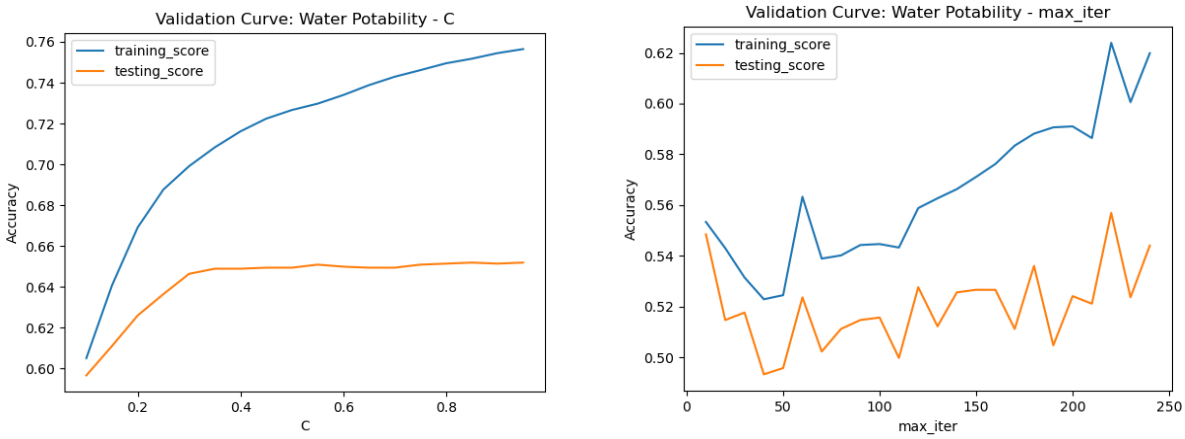


Figure 7: Validation Curves (rbf kernel) of SVM on Water Potability Dataset

ii. Kernel = 'sigmoid'

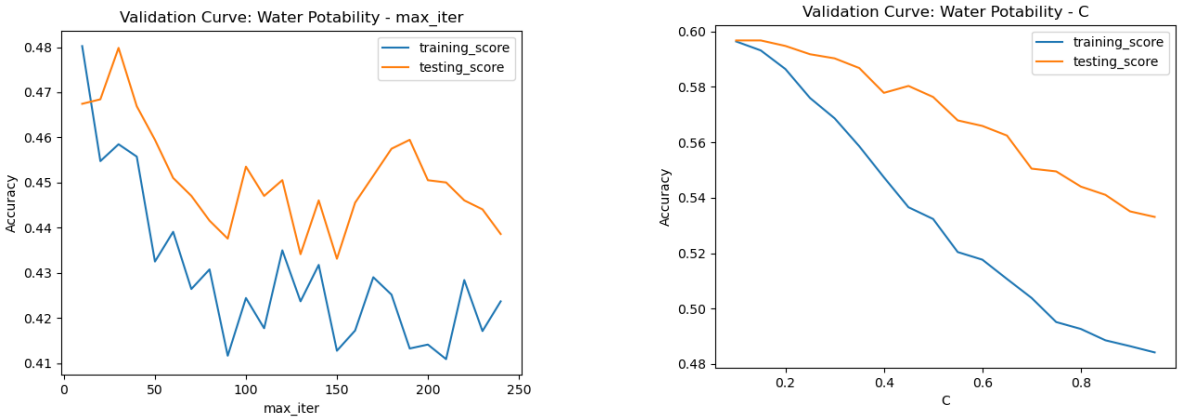


Figure 8: Validation Curves (sigmoid kernel) of SVM on Water Potability Dataset

b. Wine Quality

i. Kernel = 'rbf'

