Nila Sadeeshkumar
CS 7641 - Machine Learning
September 26, 2021

Supervised Learning Project - Breast Cancer & Diabetes

Description of Dataset - Breast Cancer

This breast cancer dataset was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg. The dataset is used to predict whether a user has breast cancer or not. There are 699 instances in the dataset. Dataset contains 16 missing values and the class distribution of the set is of the following: Benign: 458 (65.5%), Malignant: 241 (34.5%) I removed the Sample Code Number from the training set since it's a unique identifier and serves no purpose in terms of classification.

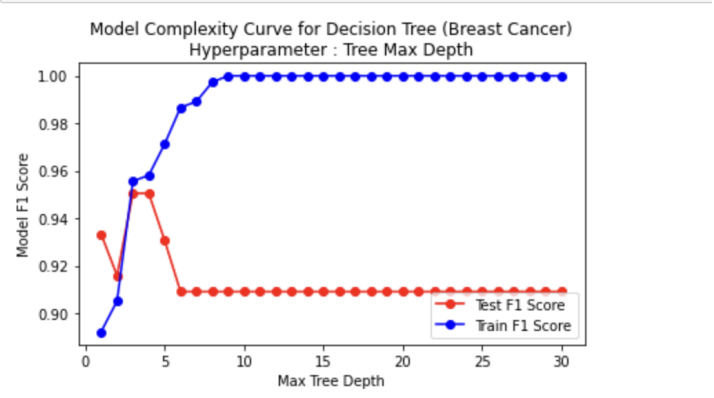
Description of Dataset - Diabetes

The Pima Indians Diabetes Dataset involves predicting the onset of diabetes within 5 years in Pima Indians given medical details. This is also a 2-class classification problem. The number of observations for each class is not balanced. There are 768 observations with 8 input variables and 1 output variable. Missing values are believed to be encoded with zero values.

Why is the Dataset Interesting?

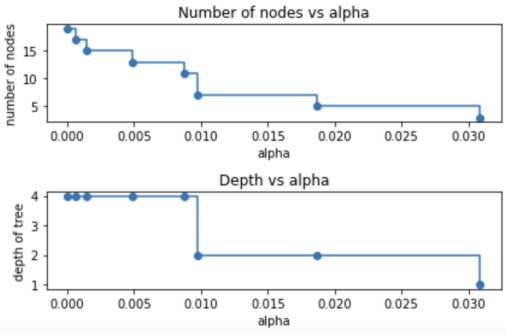
I believe the datasets are interesting for the application of use of these datasets and the results they yielded. Serious work and consideration is already being done on medical data to predict specific diseases. The breast cancer dataset predicts whether a user has cancer or not using a series of medical data. And, the Pima Indians Diabetes predicts whether the user will get diabetes within a period of 5 years. It will also be interesting to see the results of two different medical datasets and how different classification results affect them. The Diabetes dataset is unable to achieve the same accuracy rates of the Breast Cancer dataset. This could be a result of the features itself and the amount of information we are able to extract from every feature for every disease.

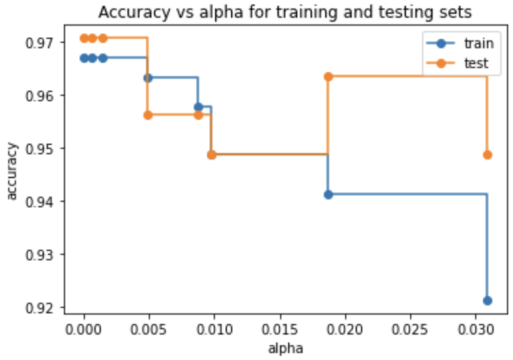
Decision Tree Classifier - Breast Cancer

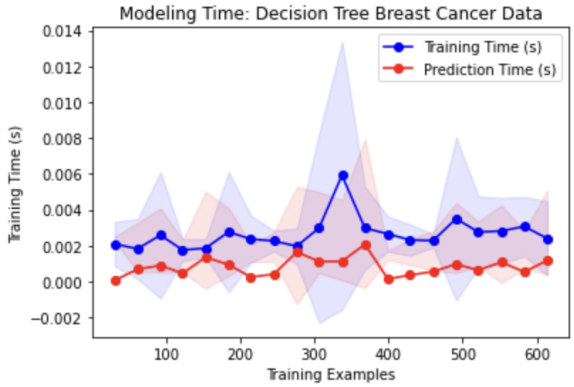


For the decision tree classifier, I first compared the max tree depth to the f1 score. As we can see the results on the left here, the ideal max tree depth is around 4-5.

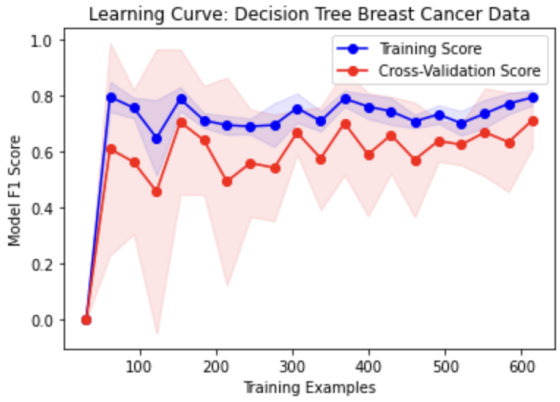
I ran a GridSearch to find the best hyperparameters. I decided on the criteria: gini, max_depth: 4, min_samples_leaf: 6.







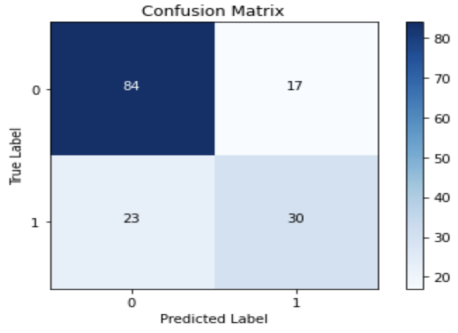
Then, I ran a pruning algorithm to find the best alpha. As we can see above in the diagram labeled Number of Nodes vs. Alpha & Depth vs. Alpha, as alpha increases the number of nodes and depth of tree decreases in order to prune the tree. From the diagram Accuracy vs. Alpha for training and testing sets, we can see that the highest accuracy for alpha is reached around 0.002.



