Supervised Learning

Daniel Bach
dbach7@gatech.edu

1 DATASETS

Two datasets will be examined in this analysis: the 'Breast Cancer Wisconsin (Diagnostic)' dataset and the ' Iris Species' dataset.

1.1 Breast Cancer Wisconsin (Diagnostic) Data

This dataset contains measurements computed from a digital image of breast mass. It can be downloaded at [1]. The measurements represent characteristics of the cell nuclei which are present in the digital image. The labels which are included in the data are: 'M', meaning malignant (tumor), and 'B', meaning benign (normal). This is a binary classification problem. Diagnosing breast cancer from images using machine learning would be helpful to reduce time needed by the doctors to take a look at the images and make their diagnosis. Since there are many features included in this dataset, the models should theoretically be more complex.

1.1.1 Data Processing

There are 569 rows of data. This dataset contains 32 columns, including the label. The "id" and the "Unnamed: 32" columns were removed before starting the analysis. Unique identifiers such as "id" could potentially mislead machine learning models as they are scalar values. The "Unnamed: 32" column only contained NaN values. Lastly, labels were encoded to either be 0 (benign) or 1 (malignant). The data was split as 70% training data and 30% testing data.

1.2 Iris Species Data

This dataset is very well known and can be downloaded at [2]. Measurements describing irises are included in this dataset as well as the labels. This is a multi-classification problem as there are three classes: Iris Setosa, Iris Versicolour, and Iris Virginica. This problem is interesting as it should be more difficult to classify data when there are more labels that are possible.

1.2.1 Data Processing

There are 150 rows of data. This dataset contains 6 columns, including the label. The "id" column was removed due to the same reason above for the breast cancer dataset. The data was split as 70% training data and 30% testing data.

2 BREAST CANCER DATASET ANALYSIS

2.1 Decision Tree

The decision tree model used for analysis is scikit-learn's DecisionTreeClassifier [3]. The default model uses the GINI index to measure the quality of the splits between nodes.

2.1.1 Base Model Results

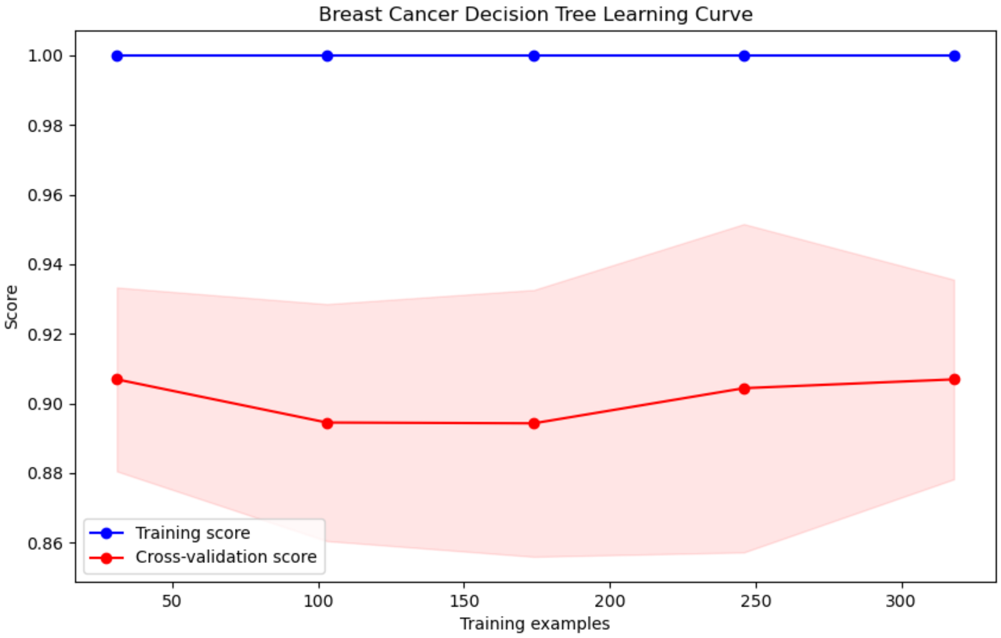

Figure 1—Learning curve for DecisionTreeClassifier

As expected, the decision tree is able to achieve 100% accuracy on the data it is trained on. This is expected because the decision tree is able to split the data continually until all leaf nodes contain one label. Pruning will be used later to see how the model performs when not overfit to the training data.The decision tree was able to gain better accuracy as more training examples were introduced when cross-validating.

