CS7641 Assignment 1 – Supervised Learning

Michael Lukacsko
mlukacsko3@gatech.edu

***Abstract*—This assignment explores 5 classifiers (Decision Tree, Neural Network, Support Vector Machine, Boosting, and K-Nearest Neighbor) and how the tuning of hyperparameters affects performance with respect to two different datasets. The differences are analyzed and discussed.**

Keywords—Decision Tree, AdaBoost, Support Vector Classifier, k-nearest Neighbor, Neural Network, Time, Accuracy

I. INTRODUCING THE DATASETS

A. Breast Cancer Dataset

The breast cancer dataset [1] is a widely used collection of patient records collected from the University of Wisconsin. The dataset contains information on 569 patients and is composed of 357 benign cases (62.7% total) and 212 malignant cases (37.3% total). Each patient record in the dataset includes various features such as the patient's age, tumor size, and the presence of certain biomarkers, as well as a binary classification for diagnosis (benign or malignant). In total, each case has 30 features.

B. Wine Origin Dataset

The data in the wine dataset *[2]* is the result of a chemical analysis of wines produced in a specific area of Italy but derived from three different cultivators. Each of the 2000 samples include 13 features such as alcohol content, acidity, malic acid, ash, alcalinity of ash, magnesium, total phenols, flavanoids, nonflavanoid phenols, proanthocyanins, color intensity, hue and OD280/OD315 of diluted wines, and a multi-class target variable indicating the wine cultivator (1,2,3). The target variable totals in this dataset are 633 (31.7%) 1's, 809 (40.1%) 2's, and 558 (27.9%) 3's.

Two differences in these datasets stand out. First, unlike the binary target class value in the breast cancer dataset, the target class of the wine origin dataset is more complex because of its multi class target represented by a 1, 2, or a 3. Second is the fact that the breast cancer dataset has 30 features per record, whereas the wine origin dataset has 13. Similar in both datasets is that they are fairly balanced, thus they provide a more realistic representation of the population and can lead to better model performance. This common denominator will help ensure that the models being tested are not biased towards a specific class.

II. PREPROCESSING THE DATASETS

Preprocessing data for machine learning is an important step because it ensures that the data is in a suitable format for the model to learn from and can help improve the model's performance. Because both datasets are well balanced, there was very little preprocessing that needed to be done. As such, all the target values represented by a B or an M in the breast cancer dataset are converted to integer target values of a 0 or 1 respectively. Lastly, the 'ID' column is dropped as it is not needed.

The wine origin dataset had integer class values to begin with and none of the features needed to be dropped. This dataset was clean to begin with and didn't need to be modified in any way.

Next, scaling of the data is performed on both datasets. This preprocessing step is vital because all features will have similar ranges, which can help prevent certain features from dominating the model. Without this step, noise in the data caused by measurement errors, human error, outlier values, or sampling bias, could potentially affect the accuracy and representativeness of the data. As a result, the sklearn scale package was used to standardize the data by removing the mean and scaling to unit variance.

To complete the preprocessing step, each of the datasets were split into training/test data using sklearn train_test_split. 80% of the data is used for training data and the remaining 20% for test data.

III. CLASSIFIER 1: DECISION TREE

A. Wine Origin Dataset

Starting with a vanilla decision tree and no hyperparameter turning, the decision tree classifier has an accuracy of 85.25%

1) Hyperparameter Tuning

The first parameter selected as part of hyperparameter tuning was maximum depth. Using a max depth range of 1-20 to control the growth of the tree via pre-pruning, it can be seen in the validation curve plotted below in figure 1 (left image) that the performance of the model as a function of the maximum depth of the tree shows the growth of the decision tree to saturation once the maximum depth of the tree exceeds 10. That is, the training score has reached a point at which increasing the depth of the tree no longer improves the performance of the model. This can be visualized in the plateauing of the training score. This is also the case with the cross-validation score. Another feature of this validation curve is the overfitting resulting from the significant gap between training and cross-validation scores. The fact that this classifier is performing better on the training data but is not generalizing well to the validation data is the cause of this.

The second hyperparameter used to tune this decision tree classifier is the post-pruning parameter cost complexity (ccp_alpha). Testing the decision tree with a ccp_alpha range between 0.0001 and 0.01 is shown in figure 1 (right image). It is clear that the greatest accuracy on the cross-validation data is achieved between alpha=0.002 and 0.0023. Moreover, even though training accuracy has decreased to 0.95 at this ccp_alpha value, this model will now

be more generalized, and it will perform better on unseen data.

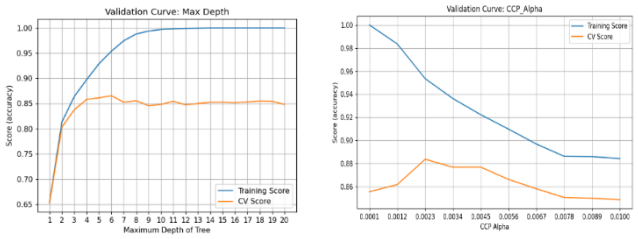

Fig 1. Decision tree hyperparameter tuning: Wine Dataset – Left: Pre-Pruning Max Depth. Right: Post Pruning Cost Complexity (CCP).

Finally, using GridSearchCV to perform an exhaustive search over a specified parameter grid, the best parameters are determined to be {'ccp_alpha': 0.0045000000000000005, 'max_depth': 5}. What is notable here is that the training score decreases, and cross-validation scores increase as more training data is used. The low cross-validation score is a sign that this classifier is oversimplifying the relationships in the data and underfitting the complexity of the problem, also known as having a bias. As a result, it could be said that this classifier, using the parameters determined above, makes predictions that are far from the true values. The learning curve plotted below in figure 2 is a visual representation of this.

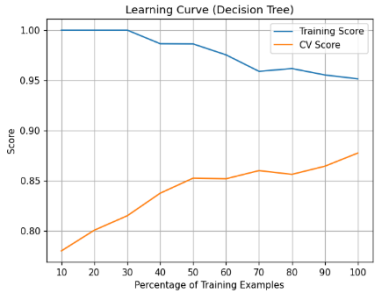

Fig 2. Learning Curve: Wine Data Set

The accuracy of this model increases 2% to 87.25% after tuning.

B. *Breast Cancer Dataset*

Again, a vanilla decision tree is used to determine a baseline accuracy. The decision tree model using the breast cancer dataset, without any tuning, produces an accuracy of 90.35%.

1) Hyperparameter Tuning

Using the same maximum depth range, 1-25, figure 3 below (left image) shows that the decision tree reaches saturation with a max depth of 7. Moreover, like the validation curve generated for the wine origin dataset, overfitting of the data is present. This is evident by the large gap between the training and cross validation scores. The cause of this is that this classifier is suffering from having high variance meaning it is too complex and is fitting the noise in the data as well as the signal. This is seen in the cross-validation scores where there are sharp peaks and then dips in the accuracy.

The second hyperparameter, ccp_alpha, also shows a high bias and overfitting of the data at the limit of the range being tested. Using the same range, 0.0001 to 0.01, note that the training score decreases while the cross-validation score increases very slightly. It is possible that more data and/or removing less important features would help with the bias and overfitting.

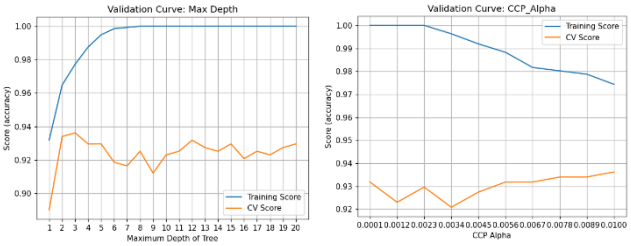

Fig 3. Decision Tree Breast Cancer Dataset – Left: Pre-Pruning Max Depth. Right: Post Pruning Cost Complexity (CCP).

The learning curve below is plotted after GridSearchCV determined the best parameters to use as {'ccp_alpha': 0.0012000000000000001, 'max_depth': 5}. Like the learning curve in figure 2, there is a large gap between cross-validation and training scores, however, it also shows an interesting difference as compared to the learning curve in figure 2.

First, the training score is greater than or equal to 0.99 for the entire set of training examples but the cross-validation score is much lower and ultimately moves away from the training score. As such, this classifier has a low bias, a high variance, and will make predictions that are very close to the true values for the training data. This will result in a training score of 1, or >= 0.99 in this case since the model will perform almost perfectly on the training data.

Secondly, the training and cross validation scores move away from each other in the last 20% of training examples. Where the training score appears to be increasing back to 1, the cross-validation score decreases from its peak of around 0.94. The plot below shows very well that, in the last 20% of training examples, this decision tree has learned the patterns in the training data too well and has started to memorize the training examples instead of generalizing to new data. As a result, this leads to a high variance, where the model performs well on the training data but poorly on the cross-validation data.

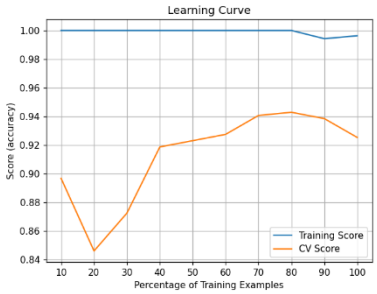

Fig 4. Learning Curve: Breast Cancer Dataset

The accuracy for this decision tree using the parameters taken from GridSearchCV has increased to over 94.00%

IV. Classifier 2: Neural Network

A. *Wine Origin Dataset*

Using a neural network (NN) classifier as the second classifier, the initial NN performed quite well with an accuracy of 97.22%.

1) Hyperparameter Tuning

To tune this NN classifier, hidden layer size and alpha are used. Starting with an alpha of -10, this parameter is tested up to a range of 10. The results in figure 5 below on the left show that when alpha=1, there is a significant drop off in accuracy for both the training and cross-validation scores. This value of 1 represents the point at which this classifier becomes too simple and is not able to capture the underlying patterns in the data. As such, where alpha > 1 results in a decrease in accuracy, as this classifier is not able to accurately capture the relationship between the features and the target. Hence, it begins to underfit the data and result in high bias and low variance and will not fit the training data well.