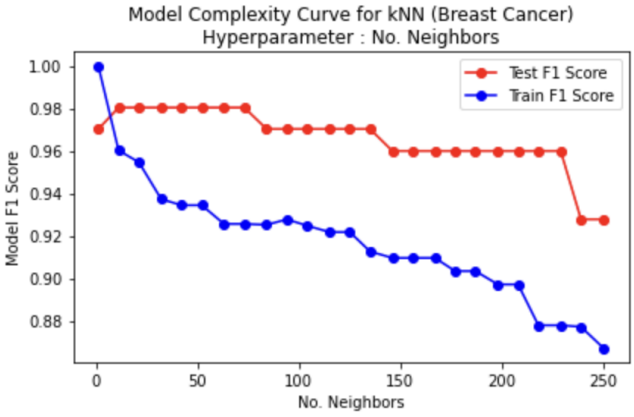
For the final classifier inputted with the best parameters, we can see how the f1 score and training time performs with x training examples.
From these results, we can see that the learning curve and modeling time curve shows that the data is underfit. This could be from the lack of samples. Overall, this model could also not be compatible for the complexity of the data provided. Although not shown here, I calculated the learning curve and modeling time without pruning the data and found it made little to no difference in underfitting the data.

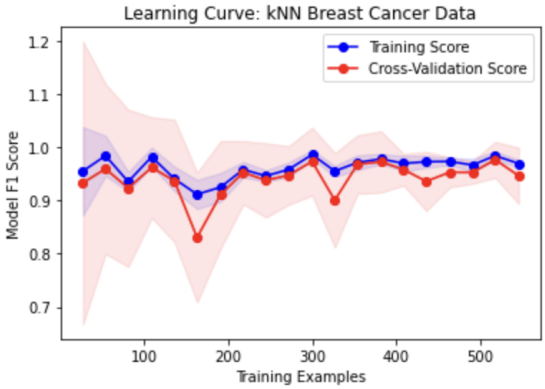
The confusion matrix to the right also demonstrates a lower accuracy and F1 score based on the data fit.

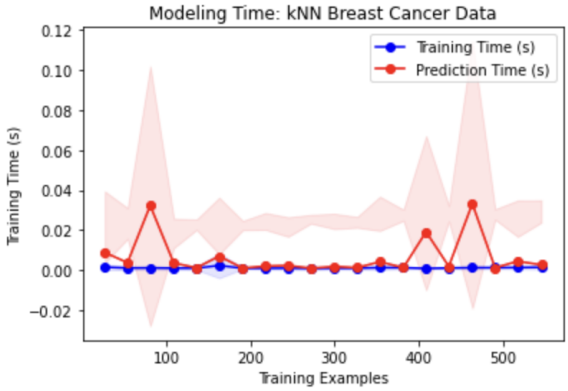
```
Model Evaluation Metrics Using Untouched Test Dataset
*******************************************************
Model Training Time (s):   0.00443
Model Prediction Time (s): 0.00012

F1 Score:  0.60
Accuracy:  0.74      AUC:       0.70
Precision: 0.64      Recall:    0.57
*******************************************************
```



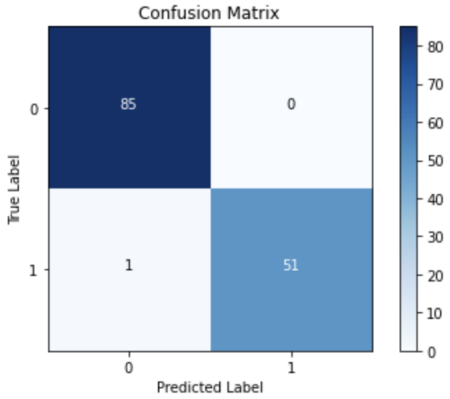
KNearestNeighbor - Breast Cancer





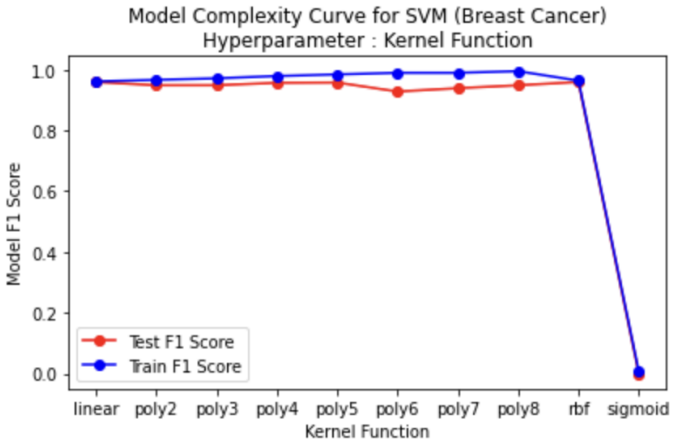



```
Model Evaluation Metrics Using Untouched Test Dataset
*****************************************************
Model Training Time (s):   0.00106
Model Prediction Time (s): 0.01444

F1 Score:  0.99
Accuracy:  0.99      AUC:       0.99
Precision: 1.00      Recall:    0.98
*****************************************************
```



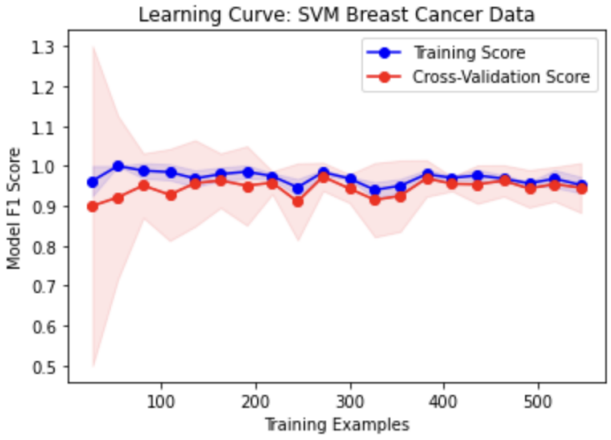
The KNearestModel best No. Neighbors is 5.
From the Model Complexity Curve, we can see that the best F1 score achieved on the Test set is 5 and then steadily decreases downwards. Based on the learning curve, we can see that the KNN Model fits well based on the cross validation score. This result is also demonstrated in the Accuracy score that's achieved of 99%.

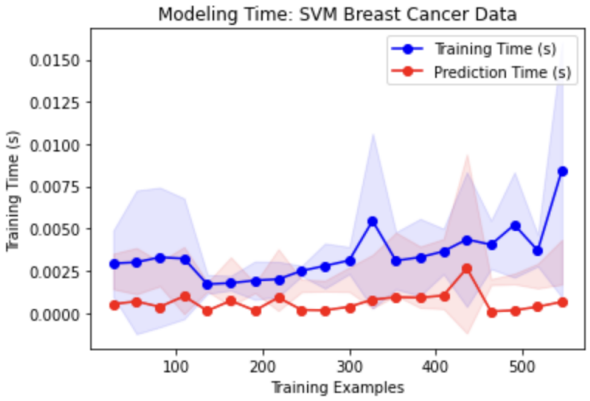
Support Vector Classifier - Breast Cancer



From the Model Complexity Curve for SVC, we can see that the best Kernel function is poly2 with an accuracy test score of 0.99.

After choosing poly2, we can run a GridSearch test to determine what the regularization parameter is. Based on these results, we get C value = 1.

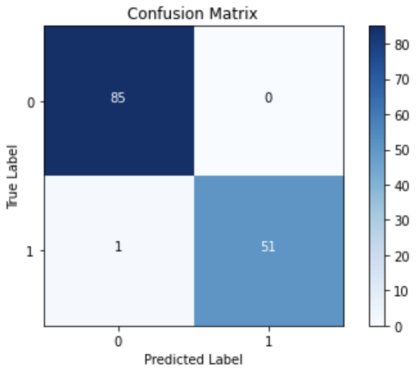




Based on the hyperparameters chosen, we get the learning curve that demonstrates a very good fit. The cross validation score fits the training score very well. The modeling time also demonstrates that prediction time is much lower than the training time.

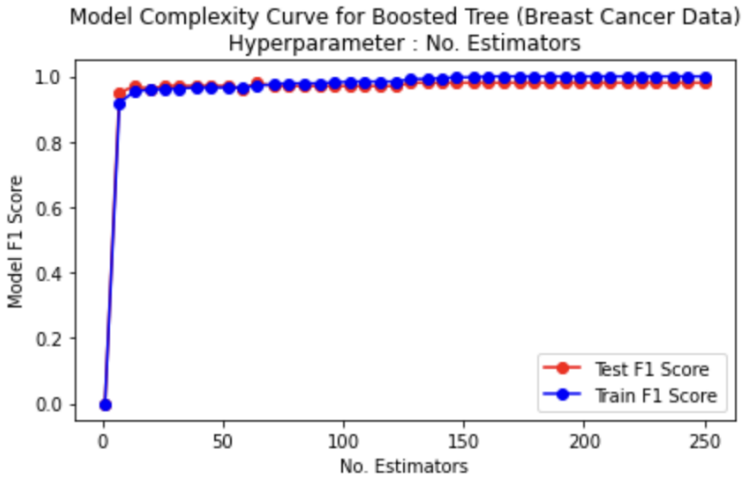
From the final evaluation metrics, we get a high f1 score and accuracy of 0.99.

```
Model Evaluation Metrics Using Untouched Test Dataset
*******************************************************
Model Training Time (s):   0.00562
Model Prediction Time (s): 0.00063

F1 Score:  0.99
Accuracy:  0.99      AUC:       0.99
Precision: 1.00      Recall:    0.98
*******************************************************
```



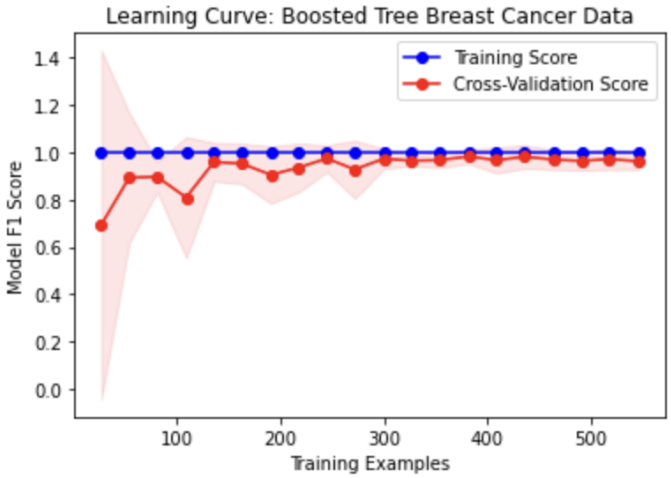
Boosted Decision Tree Classifier - Breast Cancer

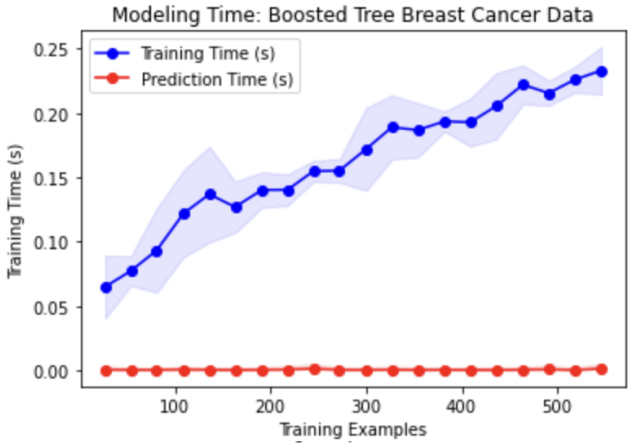
With this boosted decision tree classifier, I decided to implement a pre-pruning technique or an early stopping method instead of a post-pruning method with min_samples_leaf.



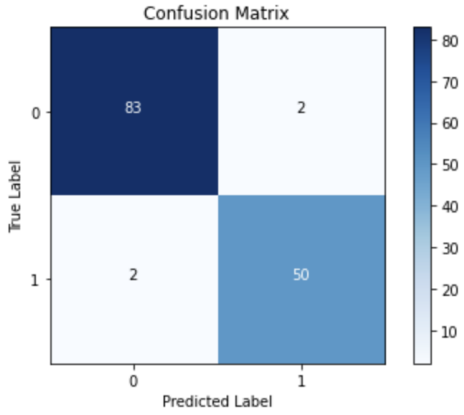
Based on this result, it's hard to read what the best number of estimators is. I performed a GridSearch for the Boosted Decision Tree Classifier and settled on the hyperparameters of loss: exponential, No. of Estimators: 100, max_depth = 6, min_samples_leaf: 3.

With the exponential loss function, we are utilizing the Adaboost algorithm.






```
Model Evaluation Metrics Using Untouched Test Dataset
*******************************************************
Model Training Time (s):   0.24614
Model Prediction Time (s): 0.00097

F1 Score:  0.96
Accuracy:  0.97      AUC:       0.97
Precision: 0.96      Recall:    0.96
*******************************************************
```



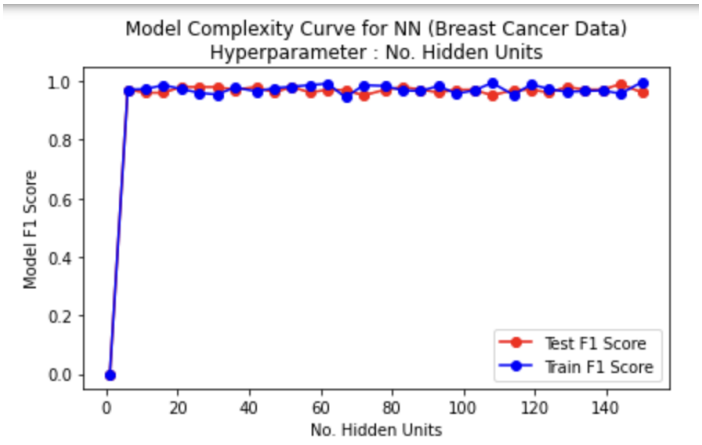
Taking a look at the learning curve, we can see based on the cross validation score that the model is fitting well. And, based on the modeling time, the prediction time is very low compared to the training time.

With the model evaluation metrics, we can see that the F1 score and accuracy of 97% is very high. Compared to the regular Decision Tree Classifier we have seen above, we can see the boosted decision tree classifier performs significantly better.

Neural Network - Breast Cancer

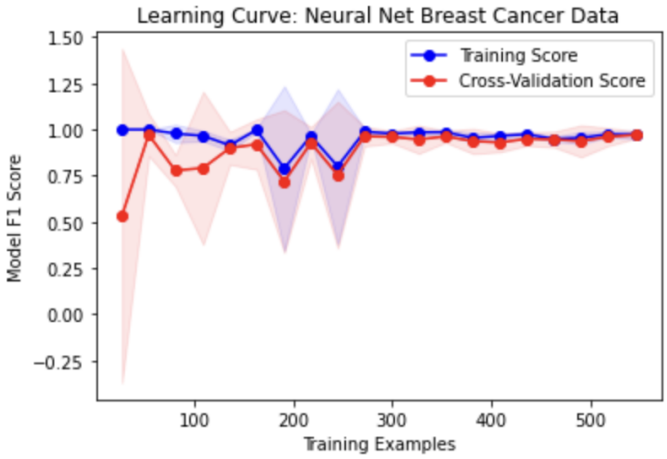
For the Neural Network Model, we are utilizing the activation function relu and the default solver: Adam. I initially thought of using the lbfgs solver because it is said to work better for smaller datasets. But, not only did the accuracy rate drop, the time spent training and predicting was a lot higher. Relu in general is faster than the sigmoid activation function because it's derivative is faster to compute. And, relu compared to adam generally served a slightly better accuracy rate and learning curve.

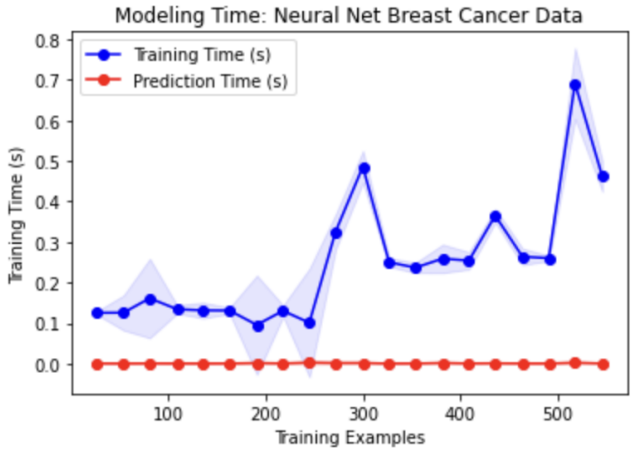
Then, I ran a GridSearch to search for the following best parameters: learning rate, hidden units, and alpha(regularization parameter).



Based on the Model Complexity Curve to the left, we can see once the number of hidden units reaches an optimal rate it doesn't vary much.

And, we prefer the lowest number of hidden units with the highest f1 score so the optimal number is: 10.



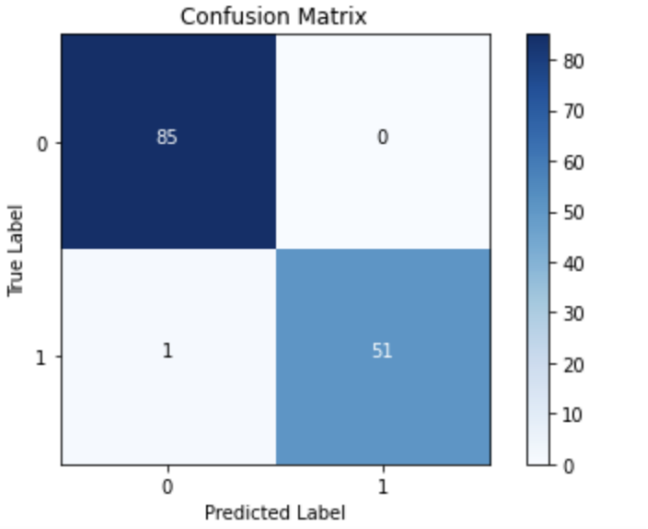



```
Model Evaluation Metrics Using Untouched Test Dataset
*****************************************************
Model Training Time (s):   0.47019
Model Prediction Time (s): 0.00036

F1 Score:  0.99
Accuracy:  0.99      AUC:       0.99
Precision: 1.00      Recall:    0.98
*****************************************************
```

The rest of the chosen hyperparameters are: learning rate: 0.01 and alpha: 0.01.

Based on the learning curve, we see that the Neural Network performs better with more training examples compared to less. And, we can see that in terms of modeling time, the higher the number of training examples the more training time occurs, although prediction time is negligible.

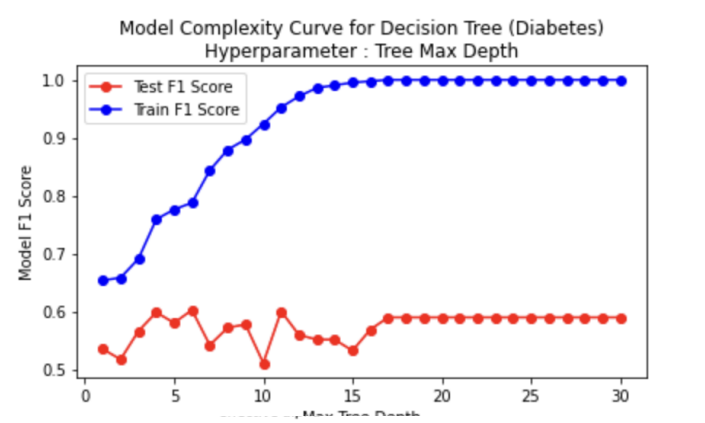
The Classifier overall performs very well with an accuracy and F1 score of 0.99.



Recommendation for Breast Cancer

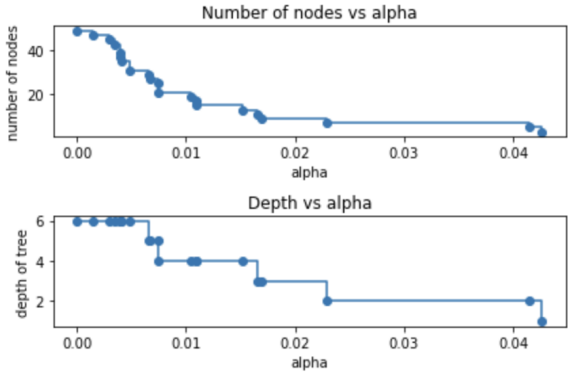
Given the details provided with these models, in order to choose the best model from the following: Decision Tree Classifier, KNN, SVC, Neural Network, Boosted Decision Tree Classifier, I will be evaluating the models on a number of characteristics. First, I chose to eliminate the Decision Tree Classifier because it did not achieve the same desired results as the rest of the models. Decision Tree Classifier had the lowest accuracy rate of 74%. Next, I chose to eliminate the Boosted Decision Tree Classifier and Neural Network. First, the training time required is significantly higher than the rest of the models. While Neural Network achieves the same results, the training time gets larger with more and more training examples. Second, Occam's Razor states that "the simplest solution is always the best". The computation time and complexity of the model is simply not needed when other models can achieve a similar level of accuracy. Support Vector Classifier and K-Nearest Neighbors serves negligible training and prediction times as well as very high accuracy. Typically, Support Vector Classifiers take care of outliers better than KNN. If training data is much larger than the number of features, then KNN is better than SVC. And, SVC outperforms KNN when there are a large number of features and smaller amount of training data. While we are currently working with a smaller dataset, the ratio of features to training data is lower. **So, my recommendation for the Breast Cancer dataset is KNN.**

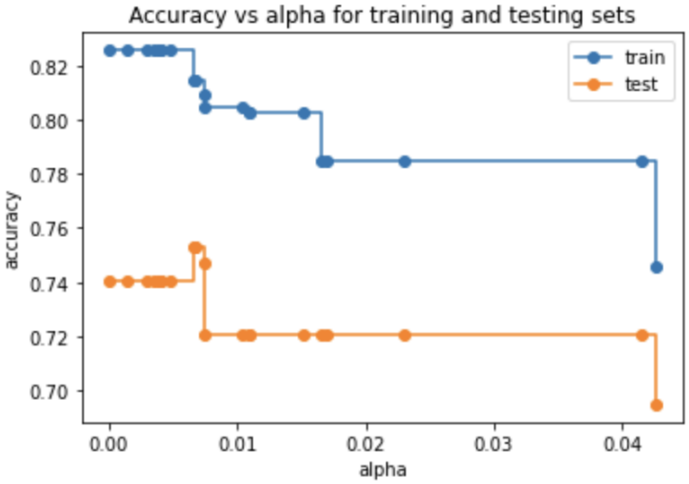
Decision Tree Classifier - Diabetes



The model complexity curve seen on the left demonstrates a smaller max tree depth achieves the optimal performance.

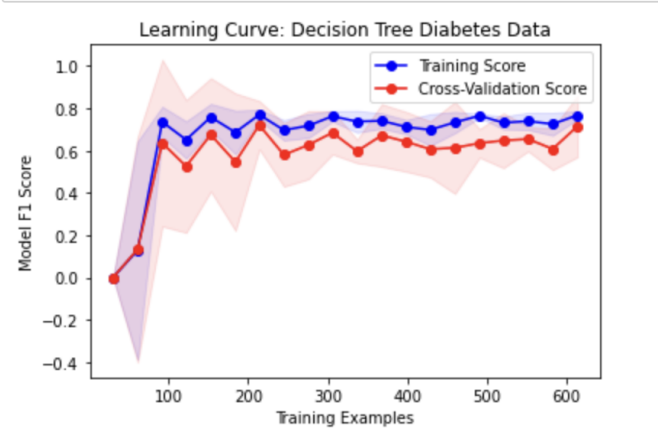
I ran a GridSearch to find the most suitable hyperparameters for this model which are the following: criterion: entropy, max depth: 7, min_samples_leaf: 17.

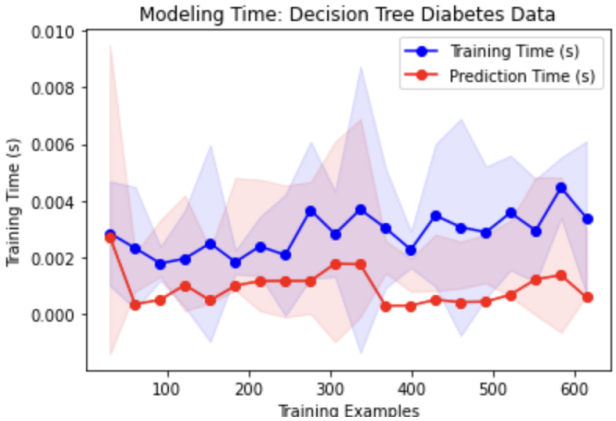




From the diagram above on the left, we can see that as alpha increases the depth of the tree and the number of nodes steadily decreases. From the graph “Accuracy vs. Alpha for training and testing sets”, we can see that the optimal alpha on the test set is just under 0.01.

I implemented this pruning on the DecisionTreeClassifier with the chosen hyperparameters chosen above.

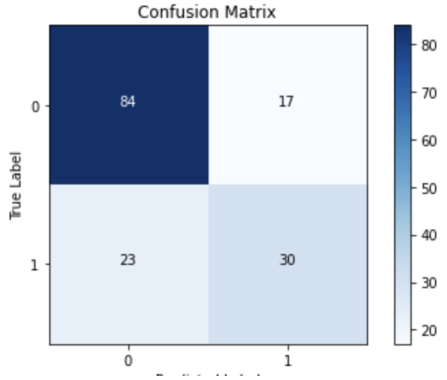




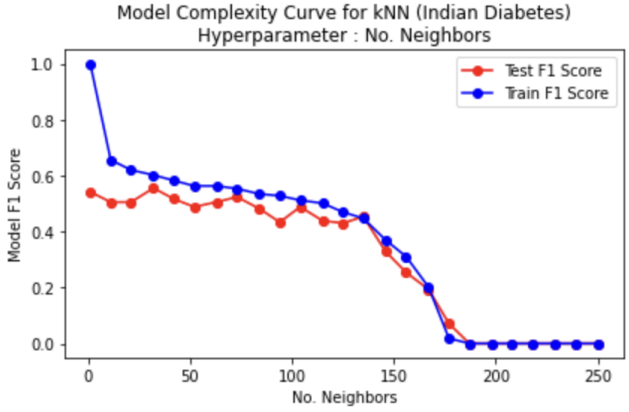

```
Model Evaluation Metrics Using Untouched Test Dataset
*******************************************************
Model Training Time (s):   0.00301
Model Prediction Time (s): 0.00009

F1 Score:  0.60
Accuracy:  0.74      AUC:       0.70
Precision: 0.64      Recall:    0.57
*******************************************************
```

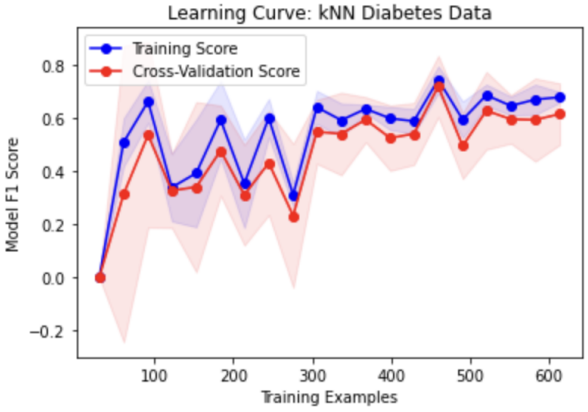
From the learning curve, we can see that there is a bit of underfitting but we can also see that with more training examples it might be enough for the right fit. From the Modeling Time Curve, we can see that the prediction time is not negligible and just under the prediction time.

And, with the model evaluation results we can see that the Decision Tree Classifier is performing with an accuracy of 0.74 but with an F1 score much lower at 0.60.

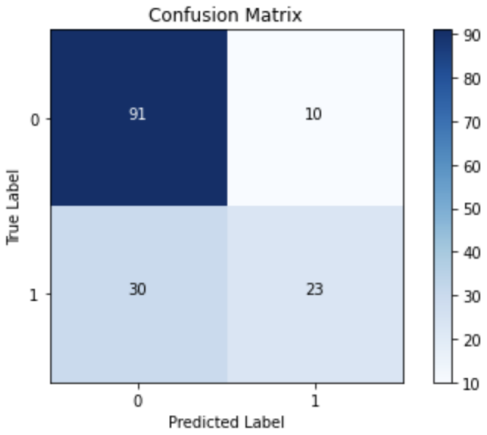


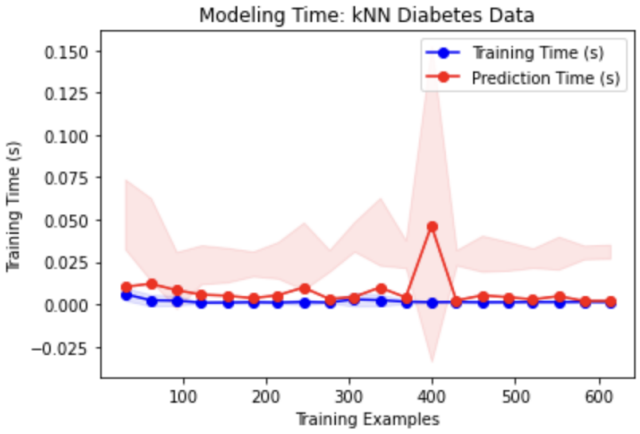
KNearestNeighbor - Diabetes Dataset





```
Model Evaluation Metrics Using Untouched Test Dataset
*****************************************************
Model Training Time (s):   0.00098
Model Prediction Time (s): 0.01551

F1 Score:  0.53
Accuracy:  0.74     AUC:       0.67
Precision: 0.70     Recall:    0.43
*****************************************************
```



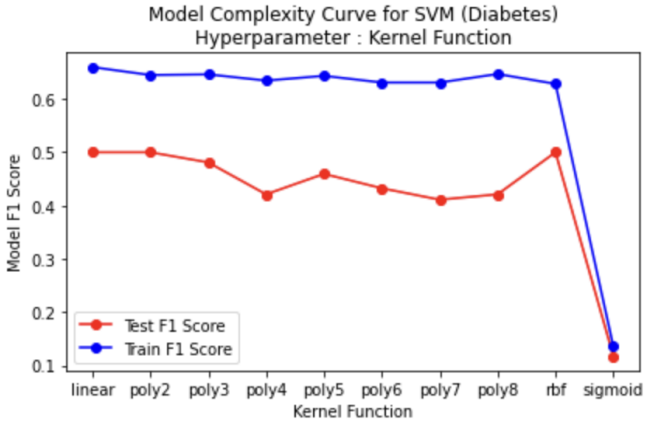


From the Model Complexity Curve, we can see that the best Number of Neighbors is 14. We also see this result after searching GridSearch for the best number of neighbors.

Based on the learning curve, we can see that the cross validation set performs well and is a good fit. And, based on the modeling time curve, we can see that the training time and prediction time is negligible although there is an unexplained spike in the prediction time around 400 training samples. This could be due to some type of an outlier or mislabeled data.

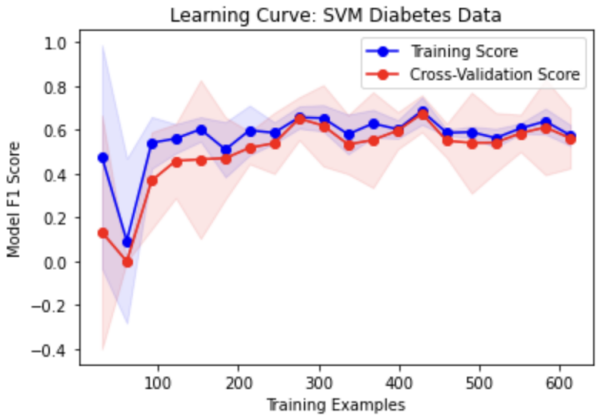
Overall, the KNN model achieves an accuracy around 74% but also an F1 score far below that at 53%.

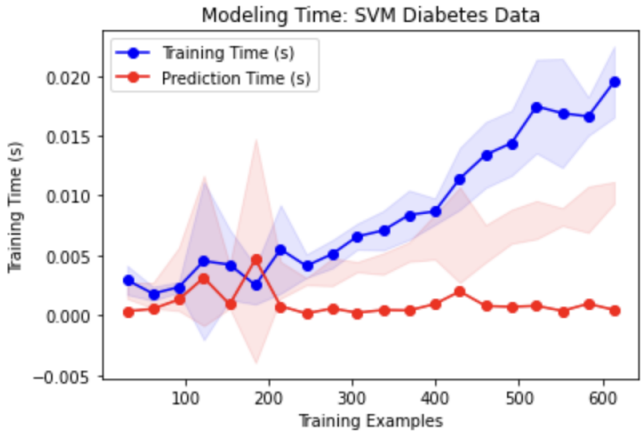
Support Vector Classifier - Diabetes



After taking a look at the Model Complexity Curve for the SVC Dataset, We can see that the test set performs best on the rbf kernel function.

Then, I performed a GridSearch to find the best hyperparameters for the regularization parameter, C which is: 1.



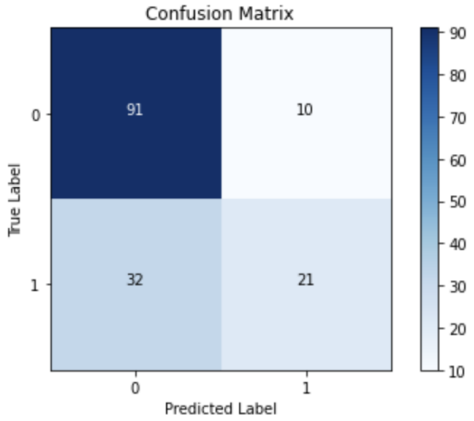


Based on the learning curve above, we can see that the model performs well once it reaches at least 200-300 training examples. There is a little bit more variation < 200 training examples.

Based on the Modeling Time curve, we can see that the training time linearly increases after the point of 250 training examples and takes up a significant amount of time compared to different models.

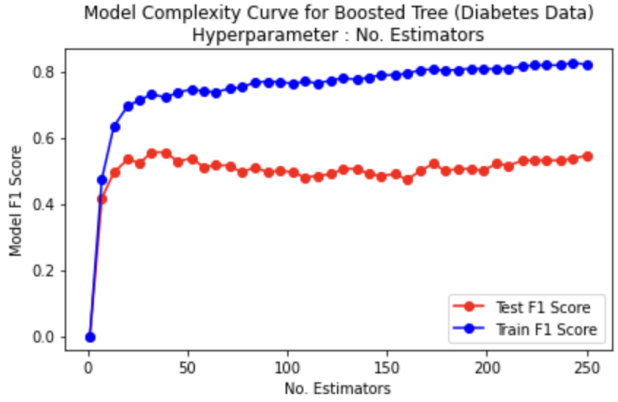
Overall, the model reaches an accuracy of 73% With an F1 score of 50% which is not ideal.

```
Model Evaluation Metrics Using Untouched Test Dataset
*******************************************************
Model Training Time (s):   0.03276
Model Prediction Time (s): 0.01320

F1 Score:  0.50
Accuracy:  0.73      AUC:       0.65
Precision: 0.68      Recall:    0.40
*******************************************************
```



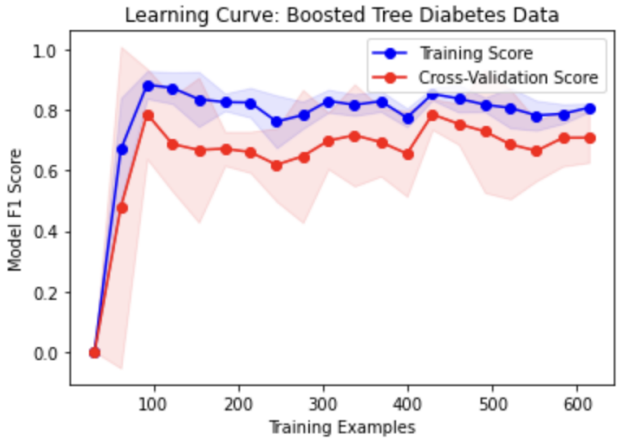
Boosted Decision Tree Classifier - Diabetes Data

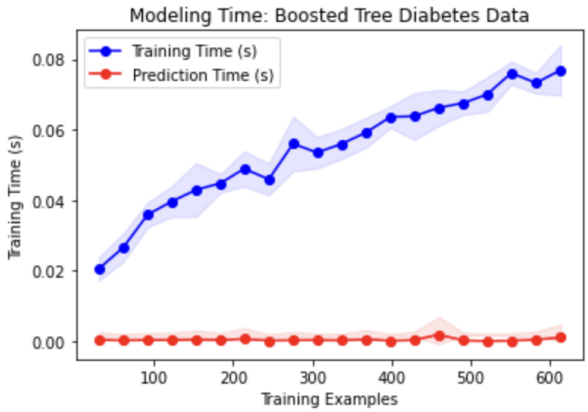
With this boosted decision tree classifier, I decided to implement a pre-pruning technique or an early stopping method instead of a post-pruning method with min_samples_leaf.



Based on the Model Complexity Curve on the left, we can see that we reach the ideal number of estimators < 50 and plateau.

Based on the Grid Search results, we get the hyperparameters of learning rate: 0.0505, max_depth: 3, min_samples_leaf: 17, n_estimators: 55.

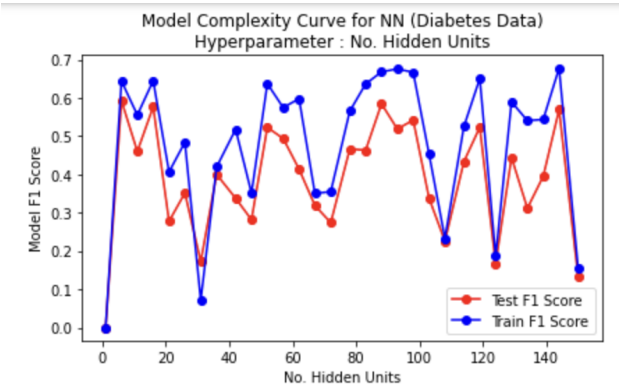


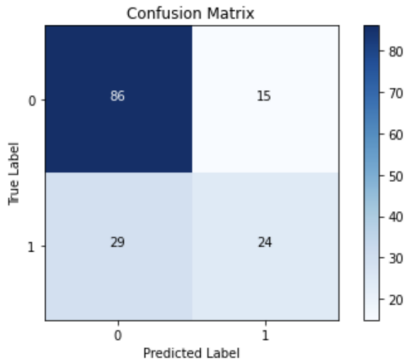


Based on the learning curve, we can see that data is a bit underfit. And, based on the modeling time curve, we can see that the prediction time is negligible and the training time increases linearly. Overall the model reaches an accuracy of 0.71 and F1 score of 0.52.

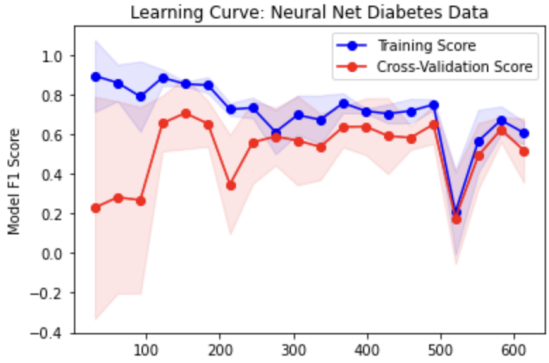
```
Model Evaluation Metrics Using Untouched Test Dataset
*******************************************************
Model Training Time (s):   0.08500
Model Prediction Time (s): 0.00064

F1 Score:  0.52
Accuracy:  0.71      AUC:       0.65
Precision: 0.62      Recall:    0.45
*******************************************************
```

Neural Network - Diabetes

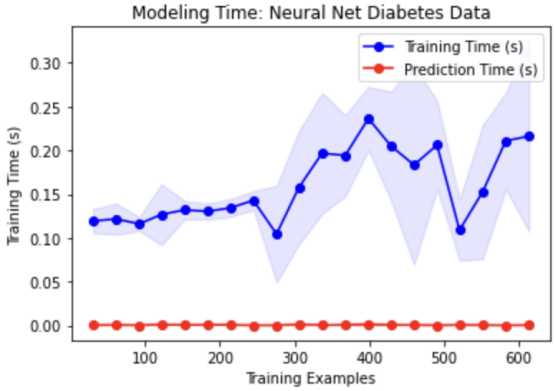




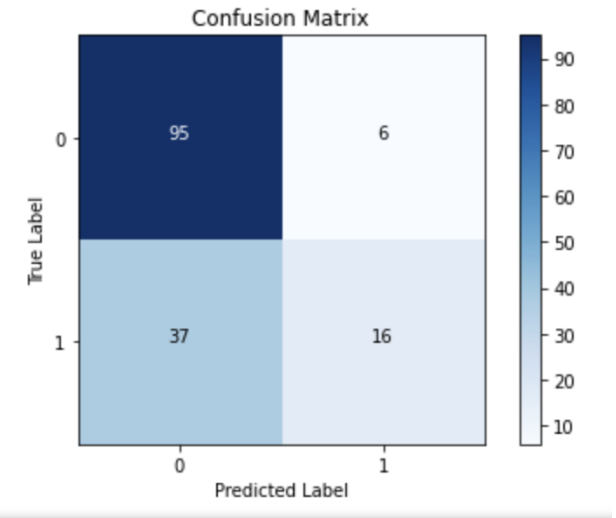
Based on the Model Complexity Curve, the best number of hidden units is 10. Also, I ran a Grid Search test to decide and tune the best hyperparameters which are the following: alpha: 0.01, hidden layer sizes: 10, learning rate: 0.01.



Based on the learning curve, the model seems to do better Once we reach around 500 training examples. Based on the Modeling Time curve, the training time generally increases over time but the prediction time is negligible.



```
Model Evaluation Metrics Using Untouched Test Dataset
*****************************************************
Model Training Time (s):   0.19473
Model Prediction Time (s): 0.00038

F1 Score:  0.43
Accuracy:  0.72     AUC:       0.62
Precision: 0.73     Recall:    0.30
*****************************************************
```



Based on the Model Evaluation Metrics, the Neural Network reaches an accuracy of 72% and an F1 score of 43% which is the lowest of all the models representing diabetes.

Recommendation for Diabetes Dataset

It's interesting to see how the Diabetes dataset performs comparatively to the Breast Cancer dataset. The Breast Cancer data set was able to achieve a far better accuracy across models than the Diabetes dataset.

In this dataset, the Decision Tree Classifier performs the best with an accuracy of 74% and an F1 score of 60%. The model is also very simple and explainable and according to Occam's Razor we want to choose the simplest model. The learning curve shows that the model accuracy increases with more training examples as compared to plateauing like some other models.

My recommendation is Decision Tree Classifier for the Diabetes dataset.