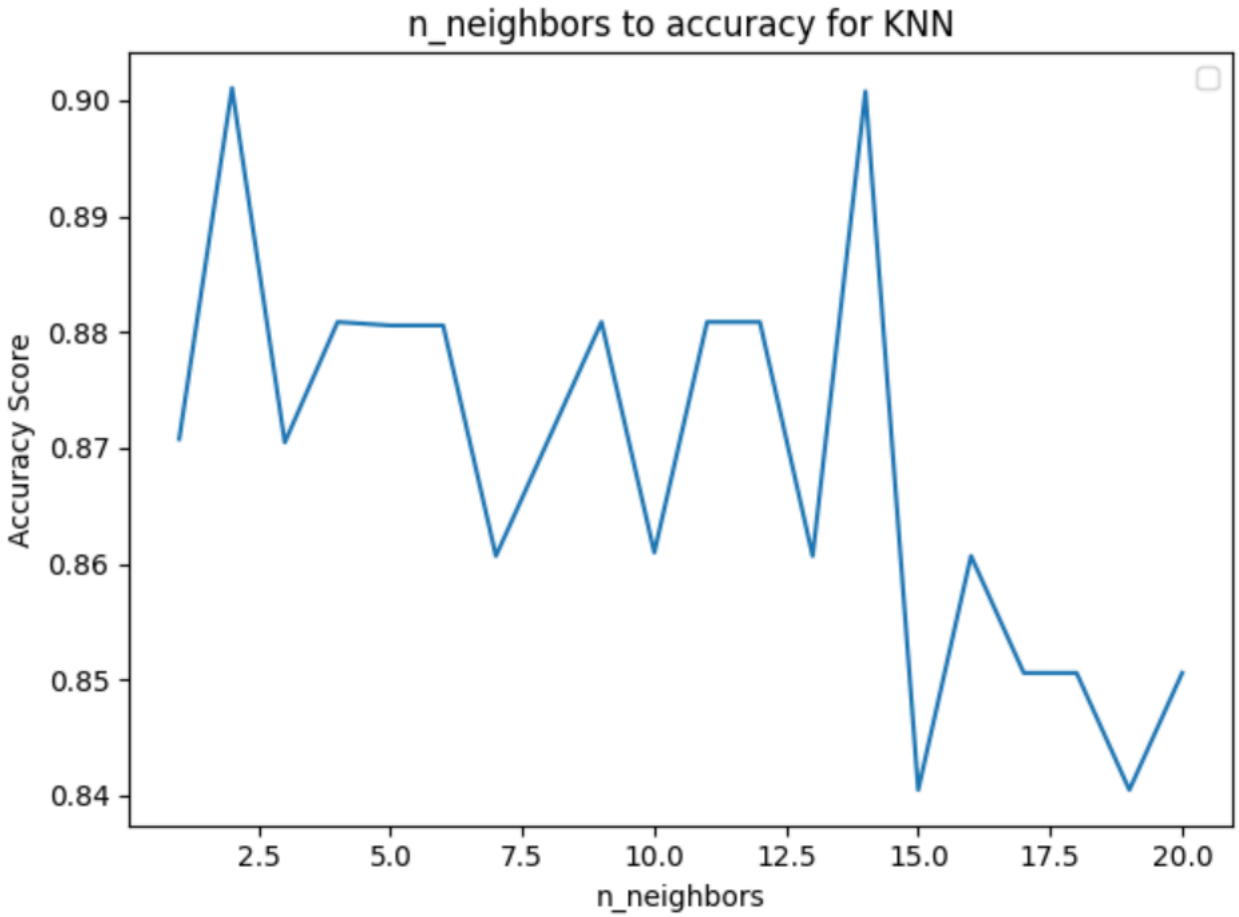
3.3 Short Answer: Parameter Tuning

Testing out different combinations of values for the n_neighbors and metric parameters, I found that the model with three neighbors and the Euclidian metric performed the best with an accuracy of 93.33% on the balanced training sets.