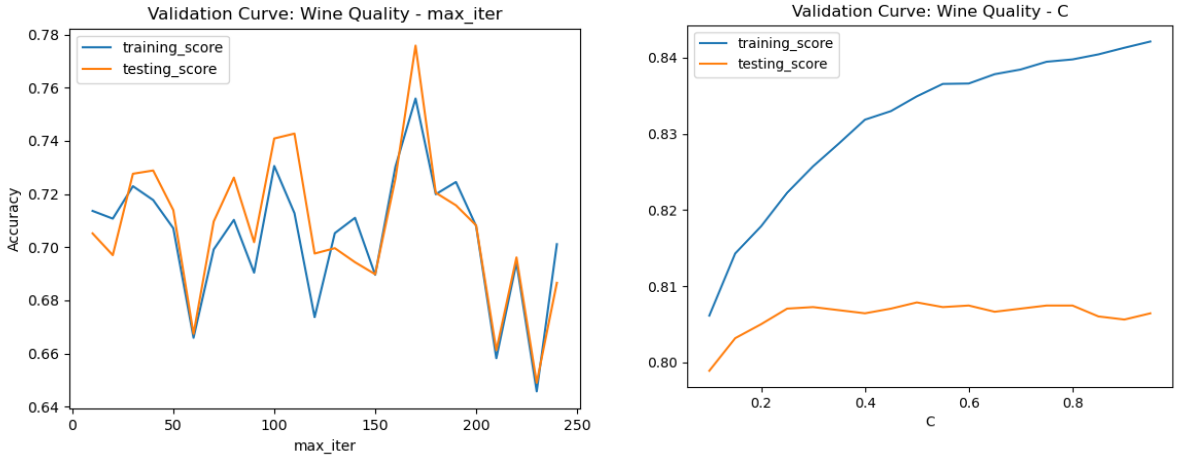


Figure 9: Validation Curves (rbf kernel) of SVM on Wine Quality Dataset

ii. Kernel = 'sigmoid'

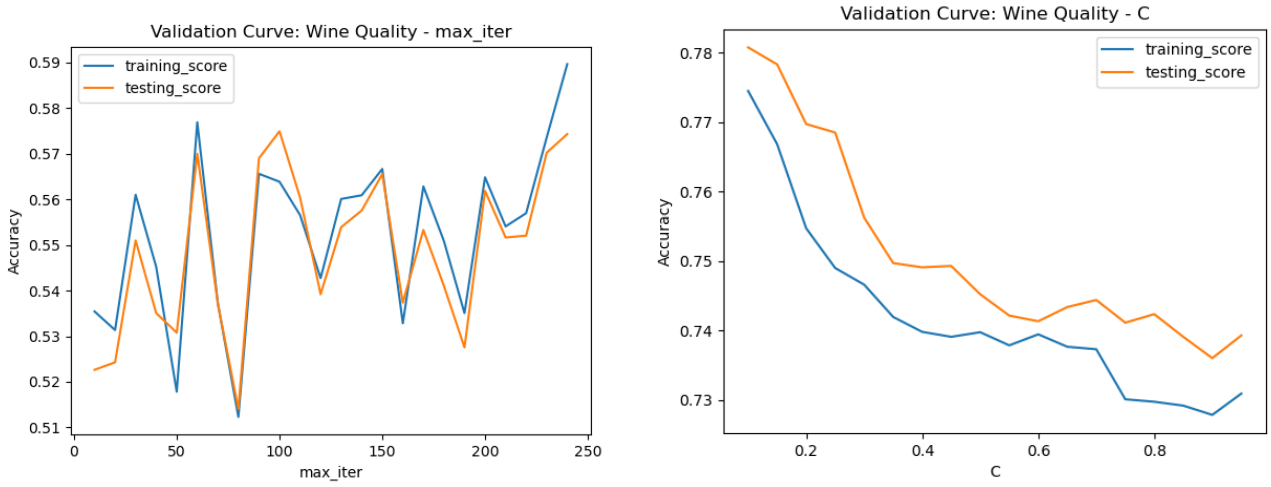


Figure 10: Validation Curves (sigmoid kernel) of SVM on Wine Quality Dataset

5. k-Nearest-Neighbors

kNN was implemented with sklearn and the parameter that was varied was n_neighbors. Because kNN works by looking at distance between points, the data was standardized before being fit by the kNN learner.