2.1.2 Tuning

Tuning for all models is done using scikit-learn's GridSearchCV. GridSearchCV allows us to input different combinations of parameters into the model and find the combination that performs the best (determined via accuracy on training data).

The best combination of parameters found for the DecisionTreeClassifier on the breast cancer dataset were:

{'criterion':'entropy', 'max_features': 'log2', 'min_samples_leaf': 2, 'min_samples_split': 2}

This model was able to achieve 91.228% accuracy on the test data.

2.2 Neural Network

The neural network model used for this analysis is scikit-learn's MLPClassifier [4].

2.2.1 Base Model Results

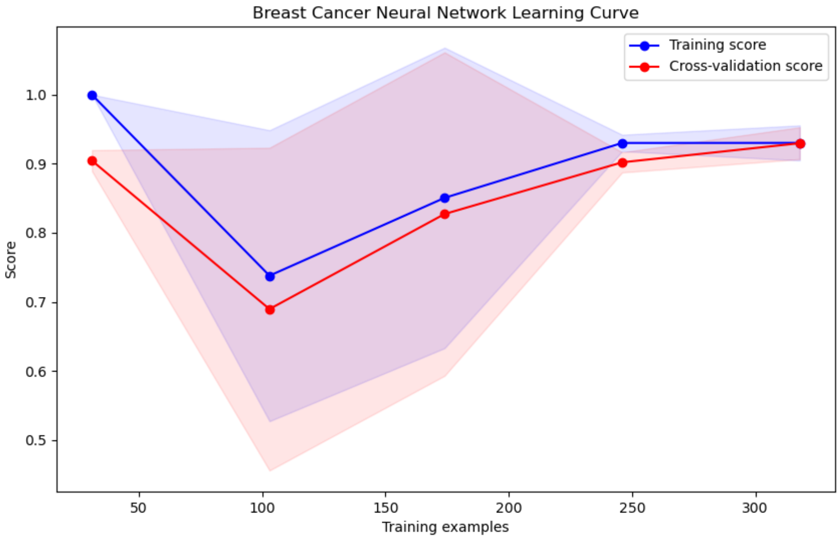

Figure 2—Learning curve for MLPClassifier

2.2.2 Tuning

The best combination of parameters found for MLPClassifier were:

{'alpha': 0.0001, 'hidden_layer_sizes': (50,), 'learning_rate': 'constant'}

The model with these parameters achieved a 95.322% accuracy on the test data.

2.3 k-Nearest Neighbors

The k-nearest neighbors model used for this analysis is scikit-learn's KNeighborsClassifier [5].