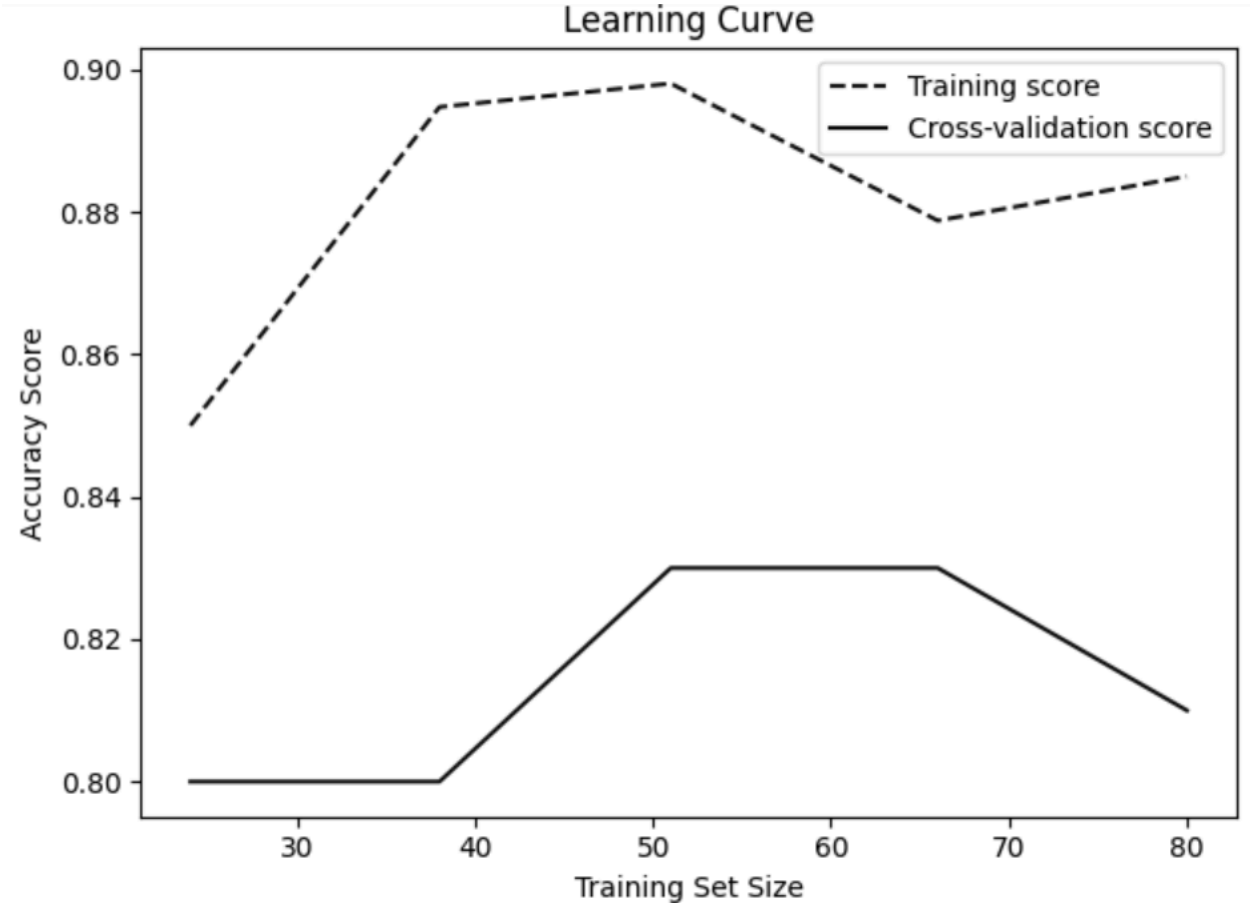
```
Model with 2 neighbors and cosine as the metric
Accuracy with balanced train and test splits:  0.8
Accuracy with biased train data:  0.6176470588235294

Model with 2 neighbors and euclidean as the metric
Accuracy with balanced train and test splits:  0.9
Accuracy with biased train data:  0.6323529411764706

Model with 3 neighbors and cosine as the metric
Accuracy with balanced train and test splits:  0.7666666666666667
Accuracy with biased train data:  0.5294117647058824

Model with 3 neighbors and euclidean as the metric
Accuracy with balanced train and test splits:  0.9333333333333333
Accuracy with biased train data:  0.6029411764705882

Model with 4 neighbors and cosine as the metric
Accuracy with balanced train and test splits:  0.6666666666666666
Accuracy with biased train data:  0.6764705882352942

Model with 4 neighbors and euclidean as the metric
Accuracy with balanced train and test splits:  0.8666666666666667
Accuracy with biased train data:  0.6470588235294118
```