1. Water Potability

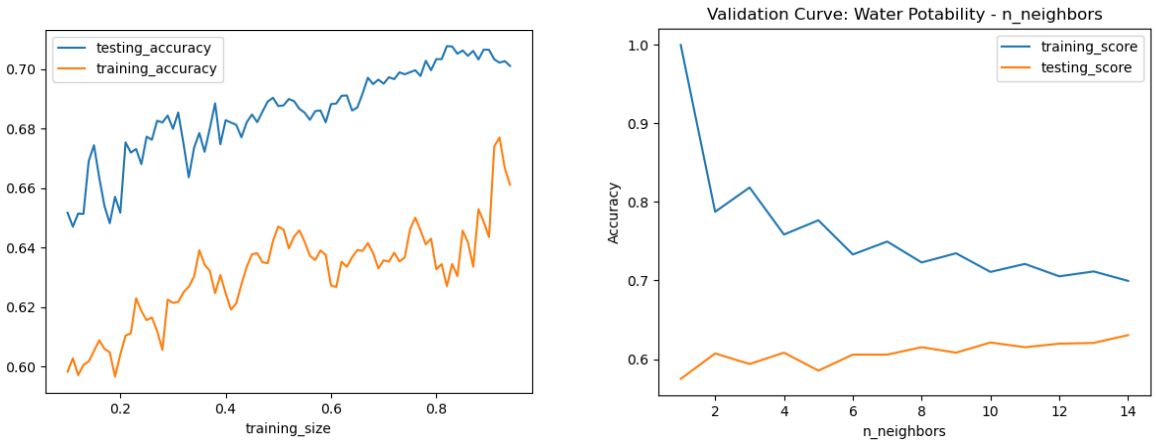


Figure 11: Learning and Validation Curves of kNN on Water Potability Dataset

2. Wine Quality

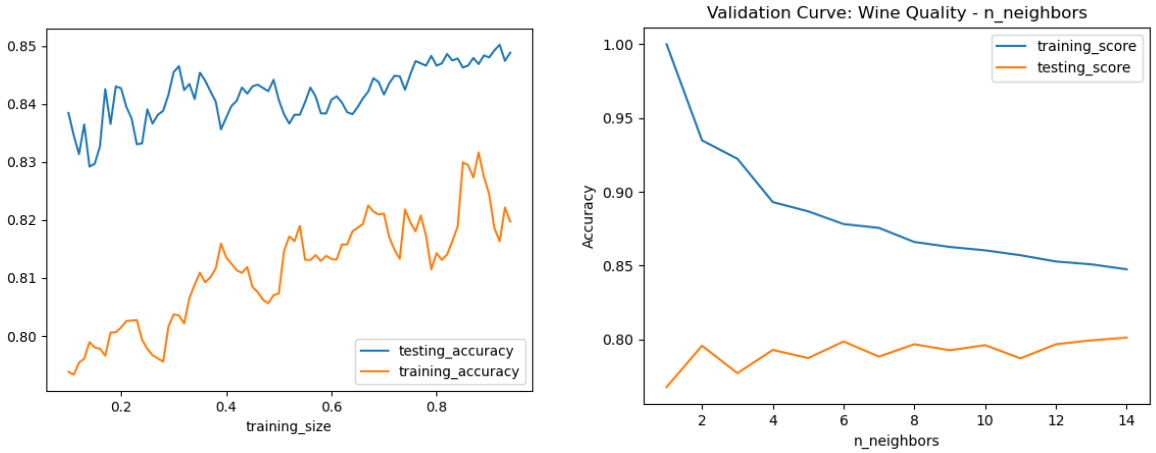


Figure 12: Learning and Validation Curves of kNN on Wine Quality Dataset

Analysis

Utilizing the best parameters found from the GridSearchCV were used for testing the accuracy of the models as well as the wall time in fitting the training data. Grid Search CV uses 5 fold cross validation and accuracy to determine which parameter is best in a certain parameter space. For pruning trees, I looked at max depth and min samples per leaf . I also varied the learning rate and iterations for the iterative algorithms. The figures below show the best accuracy for each of the algorithms as well as the time it took to train the algorithms with varying training sizes.

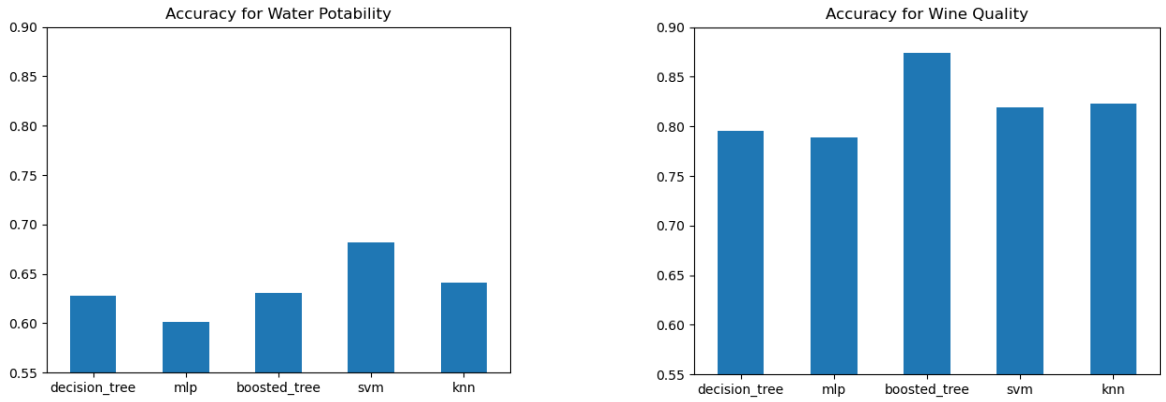


Figure 13: Accuracy of Each Learner on the Different Algorithms

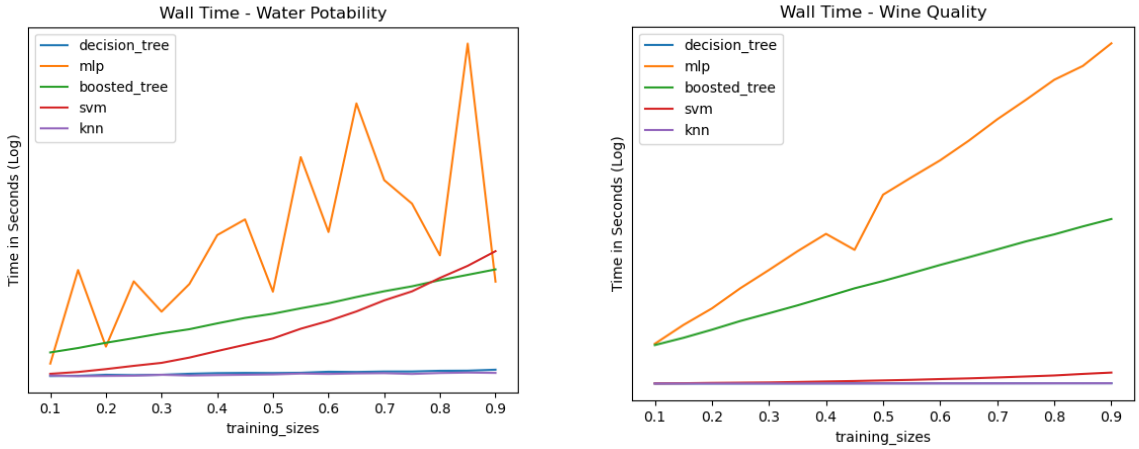


Figure 14: Wall Time vs Training Size of Learners

The algorithms worked a lot better on the wine dataset than the water potability dataset. In both, the SVM and boosted_tree algorithm performed much better than the other algorithms. It's important to note how the boosted tree performed better in both datasets compared to the normal decision tree. In figure 1 and 2 under the decision tree section, we can see evidence of overfitting which is an issue that the decision tree is susceptible to when there is no pruning done. This is shown by the accuracy tapering off and even decreasing when the training size is increased. Pruning a sole decision tree makes the learner miss out on accuracy. With a boosted decision tree, we can be more aggressive with the pruning of the weaker learner. The boosted tree was run with a max_depth of 4 which helped aggressively prune the tree.

MLP and SVM also performed decently well in the wine quality table at least, but we can see from the wall time graphs that these algorithms are very costly in terms of performance. While there is greater accuracy, the computational cost is high for both of these algorithms. Namely, MLP classification is expensive as the number of hidden layers increases and as iterations increase as

well because there is so much to compute. SVM performed better with both datasets than most of the other learners. SVM fits well with binary classification problems and the ‘rbf’ kernel helps with creating an effective decision boundary in identifying the two classes. SVM as well, especially with the rbf kernel, we can see there is also evidence of overfitting with the decrease in accuracy with increasing iterations.

Of the algorithms tested, Boosted Decision Tree is probably the better learner to use. The boosted tree takes care of the overfitting from weak decision tree learner and it is not nearly as computationally expensive as learners like MLP or SVM. It still boasts a high accuracy rate compared to the other learners. If I were to implement a learner in a production environment, I would choose one that has performs well and is computationally least expensive in training and predicting which in this case was the Boosted Decision Tree.

The performance of the learners could also be improved if the dataset was cleaned more. Looking at both target variables for the learners, I noticed that the targets are imbalanced. If I down sampled the data to make balanced classes, it would’ve helped the performance of the learners.

References

1. Kadiwal, A. (2021, April 25). *Water quality*. Kaggle. Retrieved September 26, 2021, from https://www.kaggle.com/adityakadiwal/water-potability.
2. UCI machine Learning Repository: Wine quality data set. (n.d.). Retrieved September 26, 2021, from https://archive.ics.uci.edu/ml/datasets/wine+quality.
3. *What are Wine Scores?* Wine. (n.d.). Retrieved September 26, 2021, from https://www.wine-searcher.com/wine-scores.