2.3.1 Base Model Results

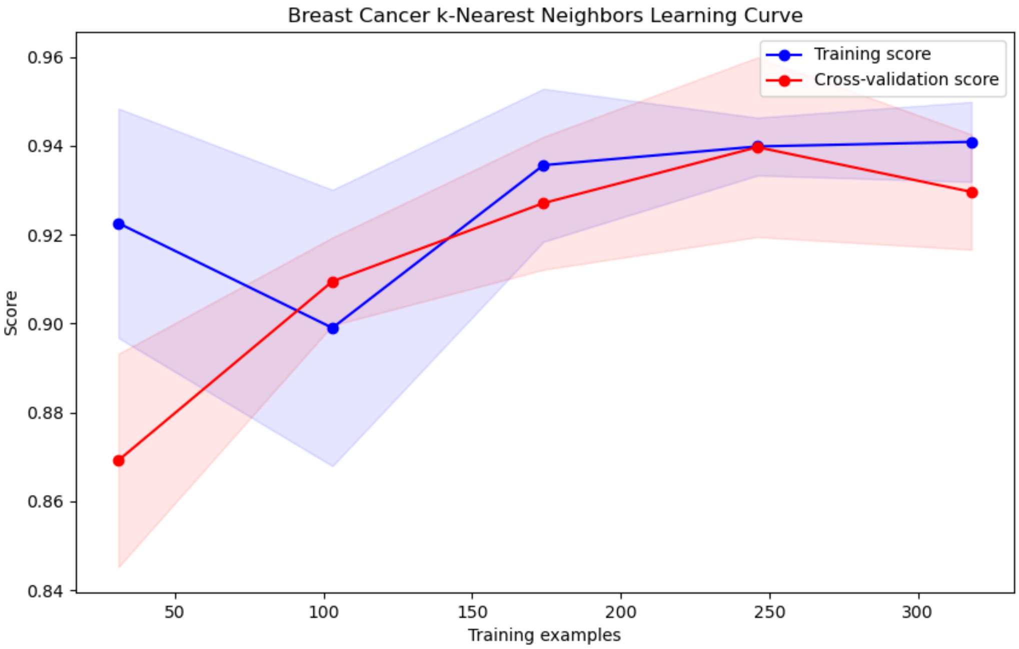

Figure 3—Learning curve for KNeighborsClassifier

2.3.2 Tuning

The best combination of parameters found for KNeighborsClassifier were:

{'n_neighbors': 7, 'p': 1, 'weights': 'uniform'}

The model with these parameters achieved a 95.322% accuracy on the test data.

2.4 Boosting

The boosting model used for this analysis is scikit-learn’s ADABoostClassifier [6].

2.4.1 Base Model Results

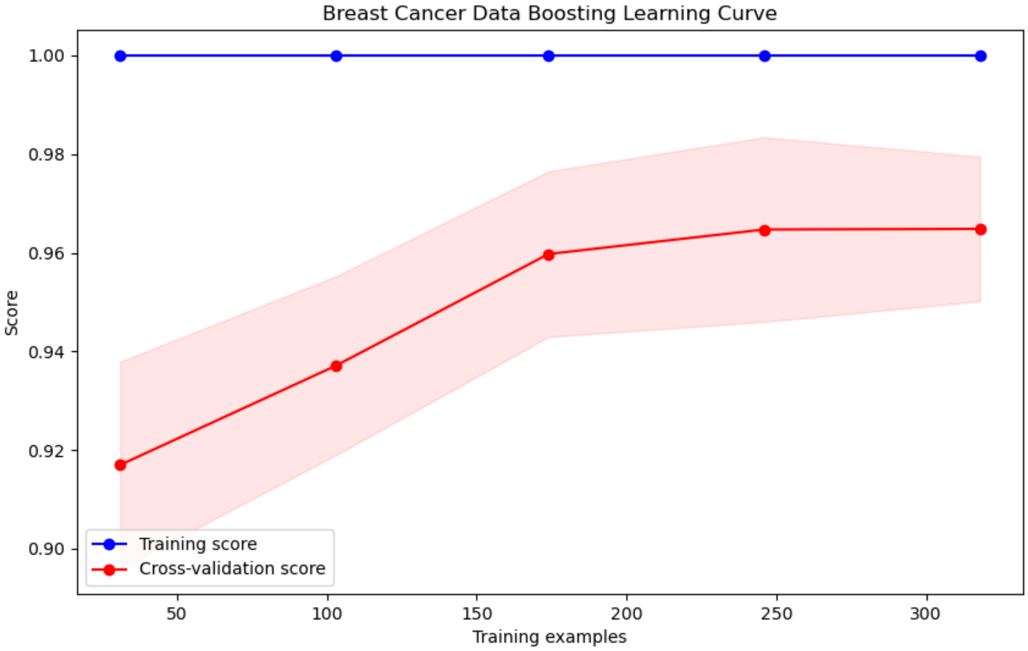

Figure 4—Learning curve for ADABoostClassifier

2.4.2 Tuning

The best combination of parameters found for ADABoostClassifier were:

{'learning_rate': 1.0, 'n_estimators': 1000}

The model with these parameters achieved a 98.246% accuracy on the test data.