3.5 Code: Plot (n_neighbors vs accuracy plot)



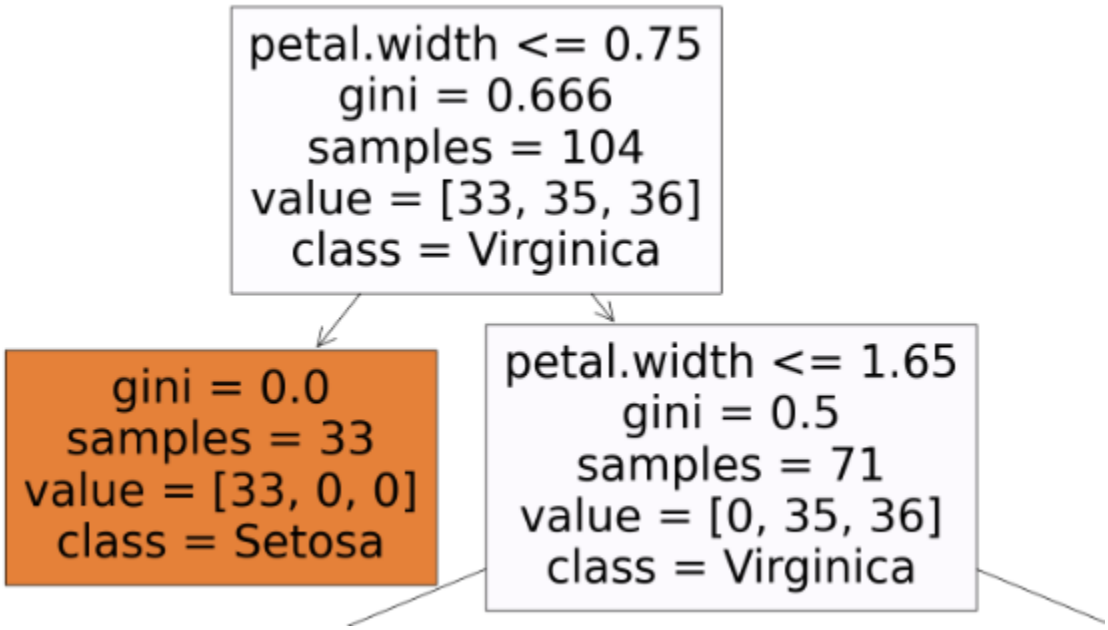
3.6 Short Answer: Learning Curve

Based on the graph, the size of the training set greatly influences the performance of the KNN model. As the training size increases from 30 to 50, there is a huge improvement in both the training and cross-validation scores as the model has more data points to learn from. However, once the training set size grew past 65, the model seemed to start overfitting to the training data since the training data kept increasing while the cross-validation score started to decrease. This implies that too much training data could be detrimental to the KNN model because there may be too many outliers in the training data, leading to the KNN model using these outliers as the nearest neighbors for some inputs and providing the incorrect output.



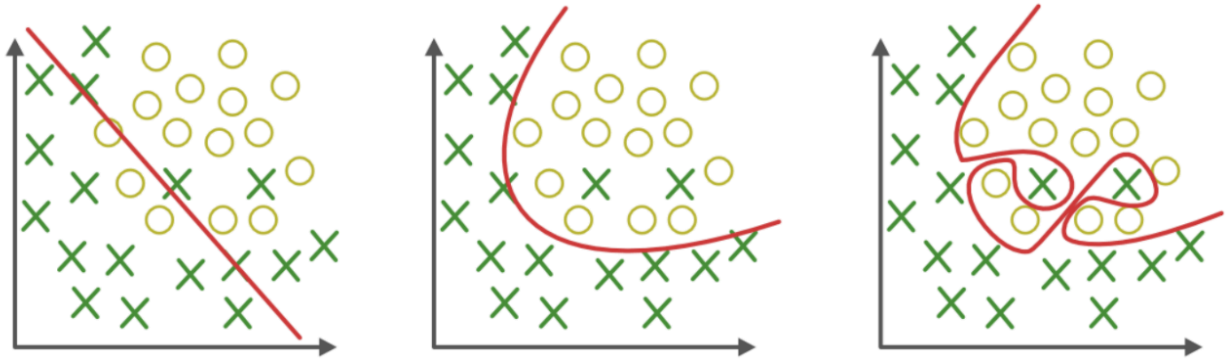
4.2 Short Answer: Decision Tree

The decision tree made this classification by following this path from the root to the leaf: the petal width is 0.2 which is less than the 0.75 threshold at the root. This decision tree then moves to a leaf node that classifies the flower as Setosa.



4.3 Short Answer: Overfitting

The classifier that is likely to overfit the dataset is the third classifier. This is because the classification boundary doesn't seem to follow any general equation (like just a line or a rotated parabola in the first two classifiers). The classification boundary for the third classifier appears to move very specifically to correctly classify two X data points surrounded by O data points, implying that the classifier has fit to correctly classify even these outliers. This makes me believe that the third classifier is likely to have overfit to the data.



4.4 Short Answer: SVM Regularization

C affects the classification boundary by adjusting the strength of the regularization penalty for the model and affects how tightly the model fits to the training data. Since the regularization penalty is inversely proportional to C, the higher values of C mean the model's classification boundary is very tight around the training data leading to very high training accuracy. The lower values of C make the classification boundary much more broad leading to lower training accuracy but a larger classification area which may result in better test performance over the models with a high value for C.