CS7641 Machine Learning Assignment #1 Report on Supervised Learning

Tichakunda Mangono

Introduction

This report seeks to apply several supervised learning algorithms to two separate datasets – one for **handwritten digit recognition** and another for **breast cancer diagnosis**. For each algorithm the report explores key algorithm learning steps, learning curve evolution, model complexity, and their effects on model performance. This analysis will lead to insights on the advantages and disadvantages of each algorithm for each of the two datasets while highlighting the different approaches across algorithms and potential points of improvements with a summary of concluding observations on supervised learning.

This analysis relies heavily on the Python libraries Scikitlearn, Keras and XGBoost for the machine learning algorithms while using Pandas, Numpy, Matplotlib and Seaborn for data manipulation and visualization.

Datasets

The two datasets selected for this analysis are the following classification problems: i) **handwritten digit recognition** (referred to as "*digits*" for the rest of this report) – a labelled dataset of 1797 (8x8 pixel) images of handwritten digits with 10 classes (from 0 to 9), and ii) **breast cancer diagnosis** (referred to as "*diagnosis*" for the rest of this report) – a labelled dataset of 569 instances with 30 features computed from a digitized image of a fine needle aspirate of a breast mass, describing the characteristics of cell nuclei present in the image.

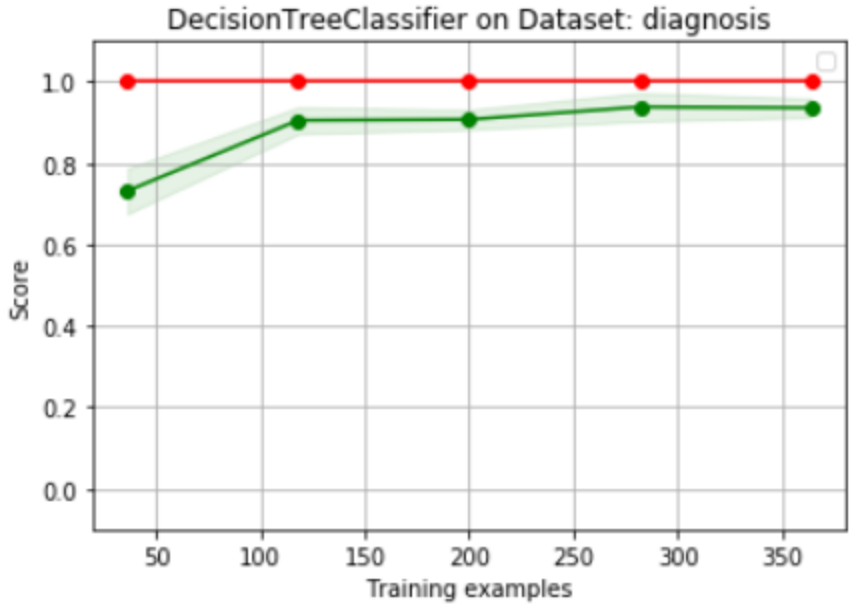
While the digits and diagnosis datasets are individually interesting for representing the major topics of optical character recognition and predictive disease diagnosis, they also represent two contrasting approaches of classifying data from images. The digits dataset represents raw, primary pixel-level information, while the diagnosis dataset is a set of carefully computed secondary features of the area of interest (cell nucleus) in an image. As a result, the feature-space of the two datasets represent two extremes - the digits data have a homogenous while diagnosis data has heterogenous feature-space. As such, we should observe interesting similarities and contrasts in the performance of different supervised learning algorithms on these datasets.

For analysis, the datasets are split into train, validation and test sets comprising 60%, 20%, and 20% of the full dataset respectively. The test set is extracted first, and then the algorithms learn on the train set and are cross-validated on the valid set several times before finally analyzing performance on the test set as a proxy for real-world, generalizable performance. Both datasets are available as native datasets within the Scikit Learn library.

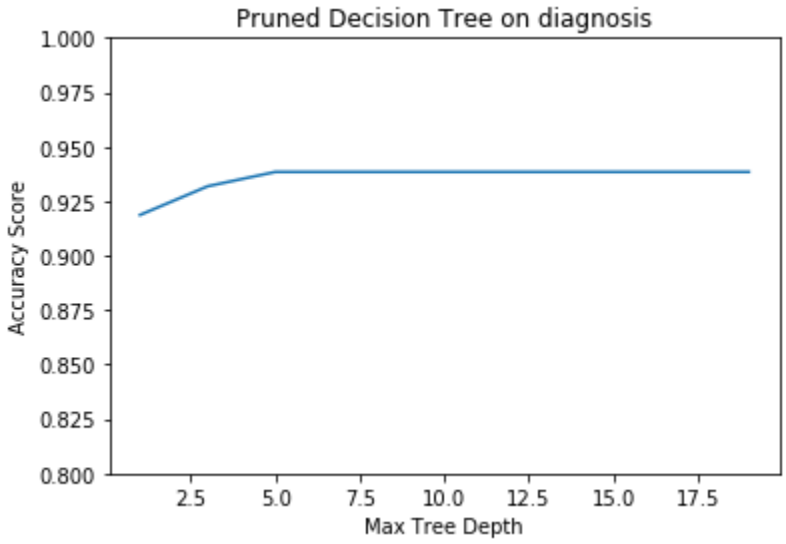
Decision Trees

Implementing a decision tree using the default "GINI" information gain criterion for splitting the features of the tree was selected as the criterion as it is fitting for a greedy algorithm such as Decision Trees. The information criterion decides on whether a feature (at is best split point – in a greedy fashion) will add the most amount of information to significantly decrease the variation in the data. This process is repeated until a stated level of tree depth or number of leaves etc. has been reached or there are no more instances left.

The charts below of cross-validated accuracy (red for training set and green for validation set) shows a high accuracy on the training set and a validation accuracy that starts low (just under 60% for digits, and just over 70% for diagnosis dataset). As the number for training set increases, the validation set accuracy increases steadily for the digits data and more quickly for the diagnosis data. Therefore, for both models, the model data you add, the better your accuracy and performance.



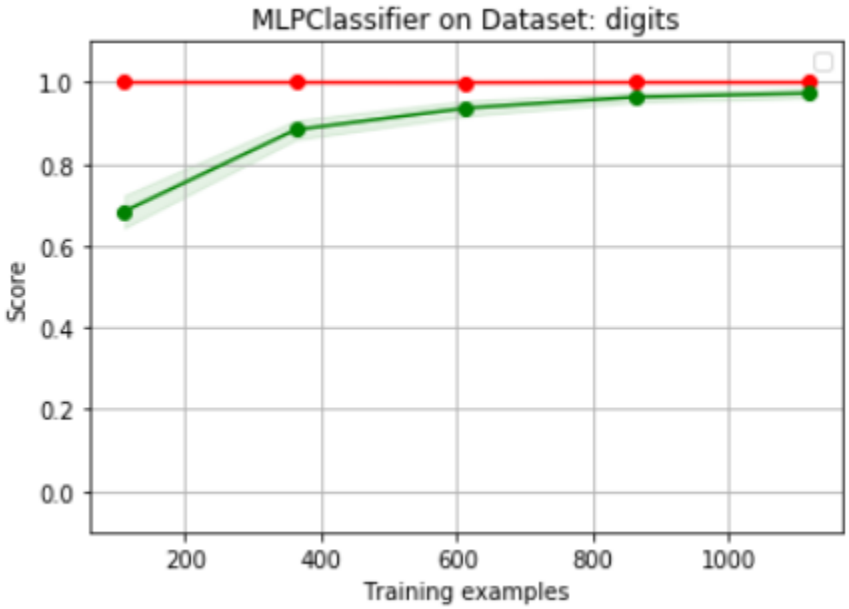
Pruning the decision tree models will allow the models to generalize better and perform more accurately in the real world. By decreasing the depth of the decision tree used to train and validate the model, the result is reduced overfitting and hence better generalization of the model across both the diagnosis and the digits dataset.



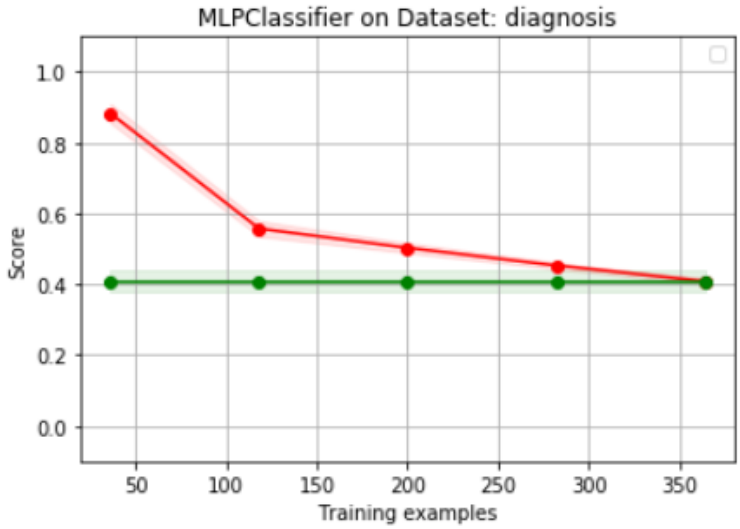
Neural Networks

Using Scikit Learn Multi-Layer perceptron algorithm and a neural network shows a trend like the Decision Tree algorithm on the digits dataset where the validation accuracy increases with the number of training examples. A high training set accuracy indicates overfitting which would

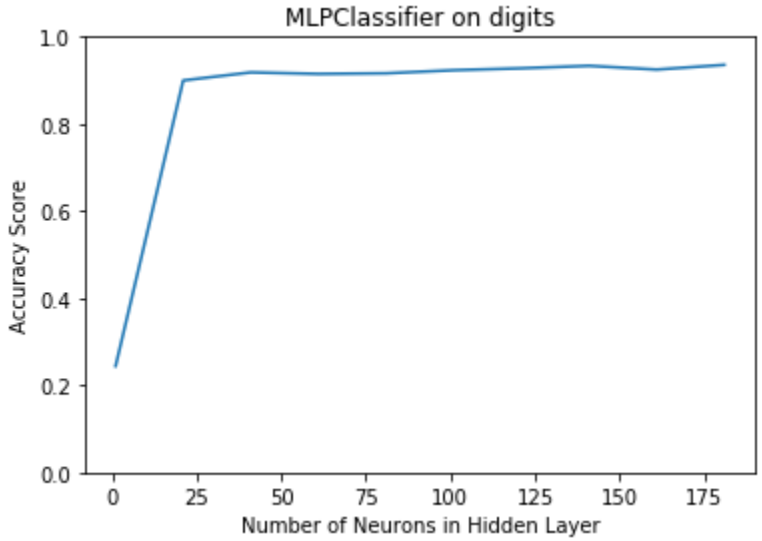
need to be reduced by using techniques such as increase the number of layers and employing techniques such as random “dropout”.



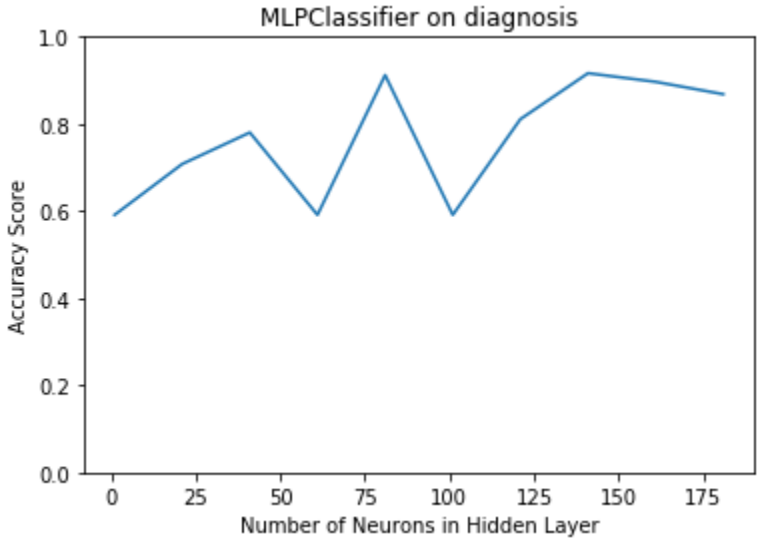
However, for the diagnosis dataset, the validation accuracy stays the same regardless of the number of training samples while the training set accuracy starts high at ~90% and decreases rapidly to about 40% to equal the validation set accuracy. Since the diagnosis training score decreases, this shows a model with significant underfitting and a high bias. At the same time, the cross-validated score does not change which means that the model as it is cannot extract any more meaningful information from the data. Because the score is very low at 40%, an increase in model complexity should give the model the flexibility it needs to begin learning from the data.



Increasing model complexity by increasing the number of neurons in the hidden layer for the digits dataset shows a monotonically increasing accuracy which settles around 90% with about 25 nodes in the hidden layer.

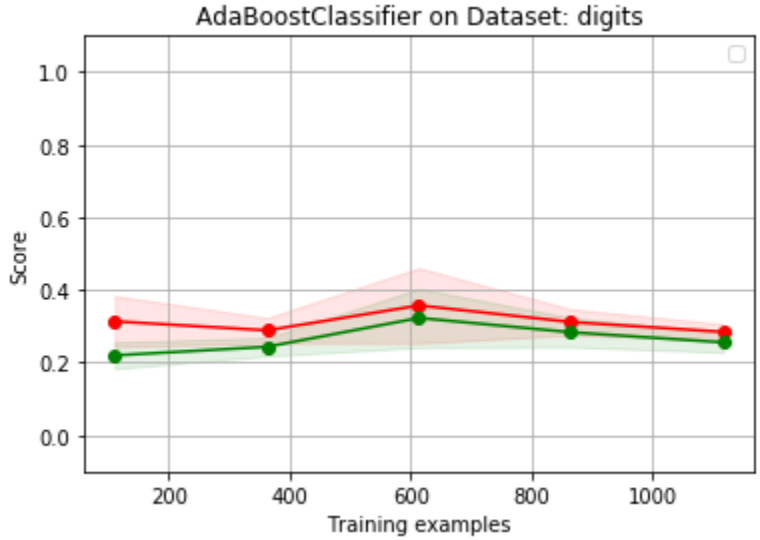


For the diagnosis data, increasing the model complexity by increasing number of neurons has mixed results which fluctuate. This indicates that in addition to more neurons, the model may respond to additional hidden layers, resulting in a deeper network which better able to generalize because it will not depend on any one layer for prediction. Currently, the model has high variance as seen in the chart below

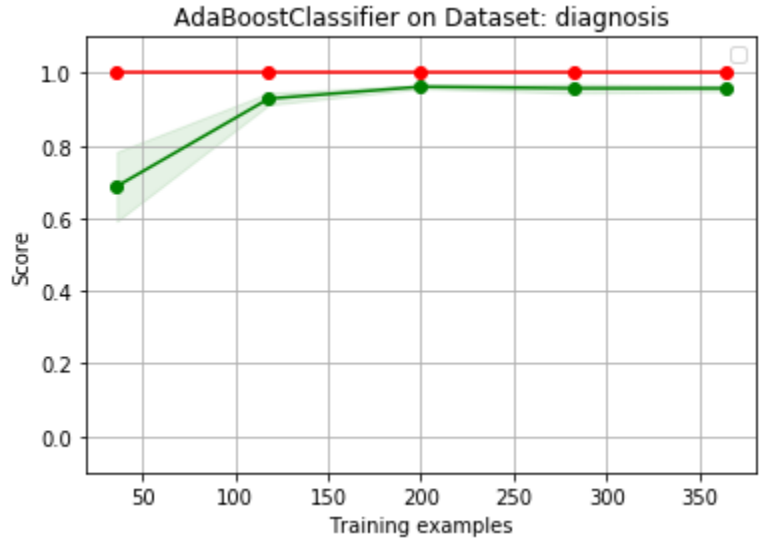


Boosting

Before pruning, the Adaptive Boosting algorithm of SciKit Learn has closely related, low values of accuracy for both its training and cross-validated data for the digits dataset. Adding training samples does not seem to affect any of the scores, which suggests underfitting of the model. So it would benefit for an increase in complexity when classifying digits, which is a multi-class problem.

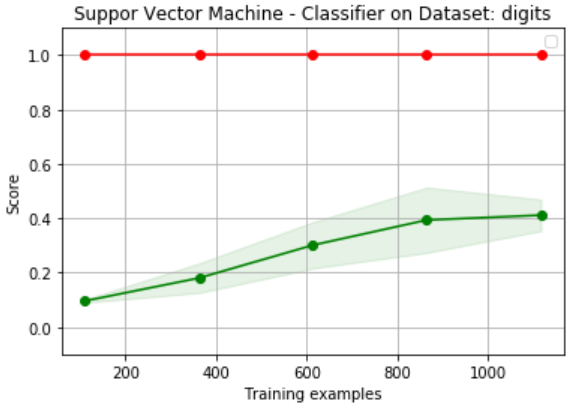


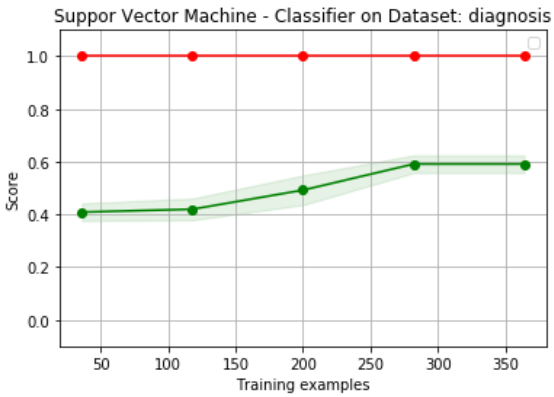
However, for the diagnosis dataset. The more data you add, the more accuracy you get on the cross-validated dataset. The training accuracy is high at 100% which suggests potential overfitting during to high variance. Thus pruning the tree as in the Decision Tree example above will help the model generalize on the test set or real world data.



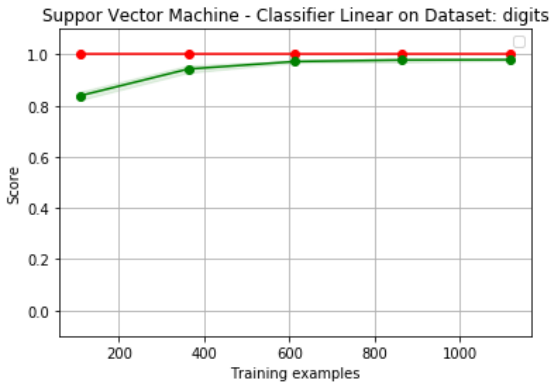
Support Vector Machines

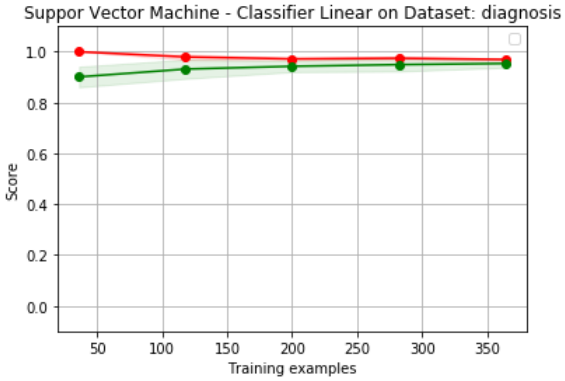
Support Vector Machines with a radial basis function as the kernel have a high training accuracy and low validated accuracy which suggests the model is overfitting. The validation accuracy starts very low and increases as the number of data points increases, but it reaches a plateau for both data sets, meaning that no changes in the data sample size will be helpful beyond any more. In general, the RBF kernel performs worse on the digits (multiclass) data than on the diagnosis (binary) data, despite similar patterns of overfitting. This requires a change in the model or decrease in complexity such as using a linear kernel instead.





Using a linear kernel for Support Vector Machines shows that the models behaves better with validation accuracy increases towards training accuracy as we increase the sample size. This is a better-behaved model to use as it will generalize better in the real world. This also suggests that the datasets classes are close to linear separability.

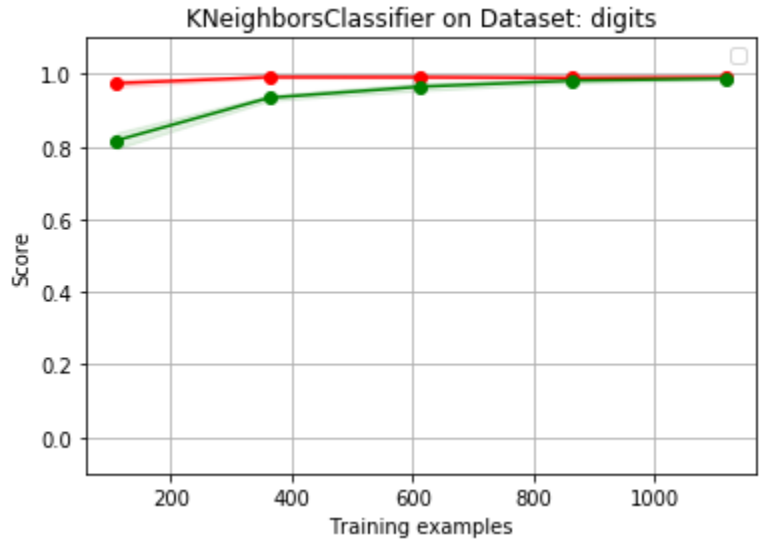


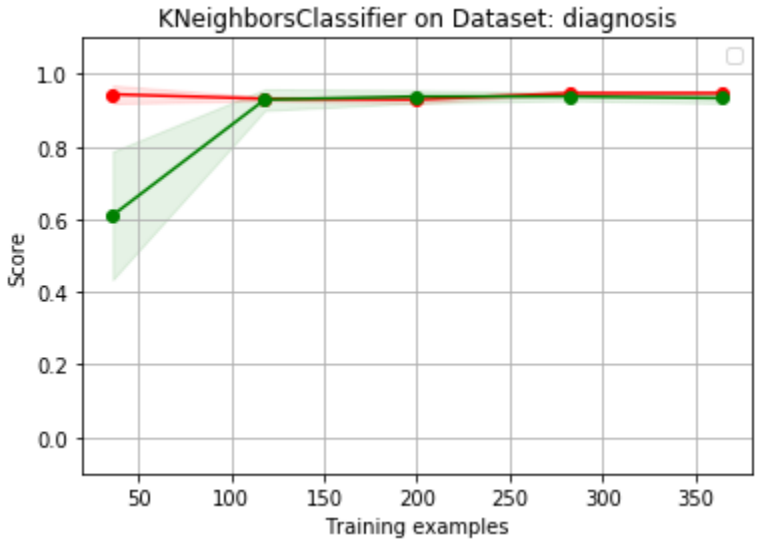


k-Nearest Neighbors

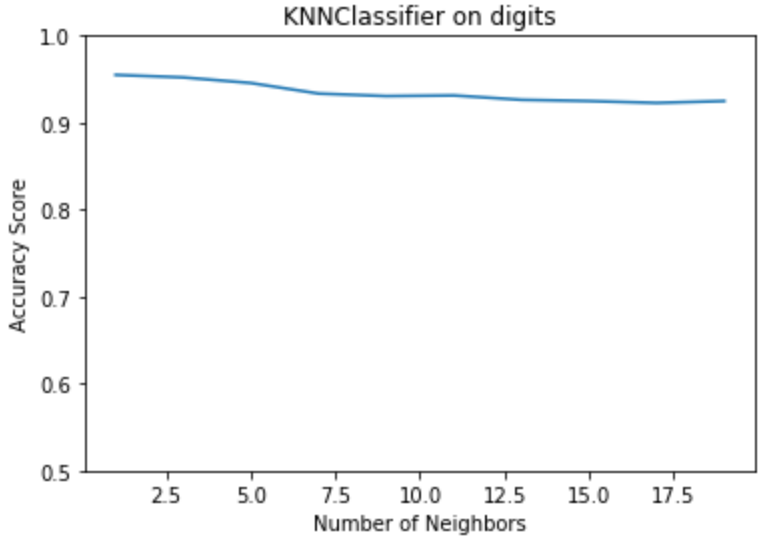
The KNN model from Scikit learn classifies points based on the majority class among neighboring instances for a specified k number of neighbors. The proximity of neighbors is determined by various distance metrics of choice, in this case Euclidean distance is used. The KNN model is appropriate and performs well on both data sets without evidence of significant overfitting except when the number of samples is very low i.e. below 100. Otherwise, the

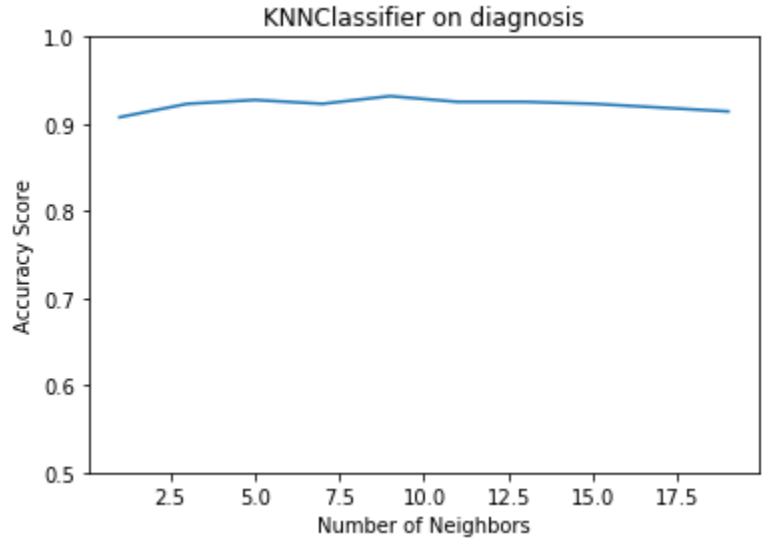
accuracy for validation increases steadily and levels out for the validation data set. This model uses 3 as the default number of neighbors.





To change model complexity, change the number of neighbors. The more neighbors, the less the model will overfit because more data points are reliable voters in the system. However, a small number of neighbors will lead to overfitting, and especially if K=1. So the best number of neighbors are somewhere in the middle





Conclusion

There is a need to look at both training and validation scores to determine the levels of over and underfitting due to variance and bias trade-offs in the model assumptions. Once learning curve is established, then more data can be obtained or a change in model complexity will reduce or increase the variance.