2.5 SVM

The SVM model used for this analysis is scikit-learn’s SVC [7].

2.5.1 Base Model Results

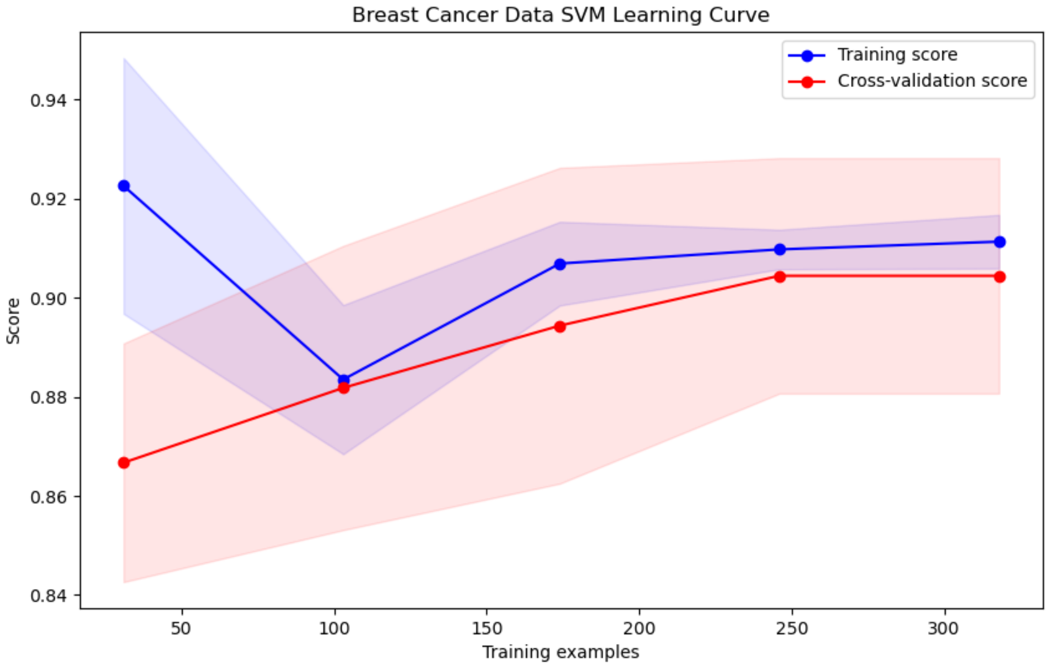

Figure 5—Learning curve for SVC

2.5.2 Tuning

The best combination of parameters found for SVC were:

{'C': 1000.0, 'gamma': 'scale', 'kernel': 'rbf'}

The model with these parameters achieved a 96.491% accuracy on the test data.

