CS4641: Assignment1

Name: Heejoo Cho

Date: Sep 24th, 2018

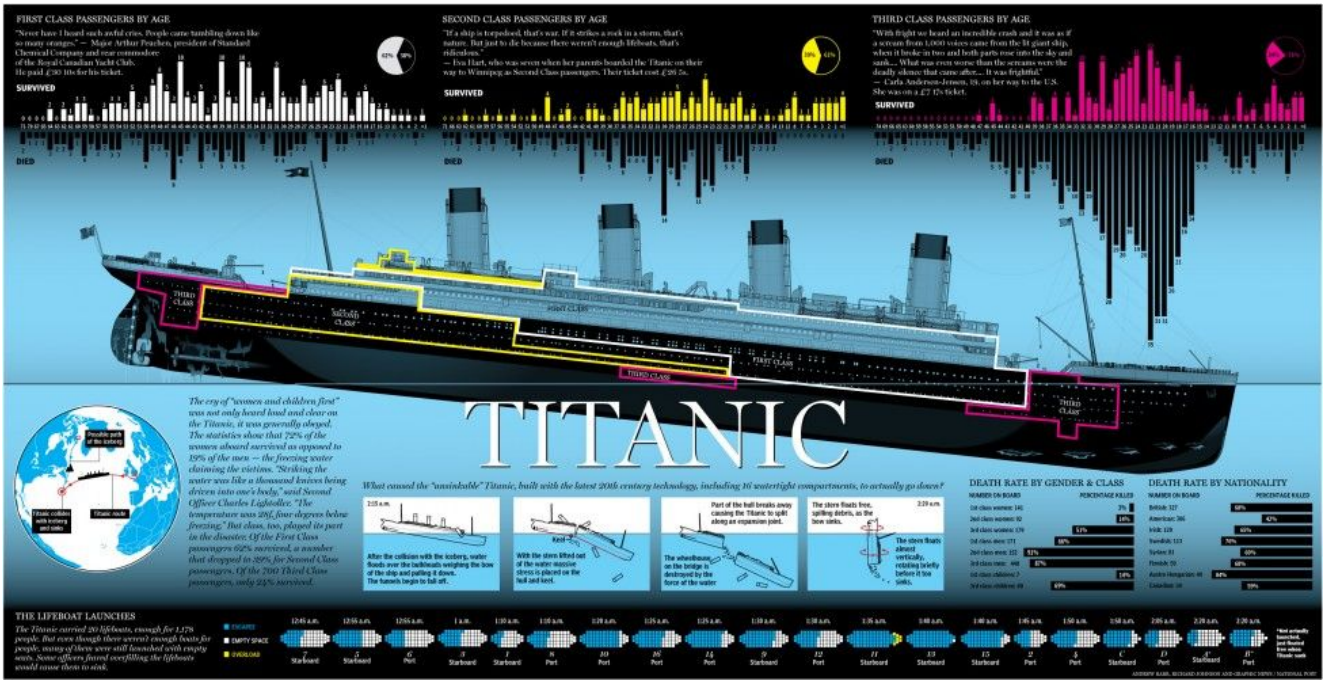
Introduction

We are given five algorithms to make machine learn: decision tree, neural networks, boosting, support vector machine, and k-nearest neighbors. Even though the goal for all of the algorithms is same: making prediction for new data by training existing data, they make different results since the processes are different. We will analyze the difference among these algorithms by testing classification problems using two different datasets: 'Titanic survivors' and 'MNIST'.

Dataset

K-fold cross validation is used to avoid bias of data. The cross-validation process is repeated five times, with each of the five subsamples used exactly once as the validation data. So, each process had 80% of different training set and 20% of different test set out of the whole data. Since this is about classification problem, the dataset, which has integer values for each attribute, or at least can be quantified, especially in some acceptable ranges, is appropriate. Considering this, the datasets chosen are:

1. **Titanic survivors**



I found that this dataset is interesting because the answer we are looking for can be binary: 1 for

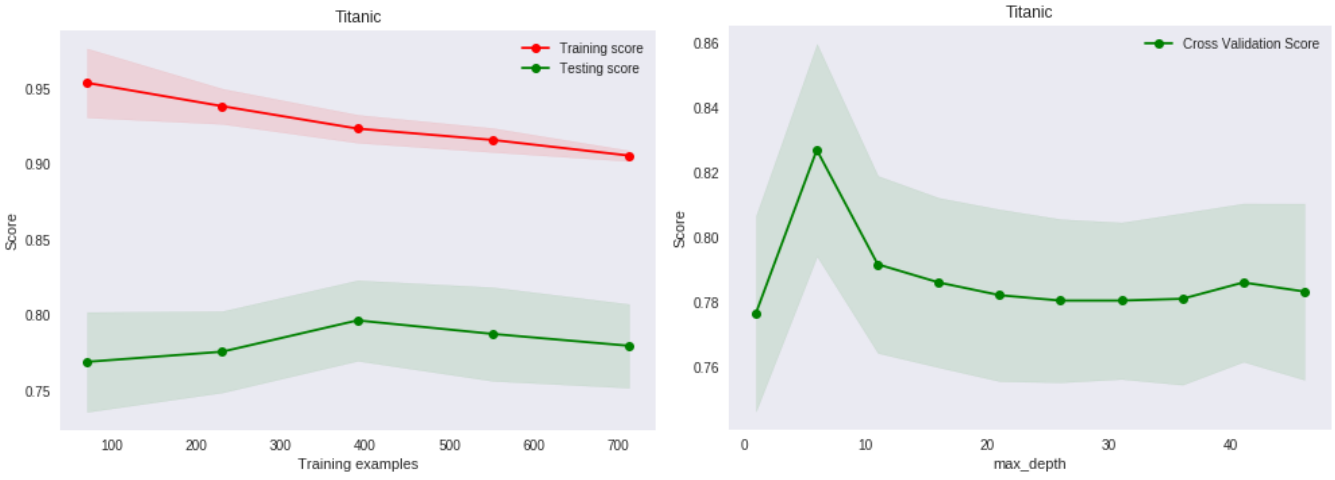
‘survived’ and 0 for ‘dead’. We assume the empty value for age is the average age for those of men and women to be reasonable. Before removing passengers’ name, their prefix is taken and different integers are assigned for each prefix as an additional attribute (For example, "Mr": 0, "Miss": 1). With same methodology, each passenger’s ticket class, sex, number of family on the boat, cabin number and port of embarkation are quantified into integers, so machine can easily classify and predict.

2. **MNIST (Modified National Institute of Standards and Technology)**

We chose MNIST data since this is one of representative uses for machine learning. It is handwritten numbers, each of which is divided by 8 X 8 small images. Since it is listed as integers according to how much the small images are filled, there is no need of feature engineering for data to be prepared for learning. It is best data for classification learning to test and filter out the best-accuracy algorithm for use of recognition of numbers by machine.

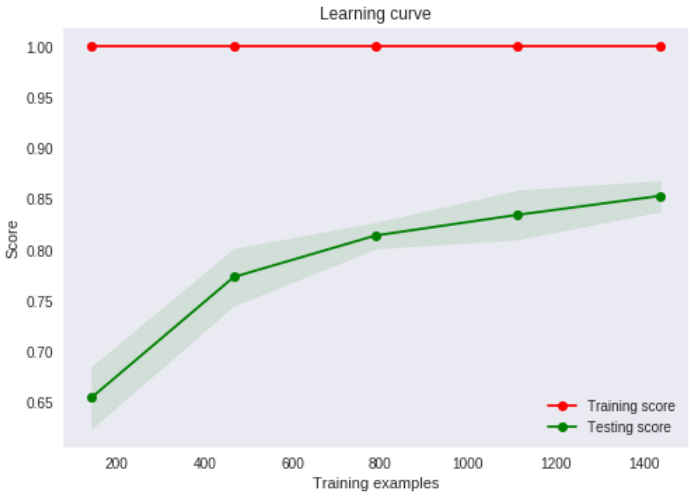
Modeling and Analysis

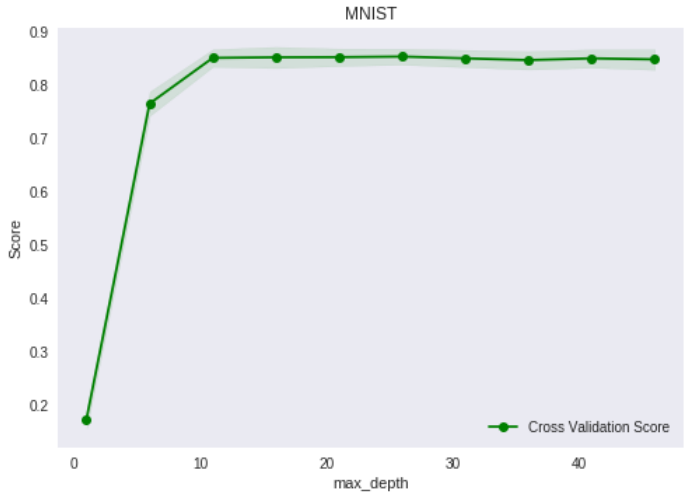
a. Decision tree



As shown above, with an average accuracy of 78.04%, the learning curve (left graph) says that the model for Titanic dataset does not show any significant change, but decreases gradually as the size of the training set increases with no pruning. One probable reason is that the additional training data

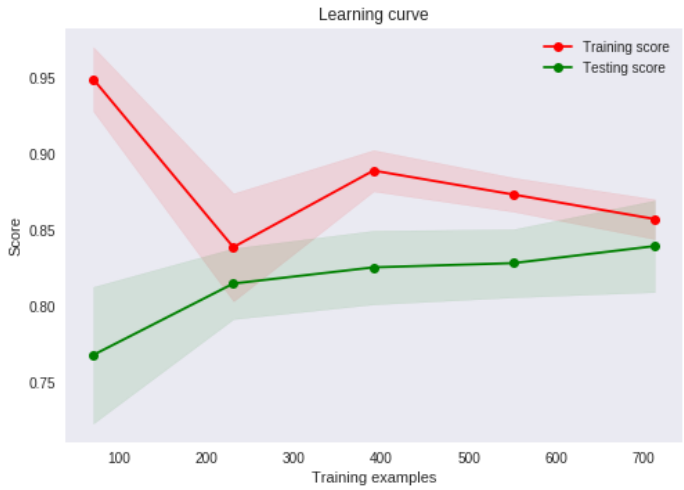
is noisy or interferes capturing useful patterns in the algorithm. Therefore, this would cause the testing accuracy to decrease. The right figure is a hyper-parameter graph showing how accuracy changes by max depth, which is the parameter of decision tree (pruning), and it hits max score (accuracy) with max depth 5. This model has big variance maybe because of randomness of output.

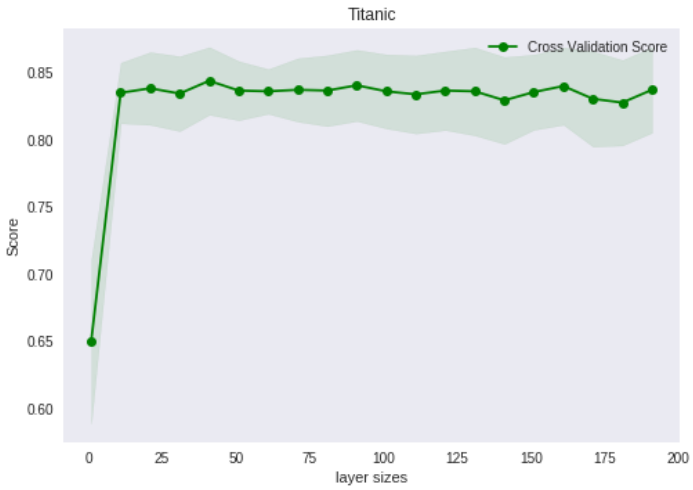




On the other hand, learning curve for MNIST gives very high accuracy (85.33%). Test score gets better by data examples and max depth increasing with very little variance. Notice that testing score makes straight line with perfect accuracy while testing score gradually increases – possible overfitting at first 200. It can be interpreted that this model is good for decision tree with bigger amount of data, so accuracy would increase by adding more data.

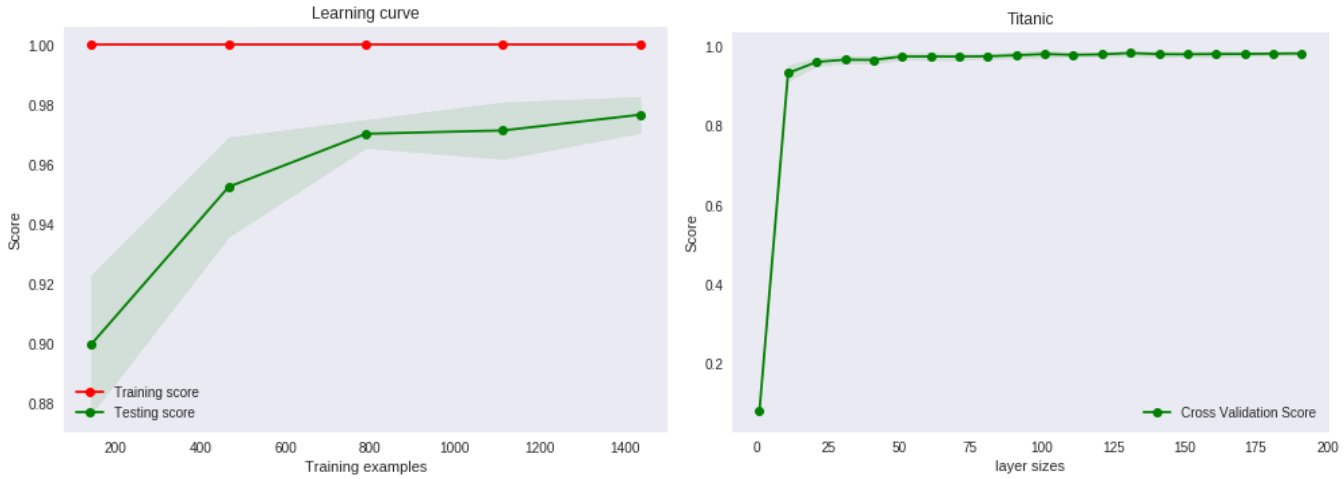
b. Neural networks





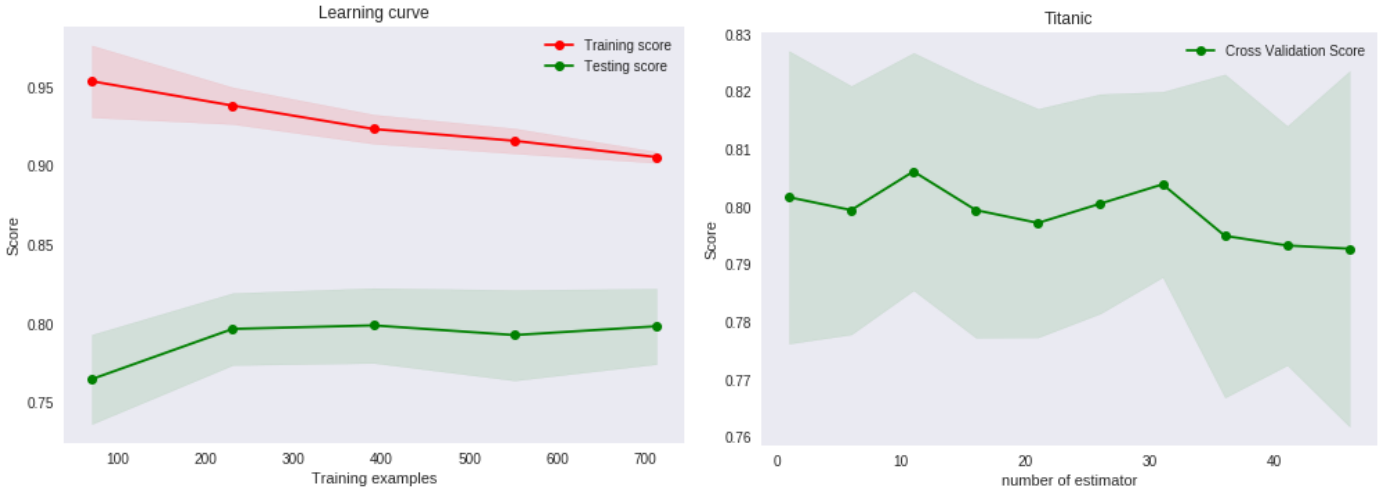
The learning curve of Titanic Survivors shows that there would be overfitting if data has small size, which fits precisely on training data, so does not predict well new data. Testing score and training score on learning curve are being merged together, which might be underftting when we gets more data. Size of hidden layers is parameter we can adjust for better prediction for neural networks

algorithm, and for this model if layers are more than 10, accuracy remains max accuracy for this prediction.

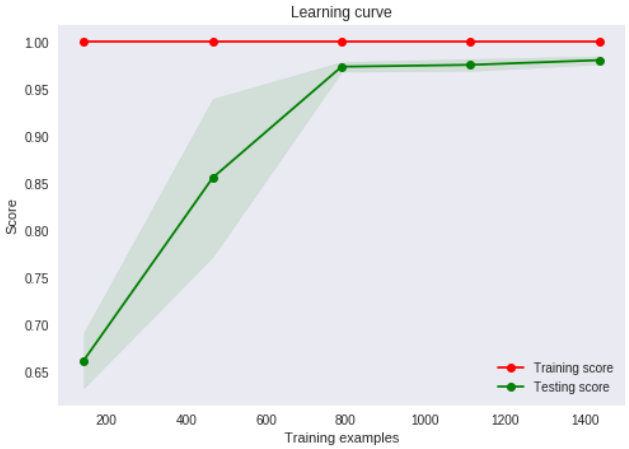


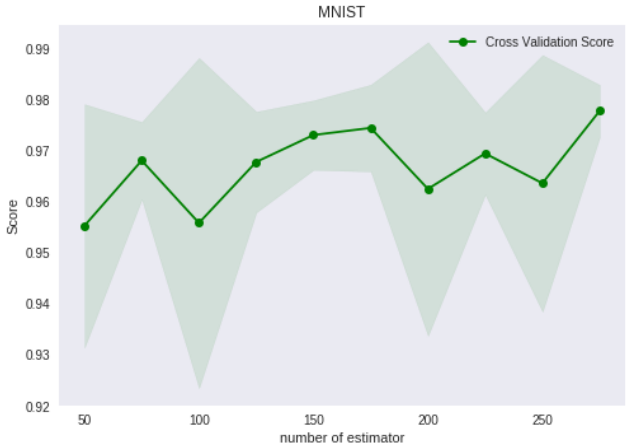
For MNIST dataset, it makes perfect accuracy for training set with no variance. As data size gets bigger, the prediction gets better. As the size of training data increases, Neural Network Algorithm predicts unforeseen data (test data) better. Same as Titanic dataset, around 10 layers, it makes its best accuracy for testing data, which is almost 100%.

c. Boosting



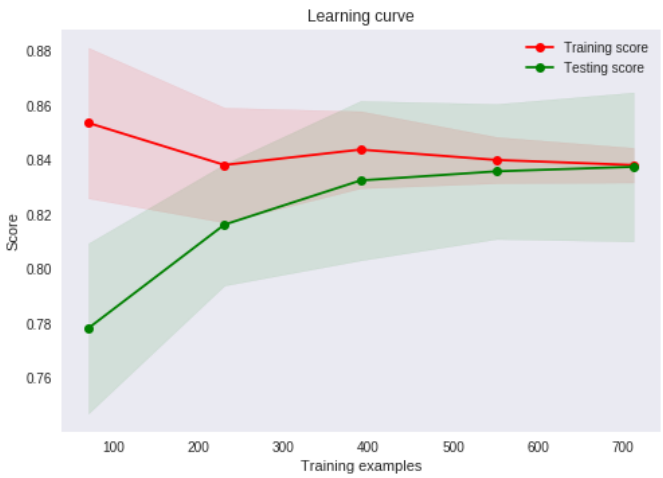
Shows very good result with high accuracy for both training and testing data with small variance. The control parameter is number of estimator, which is number of decision trees for this case. It makes better accuracy when the parameter is 10 and 30, but generally has similar score with small variance (Notice the scale of score).

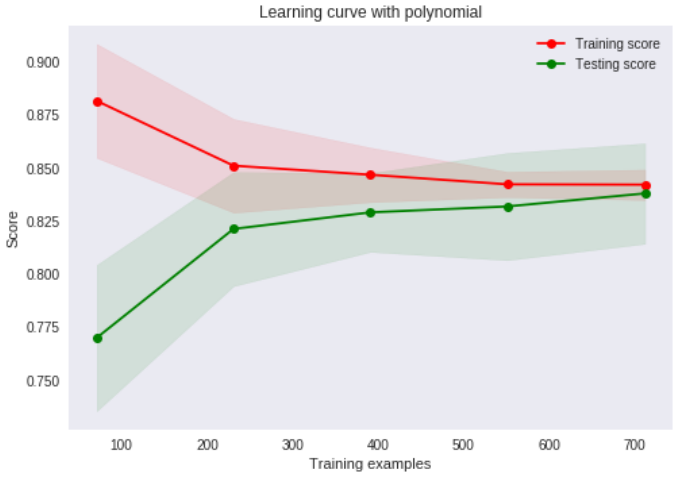




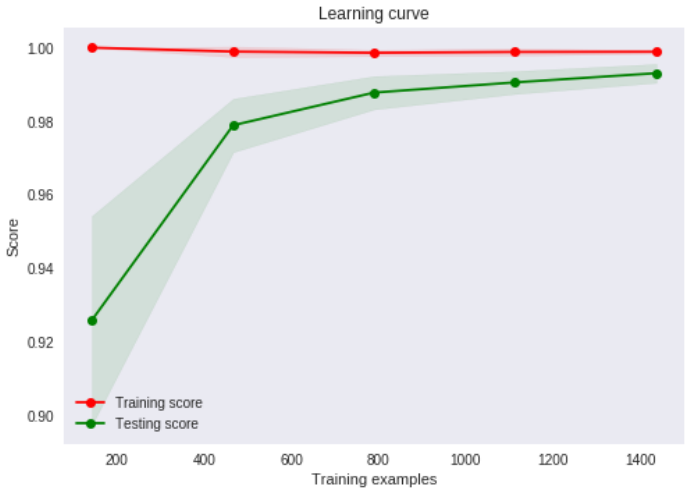
Again, for both of training data and testing data MNIST dataset makes good accuracy plot with Boosting Algorithm when its data size is over 800. It seems having huge variance when we change around the number of estimators, but the range is small (about 0.05), which mean dataset is very stable.

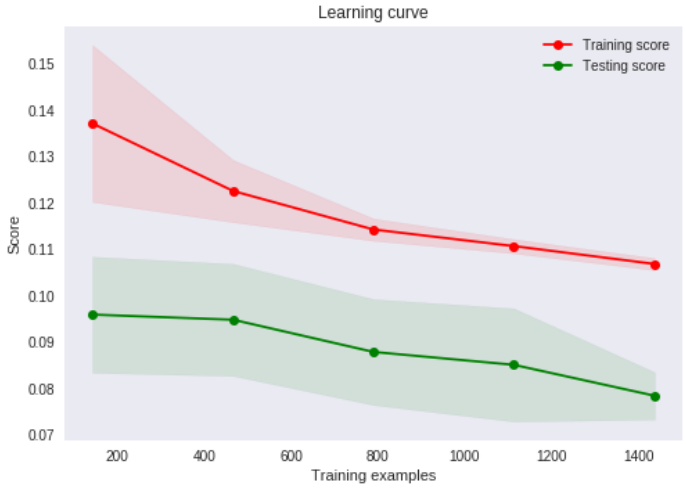
d. Support vector machine





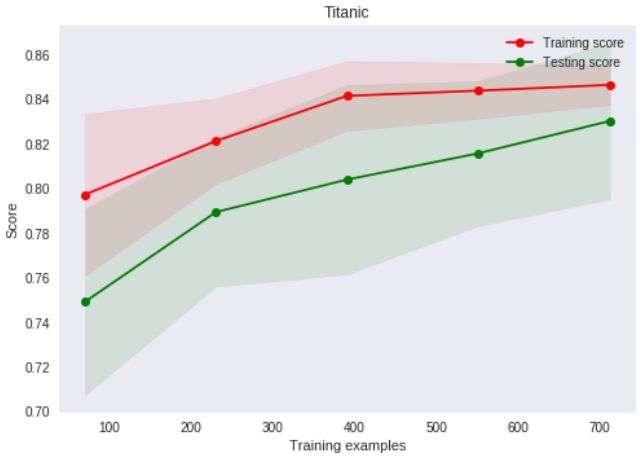
For Titanic dataset, each of kernels makes similar accuracy. The figures above are RBF kernel and polynomial kernels as examples. It has tendency to be underfitting as data set is bigger. It seems that this algorithm is not appropriate for this dataset if size of data is more than 700 examples. Since SVM is robust, it is not flexible much.

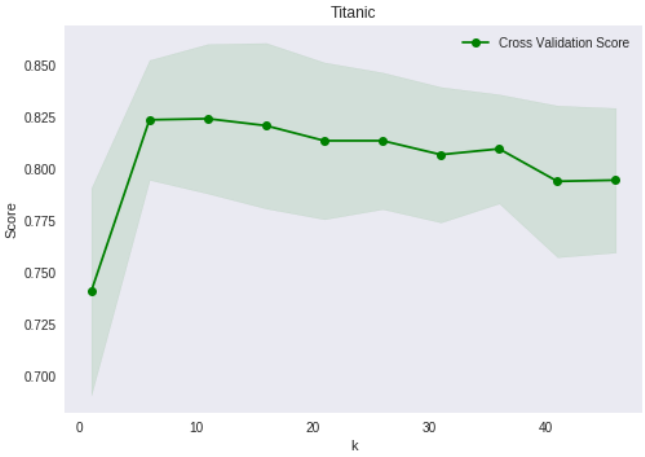




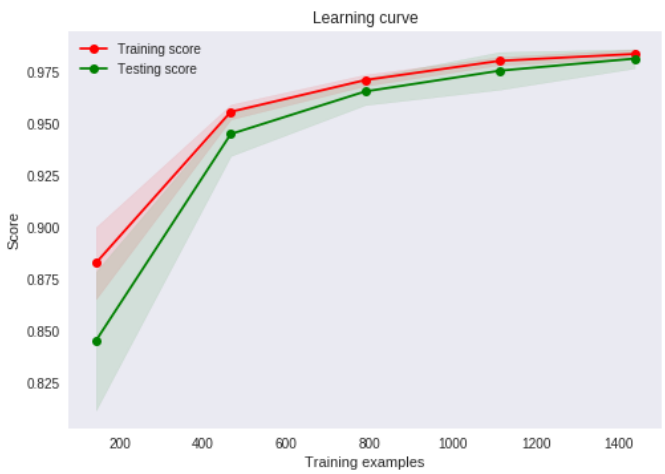
Most kernels for SVM have good accuracy (like left figure above) for MNIST, except for sigmoid (right figure) and RBF showing about 8% and 40%. Like Titanic case, it seems that It is because kernels are firm functions, so it fits really well or not.

e. *k*-nearest neighbors





This Titanic case reflects exact features of kNN model. As data set gets bigger, both training set and testing set make better prediction. And there is no evidence for overfitting or underfitting. In terms of parameter, it hits max score when k is 7 and it falls slowly. Again, it has very little variance.



Another good result for kNN. Accuracy for both training data and testing data is going to max accuracy while it decreases by k becomes bigger. It is because lots of number of k means underfitting since big range of k is covered.

Summary and Conclusion

a. **Decision tree**: Generally gives a good accuracy with relatively short process time. Easy to understand. Better to have not many examples (or attributes) in case of overfitting.
b. **Neural networks**: Takes very long time. It has a method for itself to avoid overfitting.
c. **Boosting**: Takes long time since it has multiple of decisions tree by estimator, but makes better result than Decision tree
d. **Support vector machine**: Margins significantly matters. Works well with large and clear margins, but with noise and outliers. That is, it varies a lot depending on kernels or dataset.
e. ***k*-nearset negibors:** As *k* becomes smaller, it is easy to be overfitting, and as *k* becomes bigger, it is more possible to be underfitting. Subject to the size of data. Takes instant time to train.

For Titanic Survivors dataset, boosting algorithm (, which is upper version of decision tree) made the best accuracy with the data we have. Since it has only around 700 examples and limited attributes, which do not have strong relation to output, there are different shapes by algorithms, and some possibility of underfitting.

For MNIST dataset, we could make nearly perfect prediction (98%) with training data even before we adjust parameters to make a better score. Also, testing data followed it well, too. It might be because it is clean data and has data size is very large.