**Supervised Learning Assignment**

Anurag Sahoo  
[asahoo41@gatech.edu](mailto:asahoo41@gatech.edu)

GT ID: 90368005

***Abstract—*** This report deals with the application of different Ma-  
chine Learning Algorithm on 2 different data sets. We would be  
implementing Decision Tree, Adaboost, K Nearest Neighbor, Sup-  
port Vector Machine and Neural Network on the data and study  
their performance across the data. We will also cover how the  
hyper-parameters impact the performance.

**Introduction to Datasets**:

White Wine Quality Dataset: This data set is sourced from UCI Machine Learning Repository. Wine experts are asked to grade the quality of wine on a scale of zero (very bad) to ten (very good). Various attributes of wine including alcohol, pH, density and few others is recorded for the wine that is tested. The data comprises 4898 examples and 11 attributes. The dependent variable is the quality of the wine. All the attributes of the data are numeric. I converted the regression problem into classification by labeling the quality >= 6 as 1 (indicating good taste) and < 6 as 0 (indicating bad quality). The class distribution after re-labeling is as follows: 66.52% good quality wine (1) and 33.48% bad quality wine (0). Since the attributes/independent variables are numeric and all of them would possess different scale (or range) at which they can lie, we need to scale the value of these attributes. I used the Standard Scaler which standardizes by removing the mean and scaling to unit variance. The data was split into training and test set in the 70:30 ratio. All the analysis including validation curves, learning curves and hyperparameter tuning was done on the 5-fold cross validation sets of the training data.

Breast Cancer Wisconsin (Diagnostic) Dataset: The second dataset is also sourced from UCI Machine Learning Repository. The examples are collected from the University of Wisconsin Hospitals, dealing in with the results from the diagnosis which are to be used to indicate if the person is diagnosed with Breast Cancer or not. The original dataset has 699 examples and 10 attributes excluding the predicted column (class). Class 2 indicates benign and 4 indicates malignant. I re-labelled the class 2 as 0 indicating absence of cancer cells and class 4 as 1 indicating presence of cancer cells. The class distribution is as follows: 65.52% absence of cancer cells (0) and 34.48% presence of cancer cells (1). The column “Bare Nuclei” had 16 missing instances. Since, this is breast cancer problem and we do not want our model to wrongly classify, I didn’t impute the missing values of “Bare Nuclei”. Instead, I dropped those 16 examples. Few of the attributes of the diagnosis include Clump Thickness, Uniformity of cell size and shape, Mitoses, Bland chromatin and few more. All these features are numerical and are on different scale. Hence, normalized the features to bring it to a similar scale. I used the Standard Scaler in this data as well. The data was split into training and test set in the 80:20 ratio. All the analysis including validation curves, learning curves and hyperparameter tuning was done on the 5-fold cross validation sets of the training data.

**Why these datasets are interesting?**

These datasets are interesting because they solve different real-world problems. One is related to assessing the quality of wine based on different features. The other is pertaining to the prediction of breast cancer based on the diagnosis data.

These two problems are especially interesting because of the type of data. Wine quality dataset has substantial number of examples and 11 quantitative features. Whereas the Breast Cancer data has limited number of examples. The limited examples would invite overfitting to this problem. We will look at ways to combat this problem. Both the datasets are not perfectly balanced, but they are not highly imbalanced. Also, because of the different type of problem statement, we would look at different performance metrics and try to optimize it. Wine quality problem involves classifying the quality of wine as good or bad. So, using balanced accuracy as a metric would work fine. When working on an imbalanced dataset where positives are as important as negatives, balanced accuracy is a better metric over F1 score. In case of Breast Cancer problem, we would not want the breast cancer actual positive cases to go unnoticed. Hence, we would be using recall as the metric to optimize. Because of the difference in size of data, some algorithms would perform comparatively, and some would take more time to train. We will analyze the performance of different algorithms on these datasets.

**Decision Trees**:

Decision Tree is an extremely intuitive and easy to use algorithm, but it is prone to overfitting. The default Decision Tree classifier from sklearn assumes “None” max depth, meaning it will go on to build tree until it classifies all the examples correctly, leading to overfitting.

Chart, line chart

Description automatically generatedChart, line chart

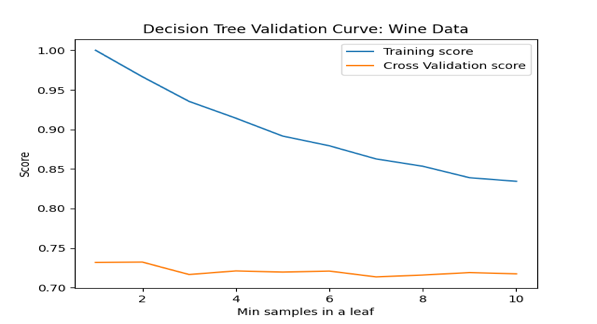
Description automatically generated

Fig 1: Validation curve on max depth

For the validation curves, we had used Balanced Accuracy and Recall as the scoring criteria for Wine and Breast Cancer dataset respectively. In both the data, after a certain value of max depth, the training score approaches 1 and the cross-validation score saturates, indicating that it is overfitting. The Decision Tree learns every intricate detail of the training data and expands tree by increasing depth of the tree. As a result, it performs exceedingly well on the training data but loses the capability to generalize and performs poorly on the cross-validation data. As per the above chart, max depth of 2 and 3 for Wine and Breast Cancer data respectively looks to be a good choice.