3 IRIS SPECIES DATASET ANALYSIS

3.1 Decision Tree

3.1.1 Base Model Results

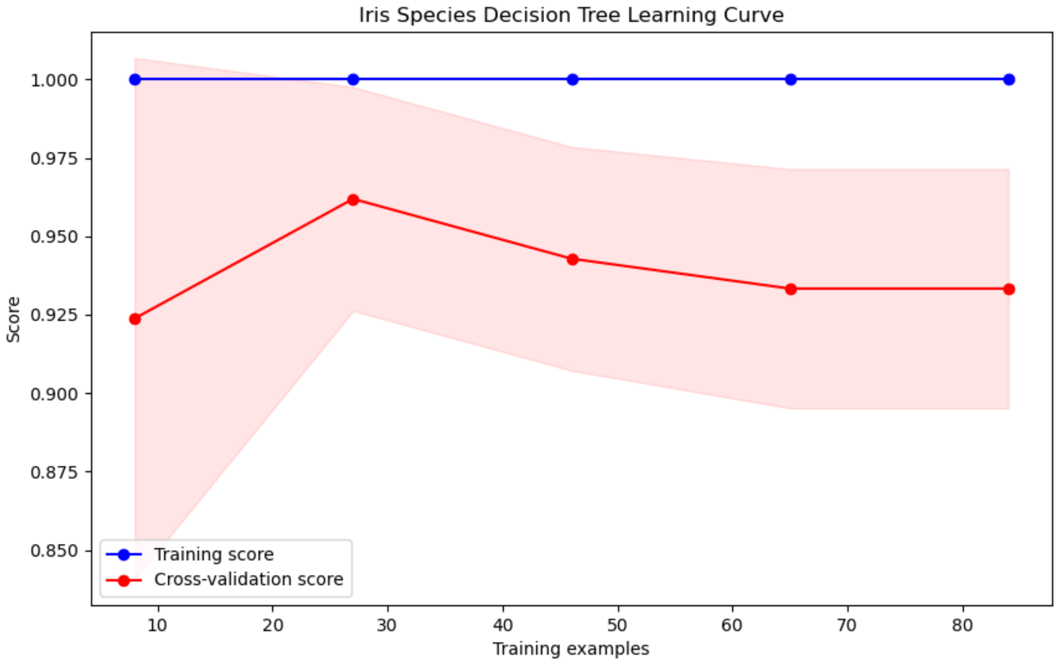

Figure 6—Learning curve for DecisionTreeClassifier

3.1.2 Tuning

The best combination of parameters found for DecisionTreeClassifier were:

{'criterion': 'gini', 'max_features': 'sqrt', 'min_samples_leaf': 5, 'min_samples_split': 4}

The model with these parameters achieved a 95.556% accuracy on the test data.

3.2 Neural Network

3.2.1 Base Model Results

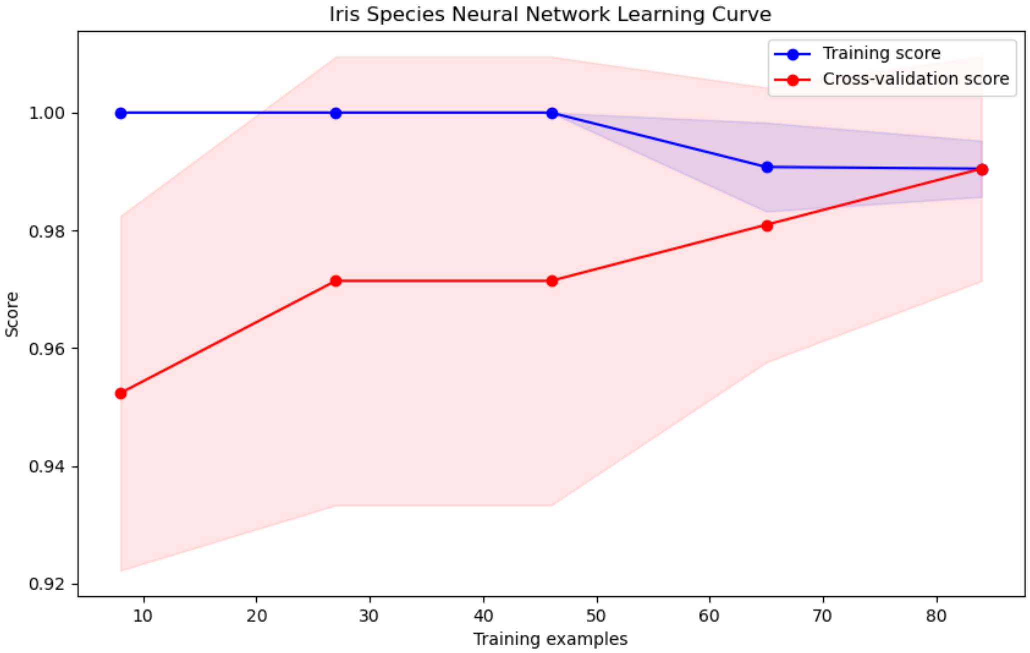

Figure 7—Learning curve for MLPClassifier

3.2.2 Tuning

The best combination of parameters found for MLPClassifier were:

{'alpha': 0.0001, 'hidden_layer_sizes': (50, 50), 'learning_rate': 'constant'}

The model with these parameters achieved a 93.333% accuracy on the test data.