The second hyperparameter, hidden layer size, is tested with a range of 2 to 30. In general, it can be expected that a hidden layer with a large number of units can capture complex relationships between the inputs and outputs, leading to improved model accuracy. However, a large hidden layer size can also lead to overfitting. The plot on the right in figure 5 below shows this well – The model performs quite well as hidden layer size increases from 2 to about 10 but, after that, this classifiers variance begins to increase and bias decreases. The gap in training and cross-validation scores indicate overfitting, although not much. That being said, in contrast to the alpha parameter tuning, accuracy continues to increase as the units get larger.

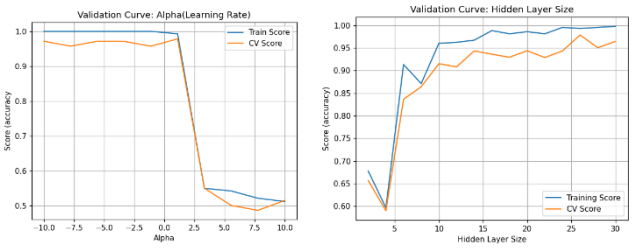

Fig 5. Neural Network hyperparameter tuning: Wine Origin Dataset – Left: Alpha. Right: Hidden Layer Size

Finally, a learning curve (right) and loss curve (left) are plotted in figure 6 below. Using GridSearchCV to determine optimal alpha and hidden layer size, it would appear that the neural network classifier is a good model to use for this multi-class dataset. With the hyperparameter values of {'alpha': 12.915496650148826, 'hidden_layer_sizes': (100,)} very little overfitting is present, and the model performs optimally after around 60% of the training examples are used. Variance in this learning curve is low as well as bias. Moreover, the loss curve shows exponential reduction over 200 epochs, an indication that this model has a very good learning rate and achieves low error values after the first 50% of iterations.

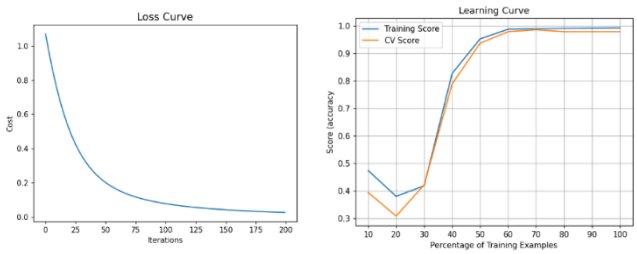

Fig 6. Neural Network, Wine Origin Dataset. Loss curve, left. Learning Curve, right

B. *Breast Cancer Dataset*

The NN classifier fitted with the breast cancer dataset performs slightly under the wine origin dataset. An accuracy of 96.49% is achieved.

1) Hyperparameter Turning

Tuning alpha and hidden layer size using the breast cancer dataset are plotted in figure 7 below. As can be seen in the plot on the left, the tuning of alpha, using the same range used in the wine origin dataset, has a similar effect. By increasing alpha, the model is restricted in a way that reduces the weight of the model. This results in a reduction of overfitting, which can be seen when alpha is equal to about 1. Equally important to note is that an alpha value greater than 1 causes underfitting of the data and accuracy drops significantly, resulting in a high bias and high variance.

The second hyperparameter, plotted on the right in figure 7, is hidden layer size. This parameter was chosen because it has a significant impact on model performance. As it relates to the breast cancer dataset, a hidden layer size less than about 4 causes this classifier to fit the data poorly and be too simple to capture the true relationship between the features and the target indicating it is suffering from high bias and high variance. After a value of 5 is reached, the balance of bias and variance is worked out, and the model performs quite well until a value of about 25 where the gap between training and cross-validation scores begin to increase because of overfitting.

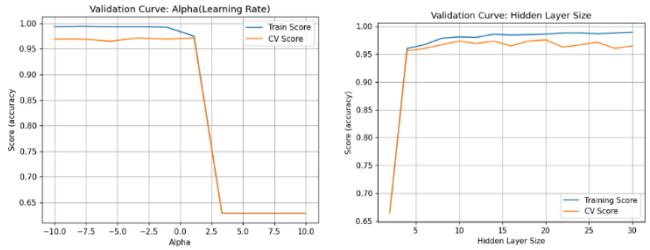

Fig 7. Neural Network hyperparameter tuning: Breast Cancer Dataset – Left: Alpha. Right: Hidden Layer Size

Plotting the loss curve and learning curve show some similarities when compared to the wine origin loss curve, however, the learning curve looks very different than it did using the wine dataset. Using the GridSearchCV parameters {'alpha': 12.915496650148826, 'hidden_layer_sizes': (50, 100, 50)}, what immediately jumps out is that optimal hidden layer size is different between the two datasets. As a result, unlike the wine origin dataset, this learning curve shows low bias and high variance, meaning that it fits the noise in the training data but does not generalize

well to new data. The result is a good performance on the training dataset but poor performance on the cross-validation dataset. Because the cross-validation score is increasing in the last 10% of training examples, it is possible that more data might increase accuracy, reduce overfitting, and balance out bias and variance.

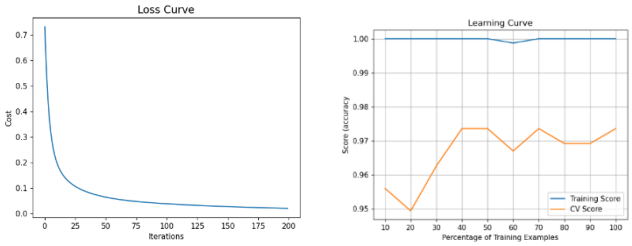

Fig 8. Neural Network – Breast Cancer Dataset. Loss Curve, Left. Learning Curve, Right

The accuracy of this classifier, post GridSearchCV optimization, is close to what it was using the neural network classifier with no tuning.

V. CLASSIFIER 3: BOOSTING

A. *Wine Origin Dataset*

Using a vanilla AdaBoost classifier and the wine origin dataset, this classifier predicts the class value with an accuracy of 89.00%.

1) Hyperparameter Tuning

The first hyperparameter used to tune this classifier is the number of estimators ranging between 10 and 250. Here, more estimators refers to the number of weak learners that are combined to form the final strong learner. A higher number of estimators results in a more complex model, and, as can be seen in figure 9 below (left image), results in a validation curve with low bias and very high variance thus causing declining cross-validation scores and the data to be overfit. This is because when the number of estimators is too large, this classifier is attempting to fit every detail of the training data, even the noise and random fluctuations. The result is an AdaBoost classifier that is very accurate on the training data but has poor generalization performance on cross-validation data.

The second hyperparameter, learning rate, is used to tune the weight given to each weak learner in the final combination of this boosting classifier. The validation curve below (right image) is different than any of the other validation curves previously evaluated as there is no substantial gap between the training and cross-validation scores. The two scores run parallel to each other for the range being tested This is an indication that this classifier is not overfitting the data and that it has a good generalization ability. Observed in the plot being referenced, there is very clearly an ideal learning rate where accuracy is the greatest. The peak is where the learning rate = 0. Also, important to note is the closeness of the training and cross-validation scores. Variance and bias are seemingly mitigated as this point, and the result is a well fit model.

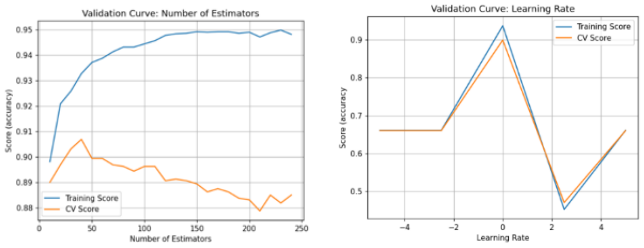

Fig 9. AdaBoost hyperparameter tuning. Wine Origin Dataset: Number of Estimators, Left. Learning Rate, Right

To conclude the AdaBoost classifier testing, a learning curve is plotted below in figure 10 using what GridSearchCV determined to be the optimal parameters as {'learning_rate': 1.0, 'n_estimators': 21}. Because of the modest gap seen between training and cross-validation scores, it could be said that there is some overfitting. However, because both the training and cross-validation scores are high, the gap in these scores is relatively small, and a plateauing of the scores, it can be said that this classifier is performing well on this dataset.

With the optimal parameters, accuracy of this classifier is increased to 92.35%, which is better than the 89% accuracy our vanilla AdaBoost classifier started with.

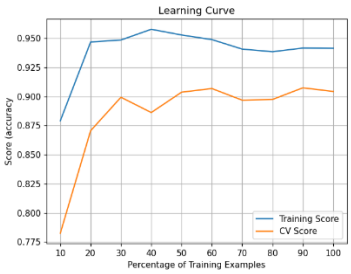

Fig 10. Boosting Learning Curve: Wine Origin Dataset

B. *Breast Cancer Dataset*

This AdaBoost classifier's baseline accuracy comes in at 98.25%. This is significantly better when compared to the wine origin dataset.

1) Hyperparameter Tuning

Using the same hyperparameters as explained above, the number of estimators ranging from 10 to 250 and a learning rate between -5 and 5, what immediately stands out is the change seen in the number of estimators validation curve. The plot in figure 11 below (left image) shows the training score plateauing at around 25 estimators. There is still a significant amount of bias in the plot, and, as a result, a large gap between the cross-validation and training scores. Furthermore, instead of the cross-validation score decreasing, it appears instead to plateau around a score of 0.97 meaning that this classifier's accuracy on the validation set has reached a maximum and is no longer improving.

The learning rate plot below, however, is very similar to the corresponding wine origin plot above. A learning rate between 0 and 2.5 causes a decrease in model performance as the weak learners are given too much weight and their predictions, but recovers where the learning rate is greater than 2.5. Here, the optimal learning rate appears to be 0. Also similar is the fact that the training and cross-validation scores run parallel with each other although a gap in scores is evident.

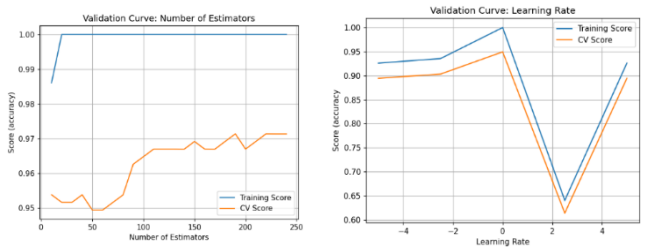

Fig 11. AdaBoost Hyperparameter Turning. Breast Cancer Dataset: Number of Estimators, Left. Learning Rate, Right

Examining the learning curve in figure 12 below, the training and cross-validation scores do not converge at any time using this set of data. Moreover, the fact that this classifiers training score is a 1 throughout the entirety of the training example, and the cross-validation score continuously increases as training examples are used, this classifier has a low bias and low variance. It would appear that this classifier is effectively able to capture the relationship in the data and its probable that, with more data, the training and cross-validation scores would converge. Also, compared to the wine origin learning curve, this learning curve shows it is well-suited to the data.

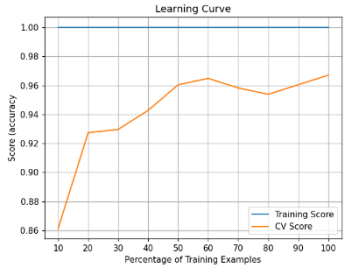

Fig 12. Boosting Learning Curve – Breast Cancer Dataset

VI. CLASSIFIER 4: SUPPORT VECTOR MACHINE (SVM)

A. *Wine Origin Dataset*

Implementing a Support Vector Machine (SVM) with no tuning yields a baseline accuracy of 92.50%.

1) Hyperparameter Tuning

The first hyperparameter used for tuning this SVM is C with a range of 0.001 to 100. The C parameter, or regularization parameter, controls the trade-off between achieving a low training error and a low testing error. As can be seen plotted on the left in figure 13, a smaller value of C (0-10) means higher variance and bias, hence a lower accuracy for training and cross-validation data. As the value of C increases, this classifier has much less evidence of bias and variance. As a result of this observation, this hyperparameters value ultimately affects the model's ability to classify the training data correctly. It can be observed that a smaller value of C results in a higher misclassification rate on the training data, while the larger values of C result in lower misclassification rates on the training data and higher accuracy. Where the C value is above 10 and training and cross-validation scores plateau, indicate this classifier has reached a balance between variance and bias. This is the optimal C value.

Next, gamma is tuned with different kernel values (RBF, Sigmoid, and Poly). Because gamma affects kernels differently, it was applicable to test the same range of gamma values (10^{-6} to 10^{1}) with varying kernels. Visually, how the kernels interpret the value of gamma differently can be seen in the validation curve below (right plot). The validation curve of the RBF kernel and Sigmoid kernel are similar in that scoring is low up to a value between 10^{-3} and 10^{-2}. After which overfitting begins to occur for both kernels. The Poly kernel begins to overfit the data around a gamma of 10^{-1}. What is common with all three kernels tested is that the lower values of gamma produce high bias and high variance, resulting in the lowest training and cross-validation scores. This is most likely attributed to the underlying relationship between the features and the target variable being non-linear. As a result, the model requires a higher gamma to produce a more complex decision boundary to capture the relationships between features and target values

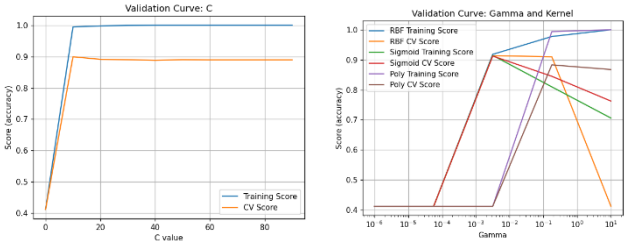

Fig 13. SVC Hyperparameter Tuning. Wine Origin Dataset: C, left. Gamma, right.

Finally, a look at the learning curve for this SVM classifier. Having fit this classifier with parameters defined by a GridSearchCV ({'C': 90.001, 'gamma': 0.003162277660168)3794}, it's important to note that the learning curve for the training and cross-validation scores for this dataset come very close to converging at a high percentage of training examples. This is a good indication that this classifier has found a stable solution and is performing well on this dataset.

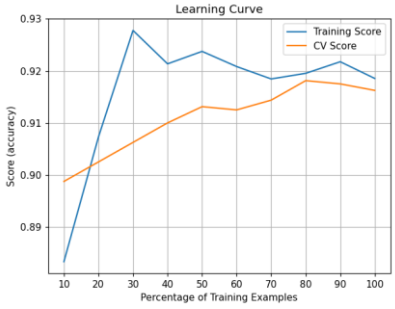

Fig 14. SVM Leaning Curve – Wine Origin Dataset

Accuracy of this model increases by 0.5% using the optimal values defined in the GridSearchCV.

B. *Breast Cancer Dataset*

The same vanilla SVM classifier, using the breast cancer dataset, achieves an accuracy of 93.86%.

1) Hyperparameter Tuning

Using a C value between 0.001 and 100, the validation curve in figure 15 looks very similar to what was produced using the wine origin dataset. To reiterate, the smaller the value of C, the higher bias and variance is, and the more misclassification takes place. As the value of C increases, the bias and variance are reduced, and accuracy increases. Eventually, at about C=12, training and cross-validation scores plateau. This occurs because the C parameter controls the trade-off between achieving a low training error and a low testing error. As a result, as the value of C increases, the model becomes more flexible and complex, which results in a lower training error. However, it should be expected, that beyond a certain point, increasing the value of C will not further decrease the training error.

The second hyperparameter tuned is gamma, again with RBF, Sigmoid, and Poly kernels. The plot in figure 15 looks very similar to what is plotted using the wine origin dataset. All three kernels overfit the data after a gamma reaches a certain value. Most likely, this can be credited to the fact that, as the value of gamma increases, the model becomes more complex. In some circumstances, this can lead to overfitting. As can be seen in the plot below, after a certain point, increasing the value of gamma does not decrease the training error for any of the kernels. Rather, it results in higher cross-validation error rates and overfitting. Again, referencing the plot below, the optimal value of gamma is between 10^{-3} and 10^{-2} for RBF and Sigmoid kernels, and closer to 10^{-1} for Poly.

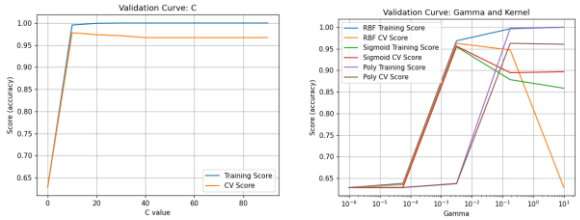

Fig 15. SVC Hyperparameter Tuning. Breast Cancer Dataset: C, left. Gamma, right.

Looking at the learning curve in figure 16 below, it is immediately noticable that the SVM accuracy on this dataset is better than what is plotted in the learning curve from figure 14. Using hyperparamters resulting from GridSearchCV ({'C': 50.001, 'gamma': 0.0031622776601683794}) , there is a large gap between the traning score and cross-validation score observed until about 50% of the traning examples signifying that this might be a more complex dataset, hence this classier needs more time to learn the solution. This in not necessarily a bad thing. Moreover, in the last 10% of the training examples, the training and cross-validation scores are fairly close to eachother. It coud be said that the model has a good balance of low variance and high bias and, as such, not much overfitting is present. This contrasts with the first 10% of the training examples where there is a very large gap and the cross-validation score is significantly lower that the training score.

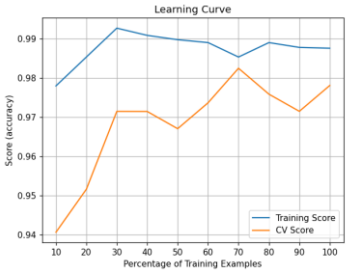

Fig 16. SVM Leaning Curve – Breast Cancer Dataset

VII. Classifier 5: K-nearest neighbors

A. *Wine Origin Dataset*

Using a basic k-Nearest Neighbor (k-NN) classifier, an accuracy of 91.00% is achieved. Using default values to predict the underlying classifier concept that similar samples are likely to belong to the same class, this model performs well on this dataset.

1) Hyperparameter Tuning

The first hyperparameter tuned for this dataset is the number of nearest neighbors ranging between 1 and 300. Understanding that the lower number of neighbors (k) means that the model is highly sensitive to the presence of outliers or noise in the data and a large k value means that the model is less sensitive to outliers but more susceptible to overgeneralizing and not capturing the true pattern in the data, the validation curve pictured on the left in figure 17 below is indicative of this relationship. Once k reaches about 40, this classifier's bias increases and variance decreases. As a result, the training and cross-validation scores decrease continuously until the limit of the range being tested is reached. This is expected behavior as the dataset totals 2000 records, and a k=300 uses about a sixth of the data (15%) when looking at a points nearest neighbors.

Similar to the number of nearest neighbors (k), a small value of p (p=1) means that the distance is more sensitive to the presence of outliers or large differences in the feature values, while a large value of p (p=10) means that the distance is more sensitive to the overall differences in the feature values. Because the value of p controls the relative importance of each feature in the distance calculation, what is observed in the plot below is an expected behavior. This is because this k-NN model is giving more importance to each feature, which is leading to overfitting. Based on the validation curve, a p value of 2 seems best though this dataset is tested on a range from 1 to 10.

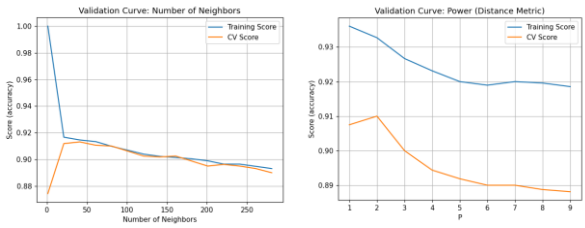

Fig 17. k-NN Hyperparameter Tuning. Wine Origin Dataset: "k", left. "p", right.

Next, GridSearchCV is used to determine the best values of k between 1 and 300, as well as p using values between 1 and 10. Determined to be {'n_neighbors': 21, 'p': 1}, the learning curve below is a good indication that this k-NN classifier is performing relatively well on this dataset. Starting out with low cross-validation and training scores, both increase as the size of training data increases. As the scores increase, bias and variance are decreasing, meaning the this classifier is capturing the relationships in the data and generalizing well to new data. Because both the training and cross-validation scores plateau, it can be assumed that this classifier reaches its maximum performance and that adding more training data will not significantly improve its accuracy. Finally, the accuracy of this classifier increases to 92.50% meaning GridSearchCV was effective and increased this classifier's performance.

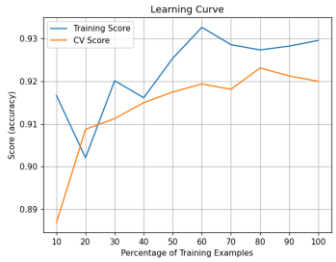

Fig 18. k-NN Leaning Curve – Wine Origin Dataset

B. *Breast Cancer Dataset*

This k-NN classifier performs the best of all the classifiers with no hyperparameter tuning. The accuracy for this k-NN classifier is very high at 96.49%

1) Hyperparameter Tuning

The result of tuning the nearest neighbor (k ranging from 1 and 300) and distance metric (p ranging from 1 to 10) results in very similar validation curves for this breast cancer dataset. What is noticeable about the validation curve for k (left image) is that the training and cross-validation scores are much closer together when the value of k is small. Similar also is the fact that bias increases causing accuracy to drop as the value of k increases beyond 50. This is most likely because the breast cancer dataset is a binary classification problem. Also similar are the scores for p where p is greater than 2. Clearly, the training score decreases where p is greater than 3 and the cross-validation scores drop for p values greater than 1. This is a direct result of the distance metric, p, becoming more sensitive to the differences between the features, and the influence of individual features on the distance calculation becomes stronger where p is > 2. Hence, the model is overreacting to noise in the data and becoming overly complex, resulting in overfitting and poor generalization performance on unseen data.

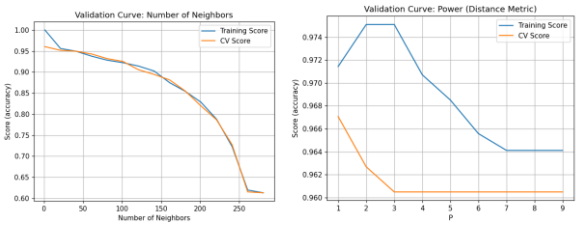

Fig 19. k-NN Hyperparameter Tuning. Breast Cancer Dataset: "k", left. "p", right.

Examining the learning curve last, this k-NN classifier is fitting the training data perfectly and has no errors in its predictions on the training data. However, this does not necessarily mean that this classifier will perform well on new, unseen data, as can be seen in figure 20 below. Because the validation score of 1 is well above the cross-validation scores, this is indicitive of low bias, high variance, and thus overfitting. After GridSearchCV determined the best parameters to be {'n_neighbors': 1, 'p': 2}, the accuracy of this model increases to 97.37%. What is concerning here, and different than the optimal k value for the wine origin dataset, is that a k value of 1 means only the closest neighbor is used for prediction, making the model highly sensitive to noise and outliers in the data. While the tradeoff of having k=1 generally leads to high accuracy, it is oftern a sign of overfitting. Figure 20 shows this well when compared to the learning curve of k-NN using the wine origin dataset.

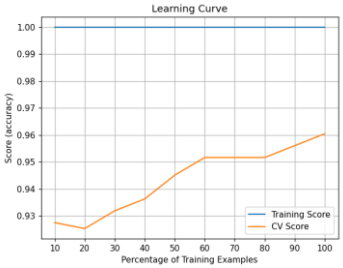

Fig 20. k-NN Leaning Curve – Breast Cancer Dataset

VIII. CLASSIFIER TIME AND ACCURACY ANALYSIS

A. *Training Times*

Referencing figure 21 below, each of the 5 classifiers training times are compared using the wine origin (left image) and breast cancer (right image) datasets. The SVM and decision tree classifiers take the least amount of time to train for both datasets. Regarding the SVM classifier, the minimal training time is most likely attributed to how this classifier utilized a high-dimensional feature space to quickly answer its quadratic programming problem. The decision tree classifier is fast in general, relative to the other classifiers. However, because of the pre pruning using maximum depth for both datasets, this classifier was the second fastest for both datasets. k-NN is in the middle of the pack but takes less than half the amount of time to train compared to the AdaBoost and neural network (NN) classifiers. The increase in training time of k-NN comes from the number of neighbors, k, to be searched and distance metric, p, used for tuning. The AdaBoost and NN classifiers took the most time to train, alternating between which one took

longer on the two datasets. The high training times for AdaBoost are attributed to the fact that multiple weak learners are trained, with each model being trained on the samples that were misclassified by the previous model. And finally, the NN classifier training time is influenced by the number of layers used in tuning as well as the way errors are propagated backward through the network to update the weights and biases of the individual neurons.

Specific to these classifiers and the wine origin dataset, what stands out is that each of the 5 classifiers took longer to train on the wine origin dataset. The wine origin dataset has more than twice the number of records, even though it has half the number of features. As a result, it is safe to say that while both the number of features and the number of records can impact the training time of a machine learning classifier, the number of records in the training set has a greater impact on the training time than the number of features.

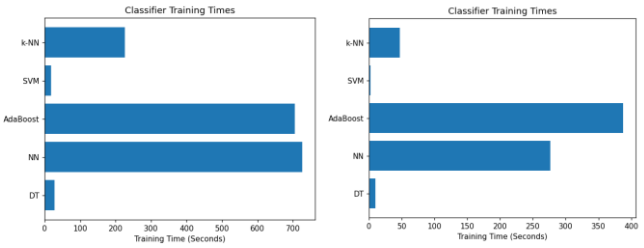

Fig 21 – Classifier Training Times. Left, Wine Origin Dataset. Right, Breast Cancer Dataset

B. *Accuracy*

Figure 22 below is an illustration of the accuracy for each of the 5 classifiers. The accuracies here are taken after being fit with the ideal parameters as suggested by using GridSearchCV.

Interestingly, the decision tree (DT) was the lowest scoring model when looking at both the wine dataset and breast cancer dataset. Looking back at the DT learning curves, this is not entirely unexpected behavior. The DT fitted with the wine origin data set suffered from underfitting, while the DT fitted with the breast cancer dataset had a training score of 1, lower cross-validation scores due to bias, and thus overfitted the data. Both underfitting and overfitting would affect the accuracy. Hence, the fact that the DT has the lowest accuracy is not unexpected.

Next, it can be seen that the AdaBoost accuracy is one of the lower scoring classifiers for the wine origin dataset but achieves the highest accuracy when the breast cancer dataset is used. Interestingly, this AdaBoost implementation uses a decision tree as its base learner. This could explain why the wine origin dataset AdaBoost accuracy is very close to the accuracy seen using the DT with the same dataset. From the breast cancer plot, it must be pointed out again that the breast cancer dataset is a binary classification problem, and the AdaBoost classifier is notoriously effective at solving these types of problems. As a result, the AdaBoost performs much better on the breast cancer dataset than it did on the wine origin dataset. This is seen in the learning curves plotted for these two classifiers as well. Given more data, it is possible the AdaBoost fitted with the breast cancer dataset would improve its accuracy even more.

The neural network (NN) and SVM classifiers have very similar accuracy for both the wine origin dataset and breast cancer dataset, although the breast cancer dataset accuracy is slightly higher. Again, the difference is because the wine origin dataset is a multi-class classification problem whereas the breast cancer dataset is a binary classification problem. Multi-class classification problems are more difficult, not only because there are multiple possible outcomes, but also because the relationships between the classes can be complex and non-linear, making it difficult for the classifier to separate the classes. Reviewing the learning curves for these classifiers, both the NN and SVM classifiers perform better using the breast cancer dataset and are in line with what can be seen in figure 22.

Last is the k-Nearest Neighbor (k-NN) classifier. On both datasets, the k-NN classifier performs well. This is credited to the data in each dataset, however, again because of the binary vs multi-class classification, the k-NN classifier performs better on the breast cancer dataset. Moreover, the fact that this classifier's accuracy is among the highest for each of the datasets is an indication that the hyperparameters used to tune this model were effective. Looking back to the learning curves, the k-NN classifier fitted using the breast cancer dataset would have higher cross-validation accuracy given more data. Fitted with the wine origin dataset, this learning curve shows a well performing model where the training and cross-validation accuracy is within 1% of each other.

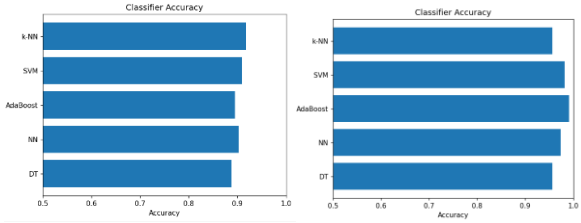

Fig 22 – Classifier Accuracy. Left, Wine Origin Dataset. Right, Breast Cancer Dataset

REFERENCES

[1] Dua, D. and Graff, C. 2019. Breast Cancer Wisconsin (Diagnostic) Data Set, Retrieved 2023-01-24 from https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data.

[2] C.Blake. Sept 21, 1998. Wine recognition data, Retrieved 2023-01-24 from https://archive.ics.uci.edu/ml/datasets/wine