The above validation curve was on minimum samples in a leaf parameter. Low value of “min\_samples\_leaf” means the tree has the full capability to grow its branches and increase bias and overfit. High value of “min\_samples\_leaf” means the tree has limited capability to grow its branches and would suffer from underfitting. The “min\_samples\_split” which indicates the minimum number of examples in order to consider a further split, behaves like “min\_samples\_leaf”. High value of “min\_samples\_split” would induce high variance and lead to underfitting. Low value of “min\_samples\_split” would induce high bias and lead to overfitting. High value of “min\_samples\_leaf” and “min\_samples\_split” ensures the model is generalized. Because of the limitation is examples of the Breast Cancer data, high value of “min\_samples\_leaf” and “min\_samples\_split” gives good validation scores. As the Wine Quality data has sufficient examples, high value of “min\_samples\_leaf” and “min\_samples\_split” increases the variance, hence cross validation scores dips slightly.

Cost complexity pruning path is a technique Decision Tree uses to prune a tree. From the plots, we see that for ccp\_alpha value between 0.001 and 0.006 in wine data and for value between 0.01 and 0.05 in breast cancer data, we get maximum validation scores. However, the train scores have reduced but model is now more generalized.

Chart, line chart

Description automatically generated

Fig 2: Validation curve on minimum samples in a leaf

Chart, line chart

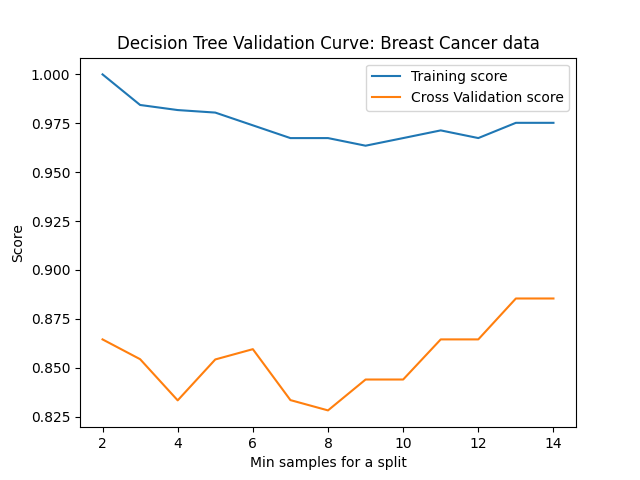
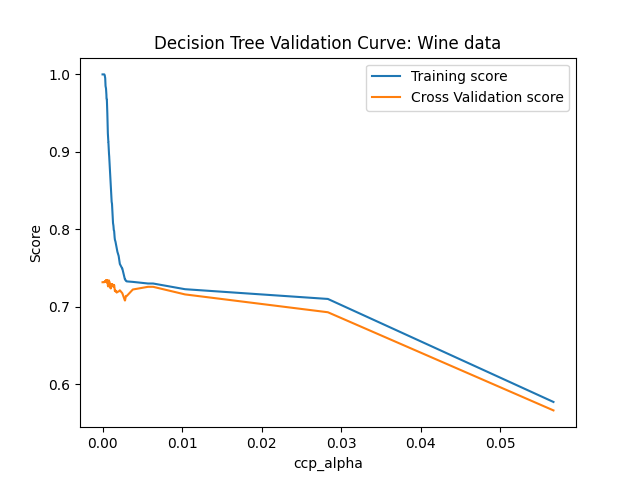
Description automatically generated

Fig 3: Validation curve on minimum samples for a split

Chart, line chart

Description automatically generated

Fig 4: Validation curve on ccp\_alpha

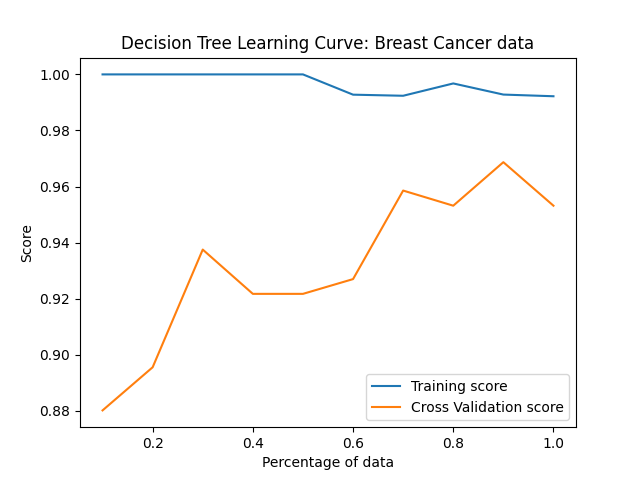
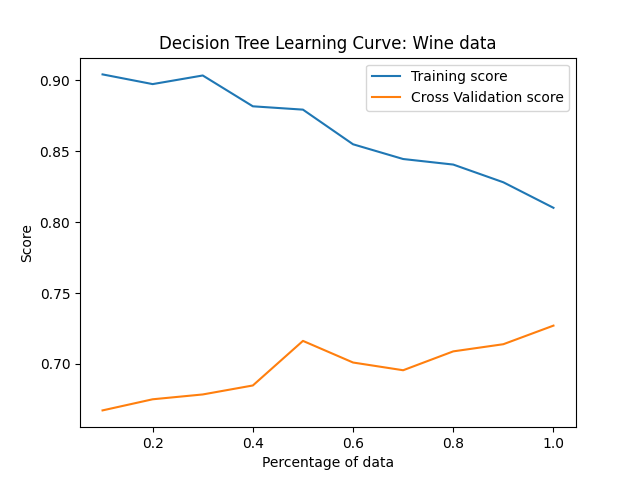


Fig 5: Learning curves

The best hyperparameters were selected using GridSearchCV and learning curve was generated. In case of wine dataset, the cross-validation score is low up until 40% of training data. With more training data, the training score is dipping, and the cross-validation score is increasing. But we see more variability around training scores. So, the model is possibly suffering from error due to bias. Both the curves haven’t converged yet, so they would benefit from more training data. In case of Breast Cancer data, the cross-validation score is low until 60% of training data and shows more variability compared to training scores. The model suffers from error due to variance and possibly overfitting. The cross-validation and training scores have not converged yet, so they would benefit from more training data. The classification report is as follows:

Wine dataset classification report: Breast Cancer dataset classification report:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-score | Support |
| 0 | 0.61 | 0.65 | 0.63 | 466 |
| 1 | 0.83 | 0.81 | 0.82 | 1004 |
| accuracy |  |  | 0.76 | 1470 |
| Macro avg | 0.72 | 0.73 | 0.72 | 1470 |
| Weighted avg | 0.76 | 0.76 | 0.76 | 1470 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-score | Support |
| 0 | 0.98 | 0.95 | 0.96 | 93 |
| 1 | 0.89 | 0.95 | 0.92 | 44 |
| accuracy |  |  | 0.95 | 137 |
| Macro avg | 0.94 | 0.95 | 0.94 | 137 |
| Weighted avg | 0.95 | 0.95 | 0.95 | 137 |

**Boosting**:

AdaBoost algorithm using a decision stump with “max\_depth” as 1 was implemented and validation curves were generated for “n\_estimators” and “learning\_rate”. The reason for using a decision stump of “max\_depth” as 1 is we want to have the base classifier simple and then convert this weak learner to strong learner.

Chart, line chart

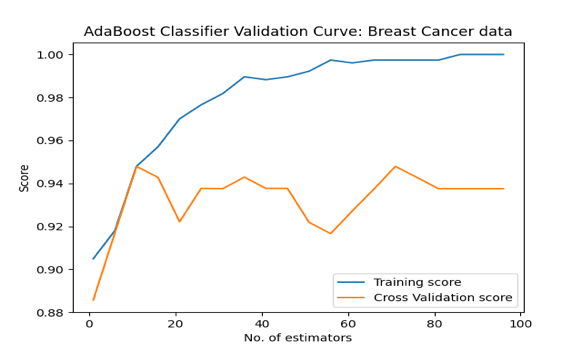
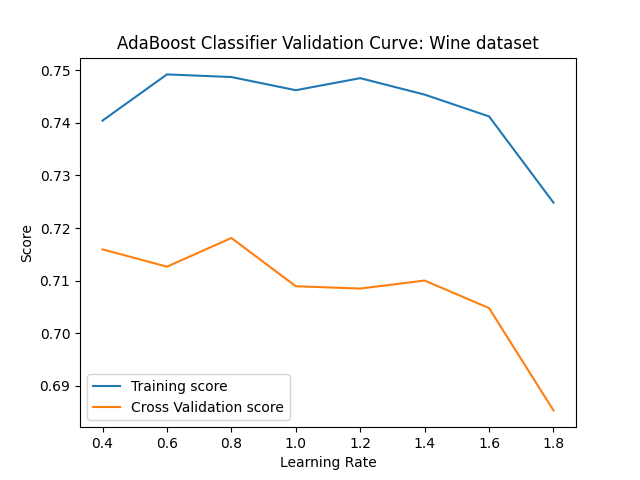
Description automatically generated

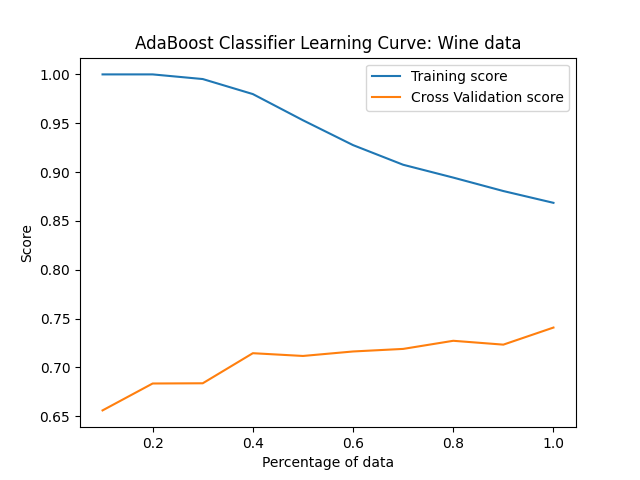
Fig 6: Validation curves on number of estimators

Chart, line chart

Description automatically generated

Fig 7: Validation curves on learning rate

“n\_estimators” indicates the maximum number of estimators after which boosting is terminated and “learning\_rate” indicates the weight assigned to each classifier at each boosting iteration. With higher value of “n\_estimators” the training and cross-validation score keeps on diverging, indicating that the model is overfitting. Extremely low value of “n\_estimators” mean that the model hasn’t learned enough and is underfitting. For an optimal range of “n\_estimators”, we see that the training and cross-validation scores are close, and those values can be chosen for the final model. Learning rate controls how much of contribution to be made with the new classifier to the existing one. There is a tradeoff between number of estimators and learning rate. Extremely low learning rate would need more iterations to converge and is likely to overfit. High learning rate would need lesser iterations and might induce variance in the model.

Chart, line chart

Description automatically generated

Fig 8: Adaboost Learning Curve

Wine dataset classification report: Breast Cancer dataset classification report:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-score | Support |
| 0 | 0.61 | 0.67 | 0.64 | 449 |
| 1 | 0.85 | 0.81 | 0.83 | 1021 |
| accuracy |  |  | 0.77 | 1470 |
| Macro avg | 0.73 | 0.74 | 0.74 | 1470 |
| Weighted avg | 0.78 | 0.77 | 0.77 | 1470 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-score | Support |
| 0 | 0.99 | 0.99 | 0.99 | 90 |
| 1 | 0.98 | 0.98 | 0.98 | 47 |
| accuracy |  |  | 0.99 | 137 |
| Macro avg | 0.98 | 0.98 | 0.98 | 137 |
| Weighted avg | 0.99 | 0.99 | 0.99 | 137 |

Again, best hyper-parameters were selected using GridSearchCV and learning curves were generated. In the Wine data, the training score and cross-validation score haven’t converged yet. Hence, more data would be beneficial for the model. We do see high variability across both training and cross-validation score, possibly the model is suffering from error due to both bias and variance. In the Breast Cancer data, the cross-validation score is much more volatile, and the training score is steadily good. It is possibly suffering error due to variance, and the model is prone to overfitting. The cross-validation score is converging with the training score, so more data may not be useful for the model. The classification report is as follows:

**K Nearest Neighbor**:

K Nearest Neighbors algorithms with a default 5 nearest neighbors was implemented. Validation curves for “n\_neighbors” and “p” were generated.

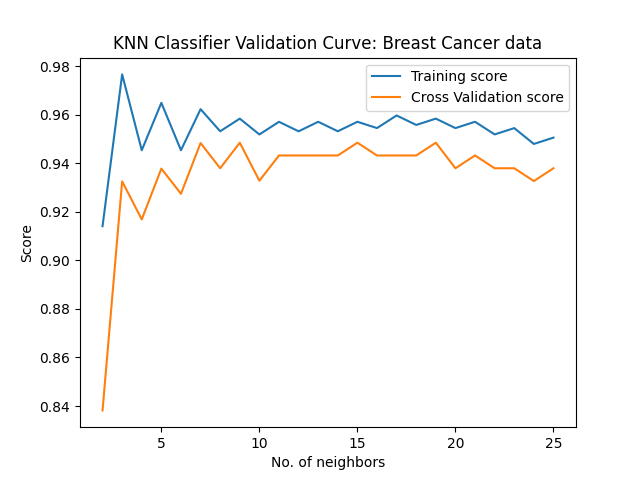
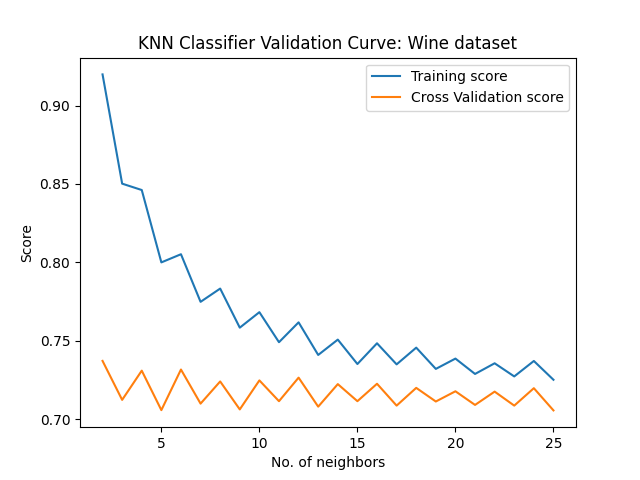
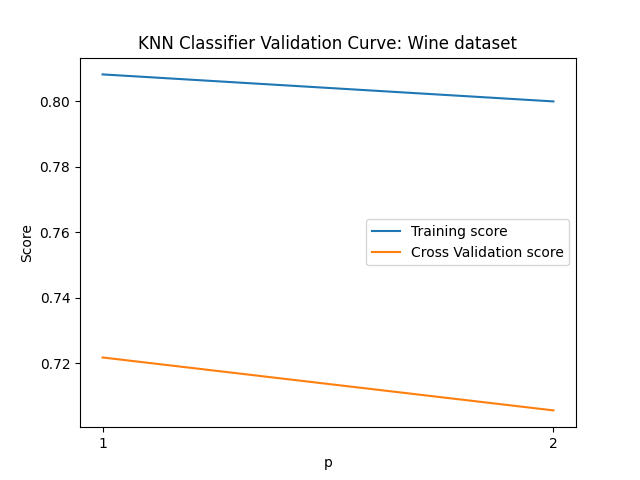


Fig 9: Validation curve on number of nearest neighbors

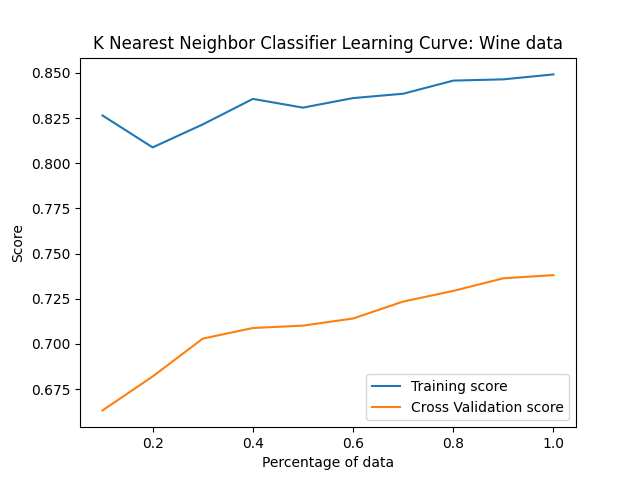
Low value of “n\_neighbors” means considering less nearest neighbors to classify; would be more volatile. Hence, the cross-validation and training score at low values of “n\_neighbors” are substantially separated. With increasing “n\_neighbors”, model learns to generalize and hence the curves are converging.

Chart, line chart

Description automatically generated

Fig 10: Validation curve on p (distance metric)

“p” value of 1 and 2 corresponds to Manhattan distance and Euclidean distance respectively. The gap between the training and cross-validations scores for Manhattan distance looks to be relatively low than Euclidean distance.

Chart, line chart

Description automatically generated

Fig 11: KNN Learning Curve

In Wine data, there might be an error due to variance, as we see more variability around cross-validation score. As far as model is concerned, model is not learning well as the training and cross-validation curves do not seem to converge. In Breast Cancer data, the huge difference in training and cross-validation scores suggest overfitting until 20% of training data. Post that the model seems to be learning well as the training and cross-validation scores seem to be converging. In case of Breast Cancer data, all the metrics show 100% as the test size is small. More data would not be required. The classification report is as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-score | Support |
| 0 | 1 | 1 | 1 | 90 |
| 1 | 1 | 1 | 1 | 47 |
| accuracy |  |  | 1 | 137 |
| Macro avg | 1 | 1 | 1 | 137 |
| Weighted avg | 1 | 1 | 1 | 137 |

Wine dataset classification report: Breast Cancer dataset classification report:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-score | Support |
| 0 | 0.71 | 0.6 | 0.65 | 583 |
| 1 | 0.76 | 0.84 | 0.8 | 887 |
| accuracy |  |  | 0.74 | 1470 |
| Macro avg | 0.73 | 0.72 | 0.72 | 1470 |
| Weighted avg | 0.74 | 0.74 | 0.74 | 1470 |

**Support Vector Machine**:

Support Vector Machine is a supervised algorithm that uses a hyperplane as a decision boundary to separate two classes.

Chart, line chart

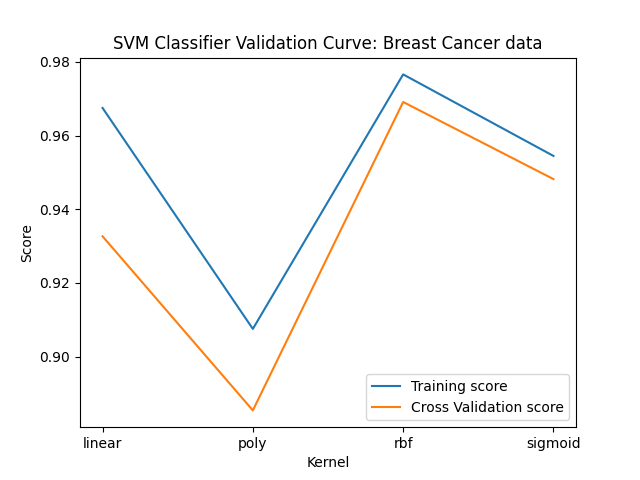
Description automatically generated

Fig 12: Validation Curve on kernel type

Chart, line chart

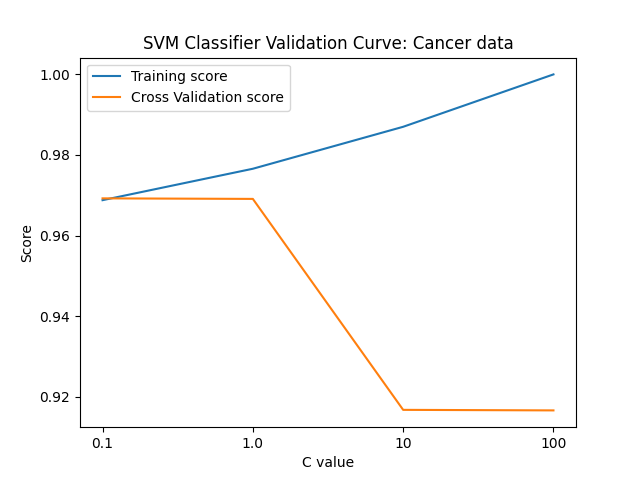
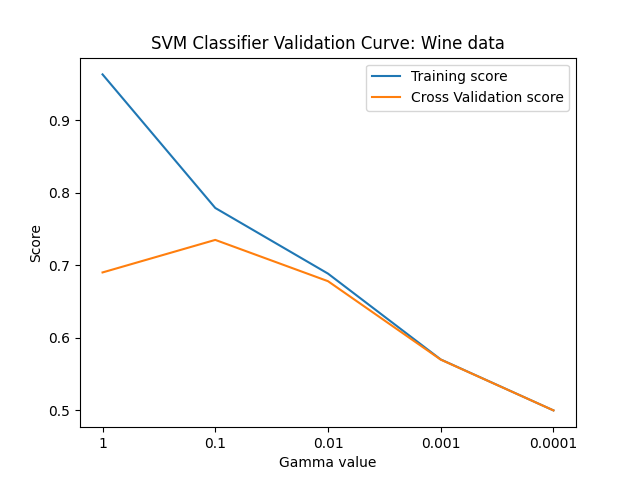
Description automatically generated

Fig 13: Validation Curve on “C”

Chart, line chart

Description automatically generated

Fig 14: Validation Curve on “gamma”

From the kernel type validation curves, it is evident that radial basis function (rbf) kernel is giving us better results in both the datasets indicating that the problems are non-linear in nature. Although, in wine dataset the training and cross-validation score are quite separable indicative of overfitting. “C” is the regularization parameter or the penalty parameter of the error term. It tells how much we want to avoid misclassifying each example. For large values of C, it will choose a smaller margin trying to get all the examples classified correctly, thereby leading to overfitting. This statement becomes evident from the validation curve as there is a big gap between the training and cross-validation score for high value of C. C value of 1 and 0.1 for wine and breast cancer data would be a good choice. Low value of “gamma” indicates that the model cannot capture the complexity of the data leading to an underfit model. Very high value of gamma would lead to an overfit model. This is evident from the Wine dataset validation curves where the training and cross-validation score is distantly different. Varying the gamma in Breast Cancer data doesn’t make much difference because of the size limitation and it drops significantly at the value of 0.0001. “gamma” value of 0.01 would work best in both the datasets.

Chart, line chart

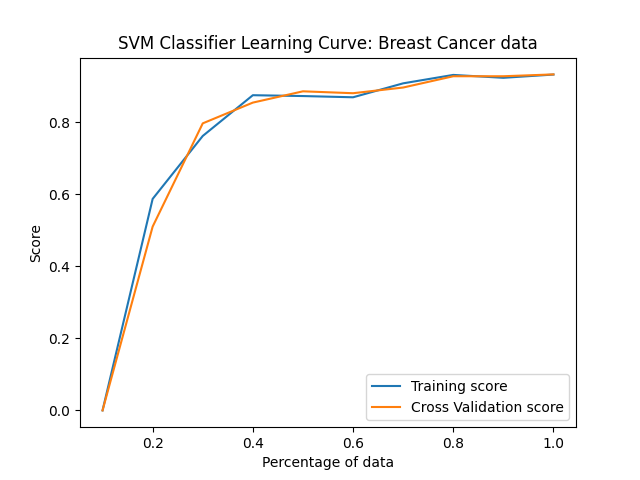
Description automatically generated 

Fig 15: SVM Learning Curves

Learning curve says similar story for each of the dataset. In the Wine data, both the training and cross-validation curve is improving until 40% of data. There might be presence of error due to both bias and variance. However, after 60% of the training data, the curves are trying to converge. More data would be helpful for a good fit. In the Breast Cancer data, the training and cross-validation curves are moving in tandem. The model is not learning enough until 20% of the data. Post that the model is learning well, and the curves are converging. So, the model is well fit. The classification report is as follows:

Wine dataset classification report: Breast Cancer dataset classification report:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-score | Support |
| 0 | 0.5 | 0.71 | 0.59 | 348 |
| 1 | 0.9 | 0.78 | 0.83 | 1122 |
| accuracy |  |  | 0.76 | 1470 |
| Macro avg | 0.7 | 0.75 | 0.71 | 1470 |
| Weighted avg | 0.8 | 0.76 | 0.78 | 1470 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-score | Support |
| 0 | 0.99 | 0.99 | 0.99 | 90 |
| 1 | 0.98 | 0.98 | 0.98 | 47 |
| accuracy |  |  | 0.99 | 137 |
| Macro avg | 0.98 | 0.98 | 0.98 | 137 |
| Weighted avg | 0.99 | 0.99 | 0.99 | 137 |

**Neural Network**:

A neural network with 2 hidden layers is built using MLPClassifier with a ‘relu’ activation function and ‘adam’ solver for weight optimization.

Chart, line chart

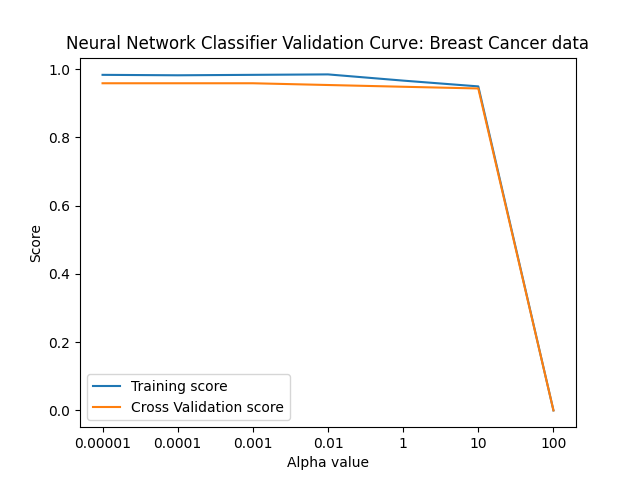
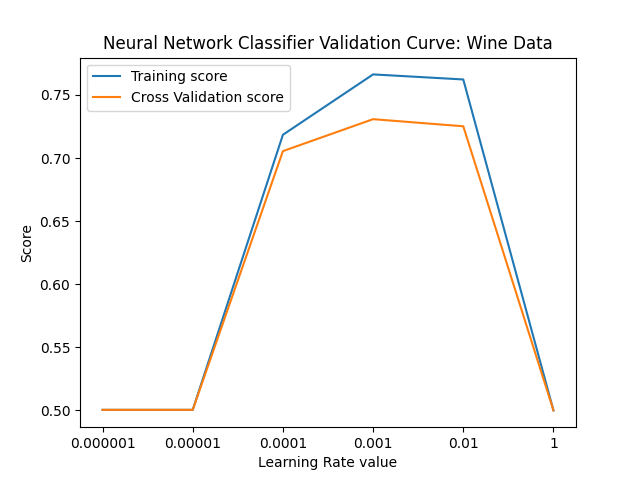
Description automatically generated

Fig 16: Neural Network Validation curve for alpha

Chart, line chart

Description automatically generated

Fig 17: Neural Network Validation curve for learning rate

Chart, line chart

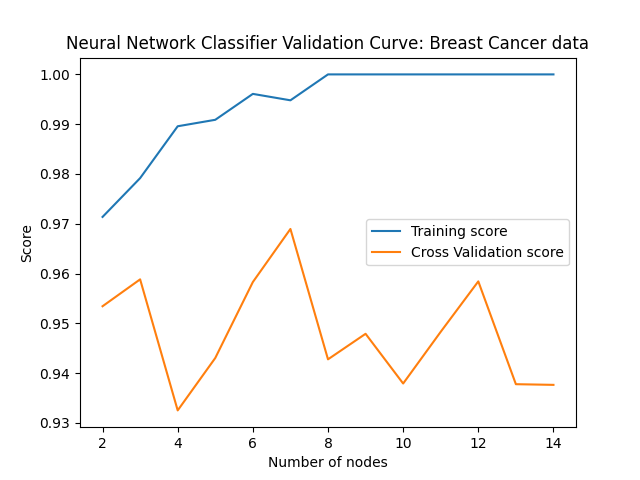
Description automatically generated

Fig 18: Neural Network Validation curve for hidden layer size

“alpha” refers to the L2 regularization component which helps in dealing with overfitting problem. At very low value of alpha, we see a decent gap between the training and cross-validation score, a sign of overfitting. Very high value of regularization would make the model not learn much and underfit. “learning\_rate\_init” controls the step size in updating weights. Very high value of “learning\_rate\_init” would update the weights vigorously and might miss the global minima of the cost function. Very low value of “learning\_rate\_init” would update the weights very slowly and might not reach the global minima before the “max\_iter”. In both the cases, the error is too high. “hidden\_layer\_sizes” represents the number of nodes in the hidden layer. High value of “hidden\_layer\_sizes” would add more complexity to the model and overfit. Hence, we see huge gap between training and cross-validation scores at high value of “hidden\_layer\_sizes”. Low value of “hidden\_layer\_sizes” would underfit the model. Hence, the training and cross-validation scores are low at low value of “hidden\_layer\_sizes”. The hyperparameters were tuned using GridSearchCV and learning curves were generated.

Chart, line chart

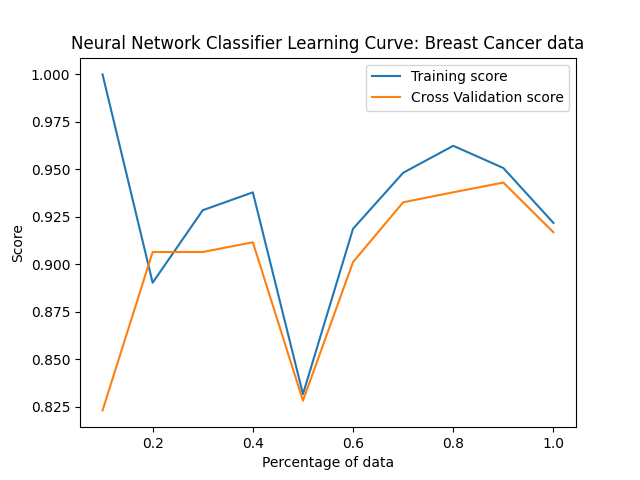
Description automatically generated

Fig 19: Neural Network Learning Curves

In the Wine data, the training error shows more variability and there might be error due to bias. With more training data, the training and cross-validation curves are trying to converge. Hence, more data would be helpful to get a good fit model. In Breast Cancer data, for the initial 20% of the data training and cross-validation score is wide apart because of overfitting. The scores dipped at 50% of the training data. Although, the train and cross-validation scores are moving in tandem, they are highly volatile. There might be an error due to both bias and variance. Although the scores are high, but the model lacks the capability to generalize. Thus, Neural Networks might fail to perform well on the Breast Cancer data because of the low dataset size.

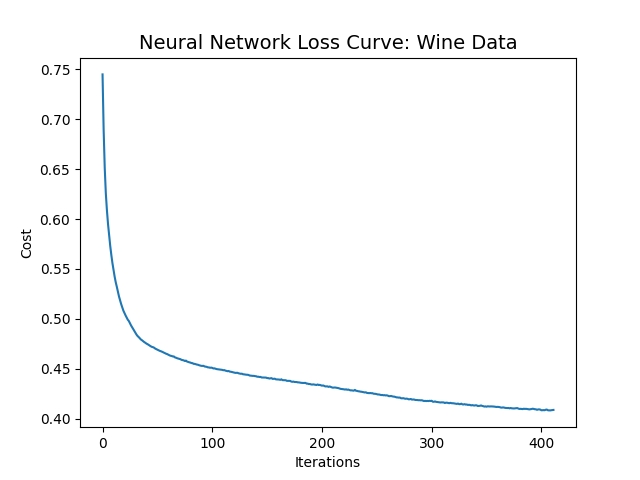
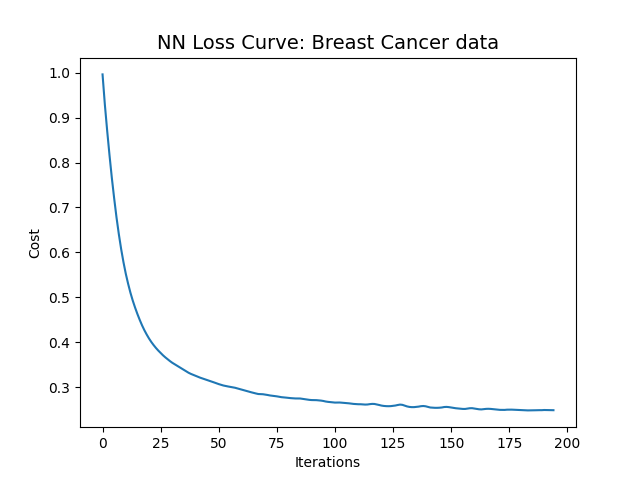
 

Fig 20: Neural Network Loss Curves

In both the datasets, the loss is decreasing with iterations. Thus, confirming that our model is learning well. The classification report is as follows:

Wine dataset classification report: Breast Cancer Classification report:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-score | Support |
| 0 | 0.63 | 0.67 | 0.65 | 466 |
| 1 | 0.84 | 0.82 | 0.83 | 1004 |
| accuracy |  |  | 0.77 | 1470 |
| Macro avg | 0.74 | 0.75 | 0.74 | 1470 |
| Weighted avg | 0.78 | 0.77 | 0.78 | 1470 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-score | Support |
| 0 | 0.99 | 0.99 | 0.99 | 90 |
| 1 | 0.98 | 0.98 | 0.98 | 47 |
| accuracy |  |  | 0.99 | 137 |
| Macro avg | 0.98 | 0.98 | 0.98 | 137 |
| Weighted avg | 0.99 | 0.99 | 0.99 | 137 |

**Comparison of Models**:

Performance Metric Comparison: Training Time (in seconds):

|  |  |  |
| --- | --- | --- |
| Classifier | Balanced Accuracy for Wine dataset | Recall for Breast Cancer dataset |
| Decision Tree | 0.719 | 0.95 |
| AdaBoost | 0.731 | 0.98 |
| KNN | 0.732 | 1.00 |
| SVM | 0.699 | 0.98 |
| Neural Network | 0.739 | 0.98 |

|  |  |  |
| --- | --- | --- |
| Classifier | Wine dataset | Breast Cancer dataset |
| Decision Tree | 0.008285 | 0.046085 |
| AdaBoost | 0.410329 | 0.036441 |
| KNN | 0.006688 | 0.003514 |
| SVM | 0.339593 | 0.015871 |
| Neural Network | 2.531 | 0.157523 |

For Wine dataset, which had sufficient training examples, Neural Network performed the best of all. In fact, Neural Network, KNN and AdaBoost classifier give similar results. SVM, however, gives us the worst balanced accuracy. The learning curves for SVM and Decision Tree show that we need more data for better convergence.

For Breast Cancer data, which is a small dataset, almost all the classifiers are converging and are giving good results.

Neural Network always takes more time to train because of its forward and backward propagation. AdaBoost classifier also takes substantial time because of the iterations from weak learner to strong learner involved. Training SVMs require quadratic optimization. SVM comparatively takes more time on larger dataset than smaller dataset. KNN is extremely fast to train as it needs to arrange the nearest neighbors only during the training time. Decision Tree also takes relatively lesser time to train because of pruning.

Prediction/Querying Time (in seconds):

|  |  |  |
| --- | --- | --- |
| Classifier | Wine dataset | Breast Cancer dataset |
| Decision Tree | 0.000924 | 0.00263 |
| AdaBoost | 0.018364 | 0.00261 |
| KNN | 0.103248 | 0.007077 |
| SVM | 0.239129 | 0.004744 |
| Neural Network | 0.002955 | 0.000367 |

Neural Networks are extremely fast to query when the weights are calculated. Decision Trees are also fast to query when we know the tree traversal. SVM and KNN take considerable time to query. SVM would take more time to query if the size of data is large. KNN takes longer time to query as it needs to find the nearest neighbor. Because of the number of iterations, the AdaBoost would take more time than the Decision Tree classifier.

**References**:

1. UCI Machine Learning Repository – Wine Quality Data Set: https://archive.ics.uci.edu/ml/datasets/wine+quality

2. UCI Machine Learning Repository – Breast Cancer Wisconsin (Diagnostic) Data Set : https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(diagnostic)

3. Learning Curve: <https://www.scikit-yb.org/en/latest/api/model_selection/learning_curve.html>

4. sklearn: https://scikit-learn.org/stable/