3.3 k-Nearest Neighbors

3.3.1 Base Model Results

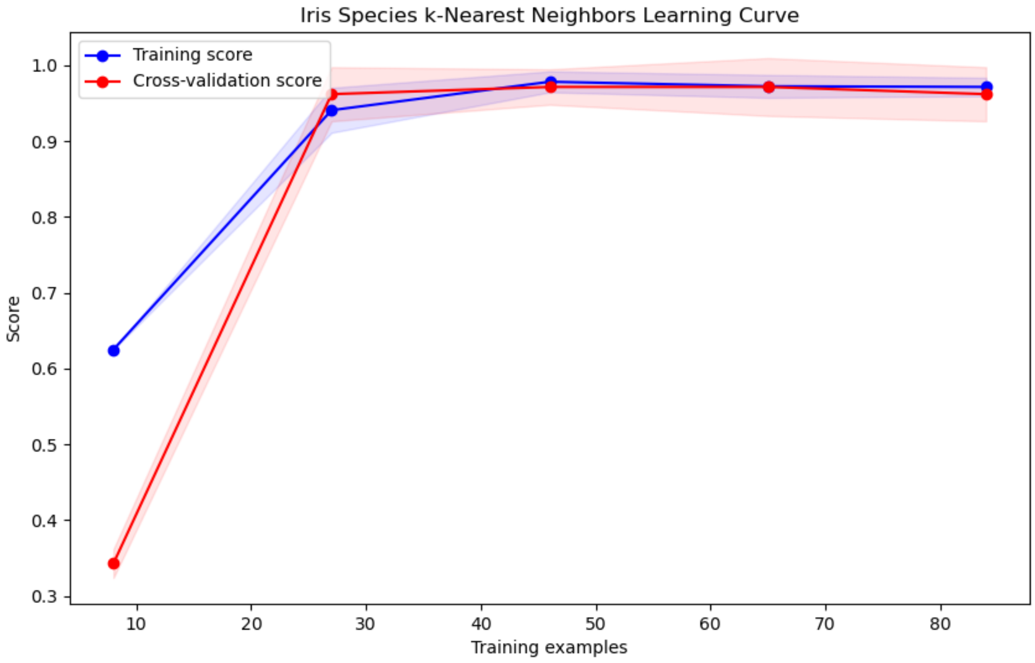

Figure 8—Learning curve for KNeighborsClassifier

3.3.2 Tuning

The best combination of parameters found for KNeighborsClassifier were:

{'n_neighbors': 8, 'p': 2, 'weights': 'distance'}

The model with these parameters achieved a 93.333% accuracy on the test data.

3.4 Boosting

3.4.1 Base Model Results

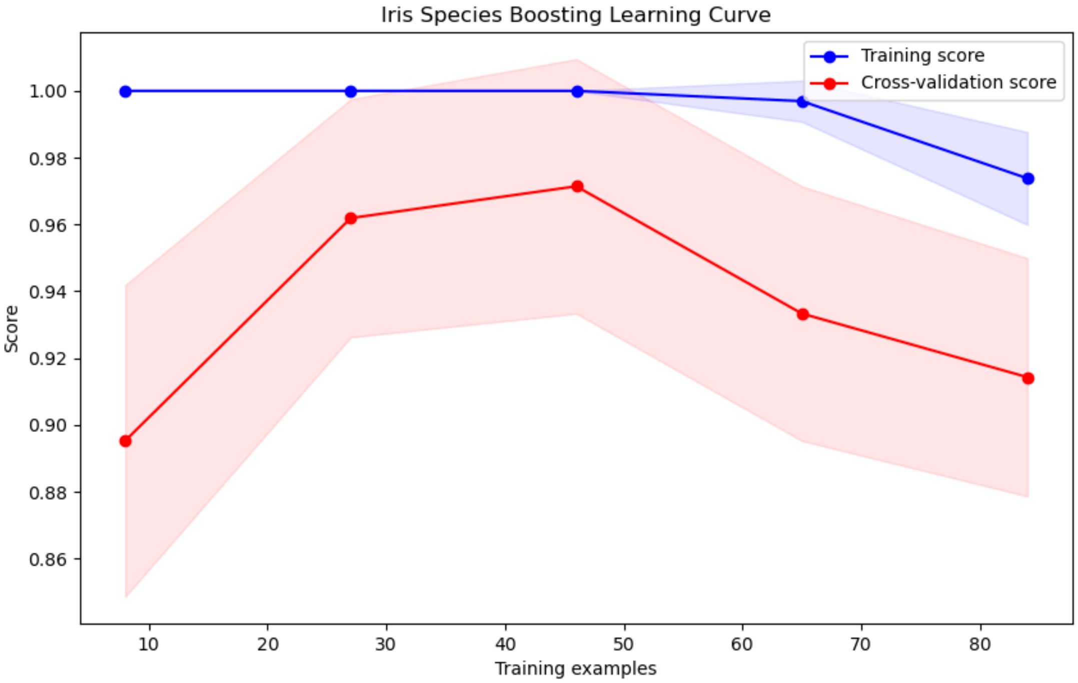

Figure 9—Learning curve for ADABoostClassifier

3.4.2 Tuning

The best combination of parameters found for ADABoostClassifier were:

{'learning_rate': 0.01, 'n_estimators': 10}

The model with these parameters achieved a 95.556% accuracy on the test data.

3.5 SVM

3.5.1 Base Model Results

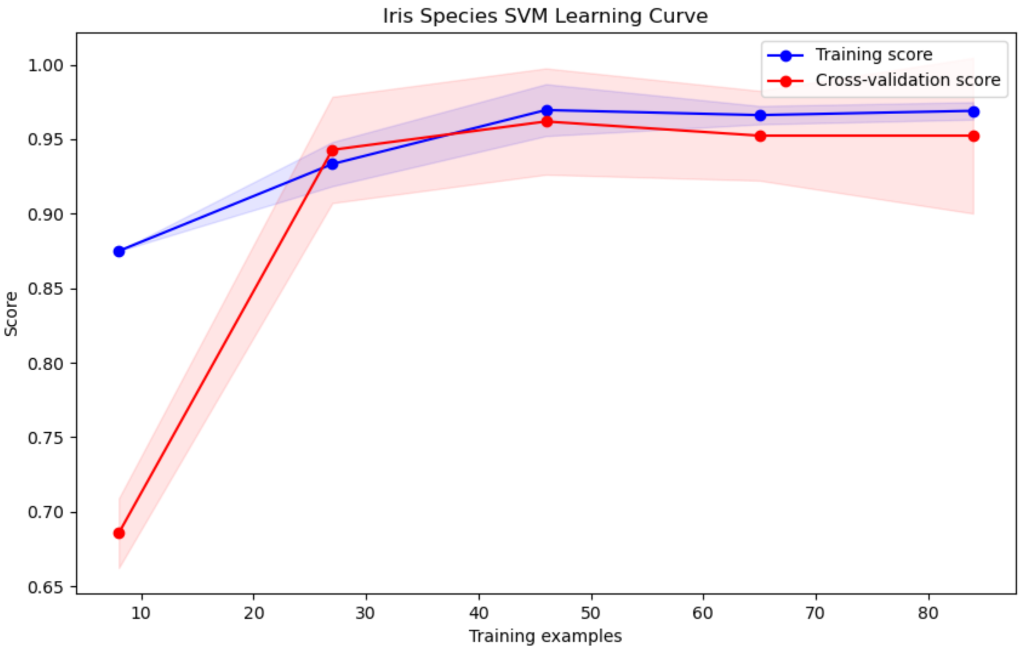

Figure 10—Learning curve for SVC

3.5.2 Tuning

The best combination of parameters found for SVC were:

{'C': 100.0, 'gamma': 'scale', 'kernel': 'rbf'}

The model with these parameters achieved a 95.556% accuracy on the test data.

4 REFERENCES

1. https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data
2. https://www.kaggle.com/datasets/uciml/iris
3. https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
4. https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html
5. https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
6. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html
7. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC