Georgia Institute of Technology CS 7641 – Machine Learning

Supervised Learning using Letter Recognition and Madelon Dataset.

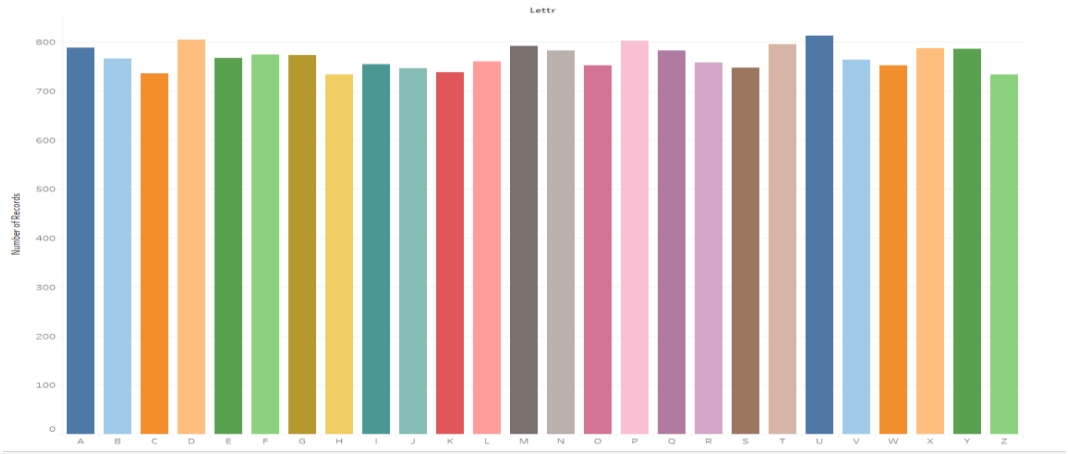
Santhanu Venugopal Sunitha (ssunitha3)

The purpose of this paper is to understand the various Supervised Learning algorithm such as – Decision Trees, Support Vector Machines, K Nearest Neighbors, Neural Networks and Boosting, by applying them to two interesting classification problems to compare their performance and characteristics.

Datasets

The datasets chosen for the analysis are:

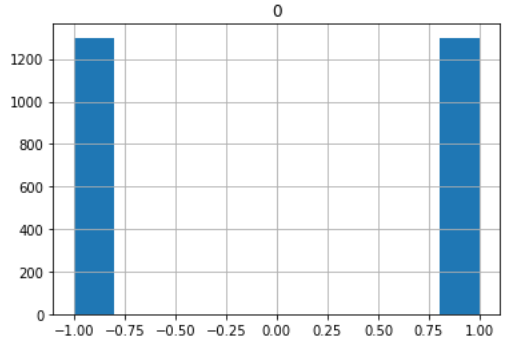
- Letter Recognition Dataset from UCI ML repository (20000 instances, 16 attributes, 26 classes) This dataset contains information about character image features. The objective is to identify each of a large number of black-and-white rectangular pixel displays as one of the 26 capital letters in the English alphabet. The character images were based on 20 different fonts and each letter within these 20 fonts was randomly distorted to produce a file of 20,000 unique stimuli. Each stimulus was converted into 16 primitive numerical attributes (statistical moments and edge counts) which were then scaled to fit into a range of integer values from 0 through 15. This is a balanced dataset as depicted in the below chart.



Class Distribution Bar Chart

- Madelon Dataset from UCI ML repository (4400 instances, 500 attributes, 2 classes) MADELON is an artificial dataset containing data points grouped in 32 clusters placed on the vertices of a five-dimensional hypercube and randomly labeled +1 or -1. The five dimensions constitute 5 informative features. 15 linear combinations of those features were added to form a set of 20 (redundant) informative features. Based on those 20 features one must separate the examples into the 2 classes (corresponding to the +-1 labels). It has a number of distractor feature called 'probes' having no predictive power. The order of the features and patterns were

randomized. The difficulty is that the problem is multivariate and highly non-linear. This is also a balanced dataset as depicted below.



Class Distribution Bar Chart

These two datasets contrast each other in an 'interesting' manner as Madelon has more attributes than Letter Recognition but Madelon has less instances than Letter Recognition. Also, Madelon is a binary classification problem while Letter Recognition is a multiclass classification problem.

Methodology

The 5 supervised learning algorithms were applied to both datasets after splitting them into training and test sets in a 70/30 ratio. Hyperparameters for the algorithms were tuned using cross validation techniques and the best parameters after the tuning were used to select the best model. The final model was tested on the test set. Model complexity curves and validation curves were plotted for each learning algorithm for further analysis. All algorithms were implemented using scikit-learn library.

Neural Networks

A multilayer perceptron was used to implement the neural network algorithm. A perceptron is a linear classifier; that is, it is an algorithm that classifies input by separating two categories with a straight line. A multilayer perceptron (MLP) is a class of feedforward artificial neural network. An MLP consists of at least three layers of nodes.

As MLP is highly sensitive to feature scaling, both datasets were scaled before training process. With regards to hyperparameter tuning, MLP requires tuning several them. So, hyperparameters like Hidden layer sizes, alpha and activation functions were tuned using cross validation and the accuracy of the MLP with optimal hyperparameter obtained from cross validation was used on the test set and the accuracy score was measured.

Accuracy Comparison

Dataset	Train Set Accuracy	Cross-Validation Accuracy	Test Set Accuracy	Training Clock Time	Testing Clock Time
Letter Recognition	0.91	0.87	0.85	5.67s	0.06s
Madelon	0.67	0.57	0.56	0.66s	0.02s

Best Hyperparameters obtained using Cross Validation

Dataset	Hidden Layer Sizes	Alpha	Activation
Letter Recognition	(32, 32, 32, 32)	relu	1e-05
Madelon	(62, 62)	logistic	1e-05

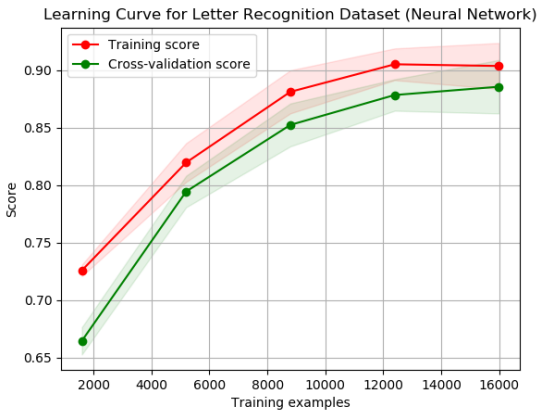
Analysis

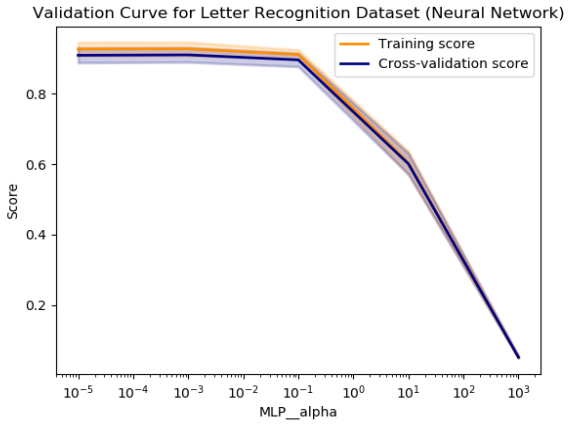
As evident from the accuracy Stats above, MLP performed well on Letter Recognition dataset and performed bit poorly on Madelon in comparison. Based on the learning curves, Letter Recognition training and cross-validation scores improved with training data and gap between the curves is narrow indicating low variance, which means that there is less overfitting of training data. Adding more data here may not improve the algorithm performance, as both curves seems to converge.

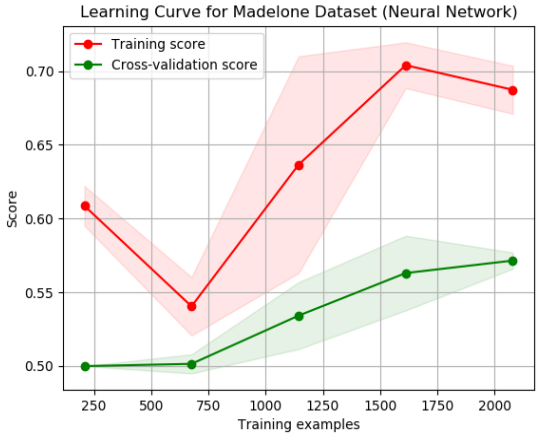
However, in the case of Madelon, there is a wide gap between both curves, which shows high variance, indicating that the model is trying to overfit the training data. Adding more data may help here in terms of performance improvement. The variance can be attributed to the high dimensionality of the dataset, in which many do not actually contribute to predicting the label and hence increasing the model complexity, which is why we have a high variance case. Hence, techniques like dimensionality reduction may also help to improve the performance of the learner.

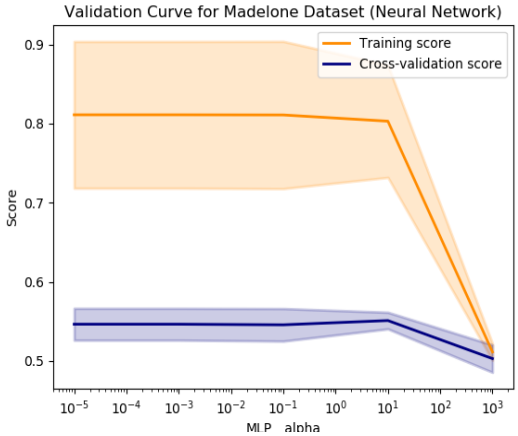
A confusion matrix comparison is made as well to further visualize which classes are being misclassified.

Comparing Learning Curves and Validation Curves

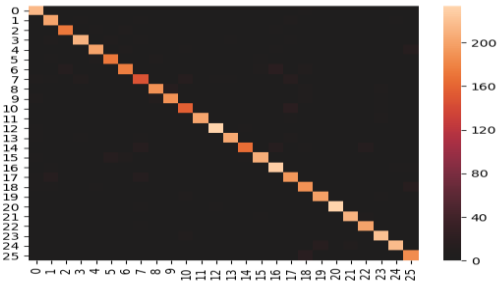


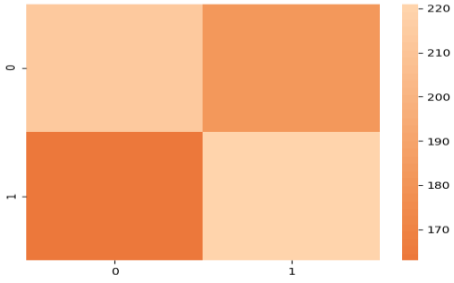






Comparison Between Confusion Matrix





Decision Tree

Decision-tree learners can create over-complex trees that can cause overfitting. Pruning can help to avoid this problem. For this analysis, Max Depth parameter was used to prune the tree (called pre-pruning).

To find the optimal Max Depth hyperparameter, cross validation was implemented with StratifiedKFold shuffling.

Accuracy Comparison

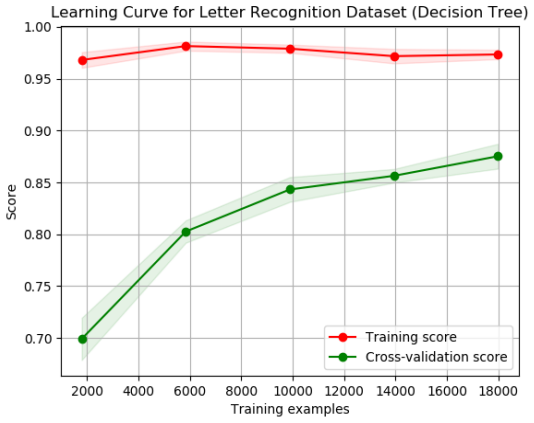
Dataset	Train Set Accuracy	Cross-Validation Accuracy	Test Set Accuracy	Training Clock Time	Testing Clock Time	Optimal Hyperparameter
Letter Recognition	0.97	0.87	0.86	0.20s	0.03s	Max Depth = 19
Madelon	0.88	0.78	0.76	0.46s	0.002s	Max Depth = 6

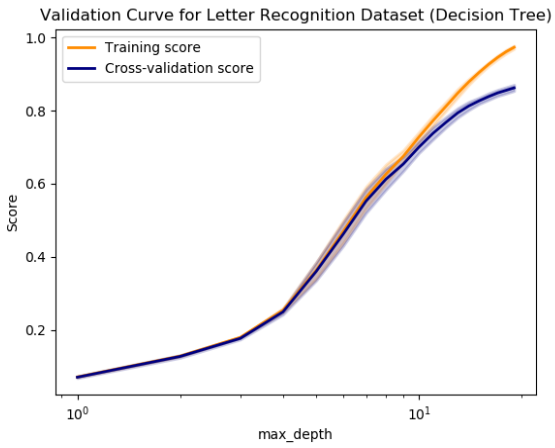
Analysis

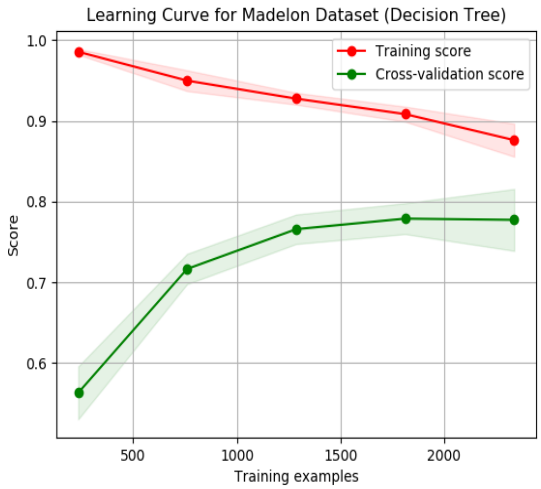
Based on the accuracy stats, decision tree performed a bit better for Letter Recognition dataset than Madelon. However, the high dimensionality of Madelon did not lower the accuracy that much in comparison to a Neural Network. This may be attributed to the pre-pruning technique, which prevented the tree from becoming complex and overfit the data.

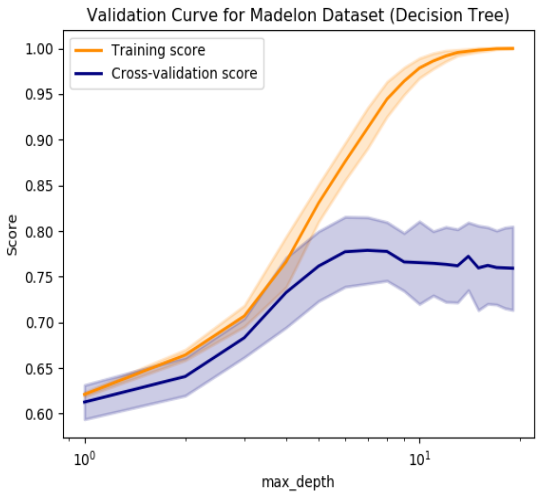
Based on the learning curves, Madelon shows high variance and hence getting more training data will help here in terms of improving the cross-validation score. Letter recognition learning curve also seems to be on the high variance side and hence more training data will help here to improve its performance or an ensemble technique like boosting would help as well as we should see below. Hence, the learner seem to overfit both datasets a bit which accounts for the low accuracy.

Comparing Learning Curves and Validation Curves









Boosting

Boosting is an ensemble method for improving the model predictions of any given learning algorithm. The idea of boosting is to train weak learners sequentially, each trying to correct its predecessor. For this analysis, AdaBoost (Adaptive Boosting) has been used with a Decision Tree Classifier as the base estimator. An AdaBoost classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.

The Max Depth for the Decision Tree learner for each dataset has been set to its optimal value based on the previous analysis (pre-pruning). Hence, this will help to understand whether boosting can improve the scores of previous Decision Tree Classifiers.

The hyperparameter tuned here is the number of estimators and the optimal value of the same is obtained using cross validation. The same is used to derive the test accuracy.

Accuracy Comparison

Dataset	Train Set Accuracy	Cross-Validation Accuracy	Test Set Accuracy	Training Clock Time	Testing Clock Time
Letter Recognition	1	0.97	0.96	14s	0.98s
Madelon	0.88	0.78	0.76	0.40s	0.004s

Best Hyperparameters obtained using Cross Validation

Dataset	Estimators
Letter Recognition	90
Madelon	1

Decision Tree Classifier Max Depth (pre-pruning)

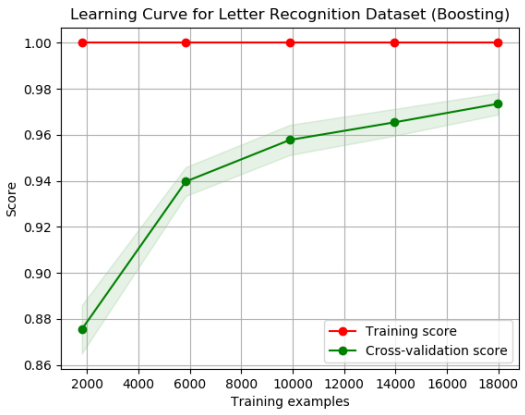
Dataset	Max Depth
Letter Recognition	19
Madelon	6

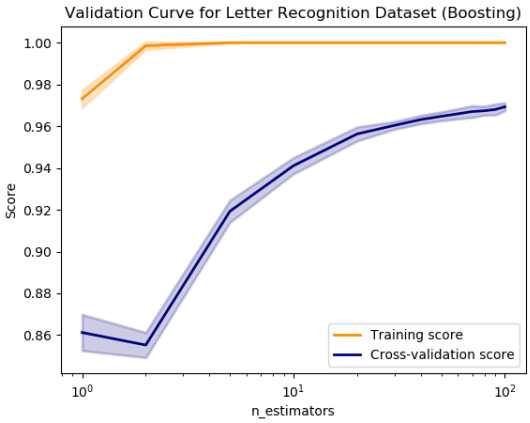
Analysis

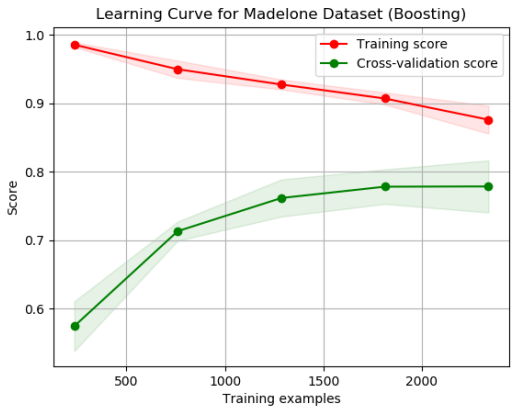
Boosting has improved the test accuracy for the Letter Recognition dataset from 86% to 96% with an optimal number of estimators of 90. The fluctuation in training score across training samples has been smoothened out by boosting and the cross-validation score was improved, implying that each instance of learners was able to reduce the error from the previous learner by focusing more on harder to classify training sample. However, in the case of Madelon, the optimal number of estimators turned out to be just 1, due to which the performance was similar to the single Decision Tree algorithm.

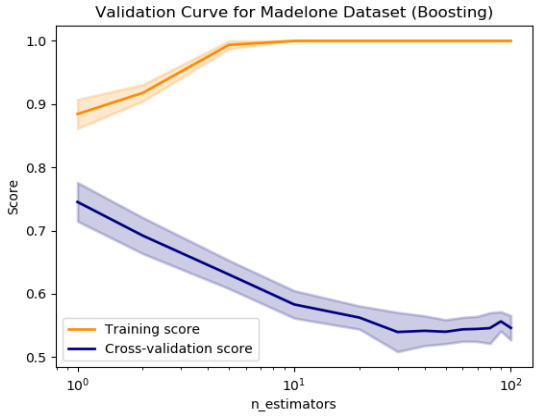
Based on the learning curves, as mentioned above for Decision Tree classifier, Madelon shows high variance and getting more data might help here. For Letter recognition dataset, validation score could be increased with bit more training samples.

Comparing Learning Curves and Validation Curves









Support Vector Machines

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. In two-dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side. SVM with 'linear' and 'rbf' kernels were used to conduct analysis on both datasets.

Hyperparameter tuning was performed on the penalty factor - C using cross validation using StartifiedKFold shuffling.

Linear Kernel

A linearSVC was used instead of a regular SVM with kernel='linear'. The linear-SVM uses a linear kernel for the basis function and is a more flexible and scalable implementation of SVC with linear kernel.

Analysis

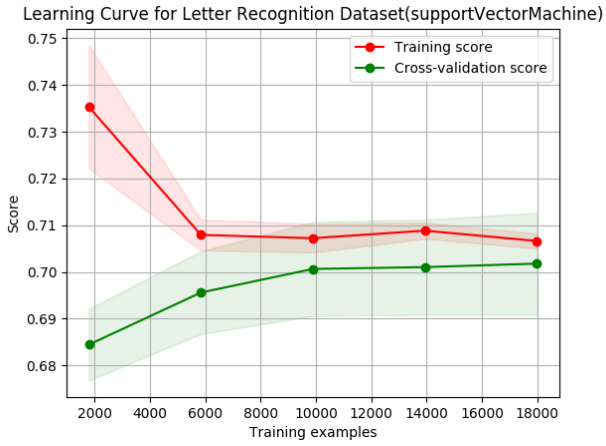
Looking at the learning curve for Letter Recognition dataset, a high bias case is being observed wherein training accuracy seems to decrease with sample size and hence adding more data here is not going to help with the performance. This observation seems to contrast with the above three learners.

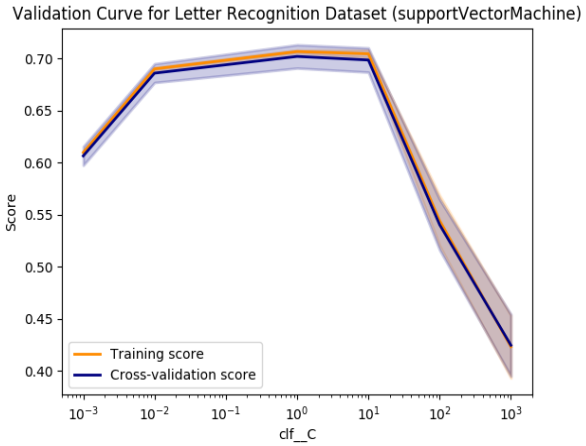
Madelon learning curve shows high variance as the training accuracy seems to decrease and the cross validation accuracy seem to increase a bit with training size. So, interestingly getting more data here may help improve the accuracy of cross validation.

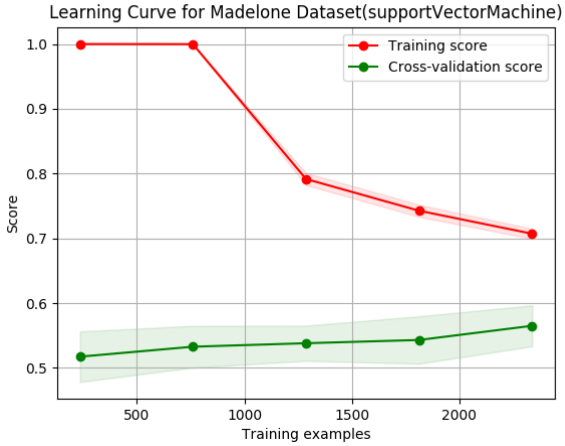
Accuracy Comparison

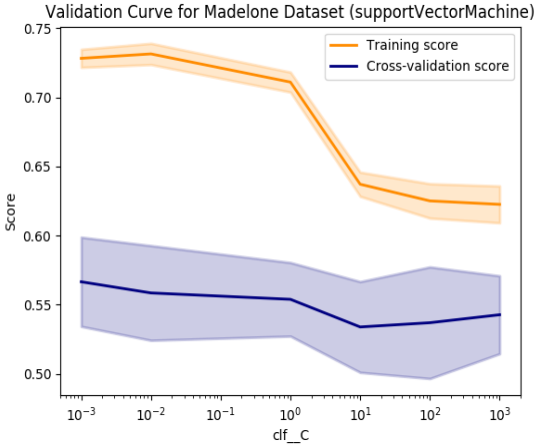
Dataset	Train Set Accuracy	Cross-Validation Accuracy	Test Set Accuracy	Training Clock Time	Testing Clock Time	Optimal Hyperparameter
Letter Recognition	0.70	0.70	0.70	22.91	0.03	C = 1
Madelon	0.70	0.57	0.53	0.10	0.007	C = 0.001

Comparing Learning Curves and Validation Curves









RBF Kernel

An RBF kernel uses a Radial Basis Function for classification. As evident from the Stats and Learning curves, RBF outperforms LinearSVC for Letter Recognition dataset. The training accuracy is still around the maximum and the validation score could be increased with more training samples.

Accuracy Comparison

Dataset	Train Set Accuracy	Cross-Validation Accuracy	Test Set Accuracy	Training Clock Time	Testing Clock Time
Letter Recognition	1	0.97	0.96	4.4s	4.3s

Madelon	1	0.59	0.56	2.14	0.97

Best Hyperparameters obtained using Cross Validation

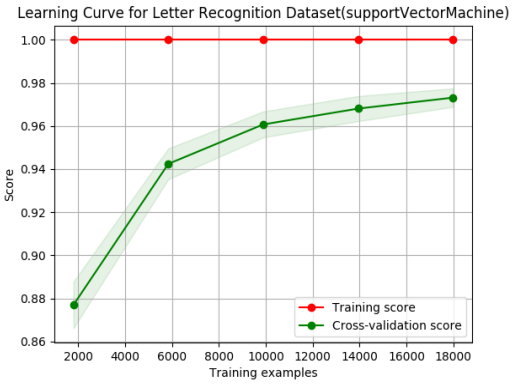
Dataset	Penalty C	Gamma
Letter Recognition	1000	0.001
Madelon	1	0.0001

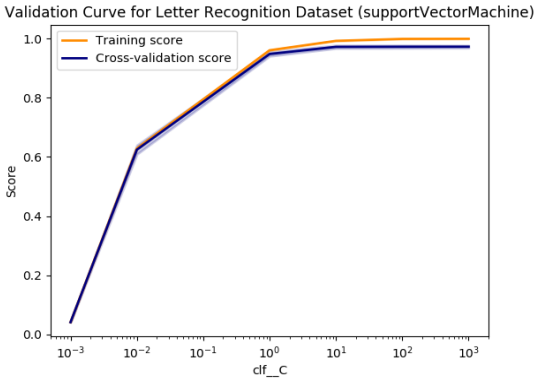
Analysis

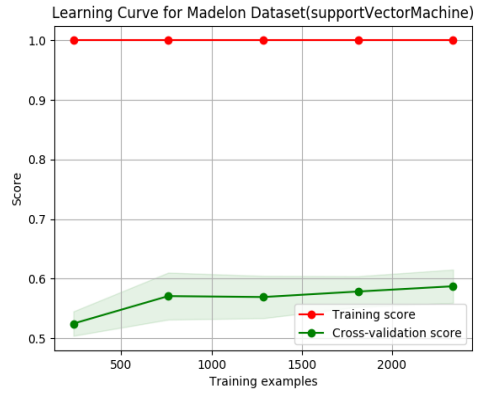
The learning curve for Letter Recognition looks similar to Boosting and hence more data can help improve the cross validation accuracy. RBF performed miles above Linear kernel here.

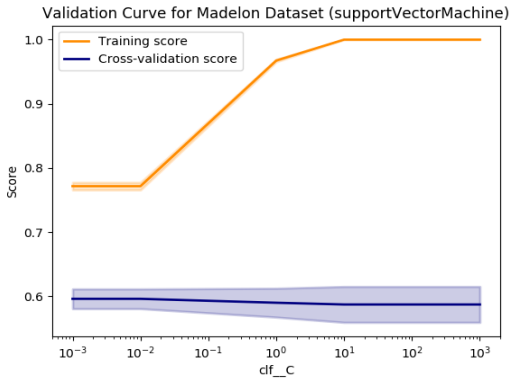
The learning curve for Madelon shows a large gap between training and cross validation curves which indicates that adding more data wont help. The classifier has low accuracy probably due to the high dimensionality and non-linearity of the data.

Comparing Learning Curves and Validation Curves









K Nearest Neighbors

KNN is a non-parametric, lazy learning algorithm. It makes prediction by finding the label of the K nearest data points of the given data point using a distance function.

The best hyperparameter K was selected using cross validation with StratifiedKfold shuffling.

Accuracy Comparison

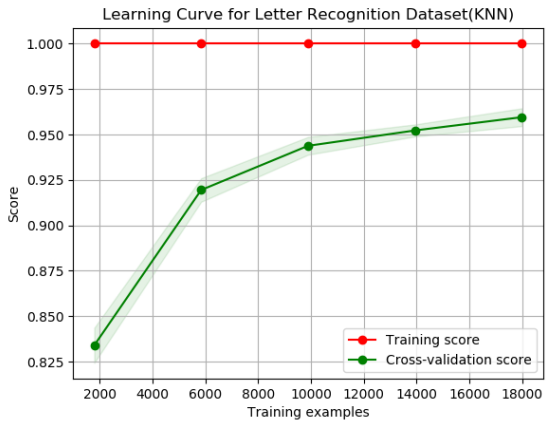
Dataset	Train Set Accuracy	Cross-Validation Accuracy	Test Set Accuracy	Training Clock Time	Testing Clock Time	Optimal Hyperparameter	
Letter Recognition	1	0.95	0.94	0.27	2.79s	K = 1	
Madelon	0.79	0.76	0.73	0.03	1.81s	K = 25	

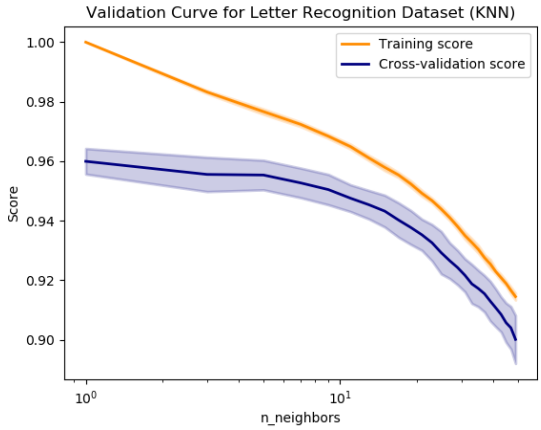
Analysis

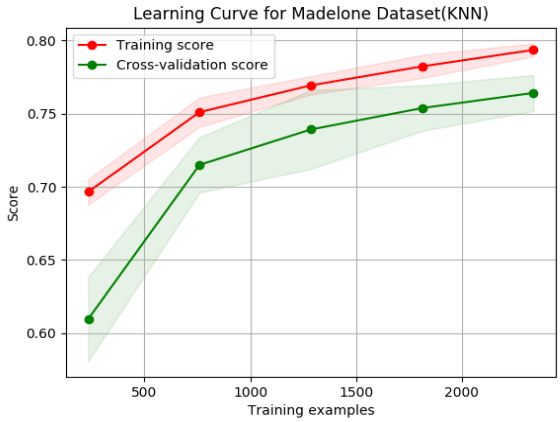
Like other learners, KNN seems to perform well on Letter Recognition. However, as the accuracy of KNN can be severely degraded with high-dimension data because there is little difference between the nearest and farthest neighbor. Hence, it does not perform well for Madelon dataset.

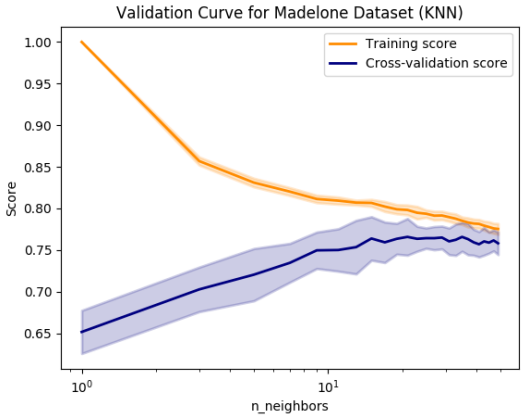
Based on the learning curves, Letter recognition dataset training score is still around the maximum and the validation score could be increased with more training samples. With regards to Madelone, the training score and the cross-validation score are both not very good at the end and hence, more data will not help here.

Comparing Learning Curves and Validation Curves









Conclusion

Based on the above analysis, the performance of the supervised learners will depend on the nature of the dataset and tuning of various hyperparameters. All learners, except for Linear SVM, performed well on Letter Recognition dataset. SVM with RBF and Boosting seems to have the best accuracy scores (96%) among others.

The training time for Boosting was very high due to high number of estimators while SVM with RBF took less time comparatively. However, Boosting took very minimal time to predict compared to SVM with RBF.

Hence, in terms of training time, SVM with RBF would be the best choice while in terms of testing time Boosting would be the best choice.

All leaners did not perform well on the Madelon dataset with all learners except K nearest neighbors and SVM with RBF, showing promising accuracy improvement with more training data.

Both Decision Tree and Boosting showed 76% accuracy on Madelon. With its lower training and testing clock time, a simple Decision Tree classifier would be the ideal choice for this dataset.

Learner	Letter Recognition Test Set Accuracy	Madelon Test Set Accuracy
Decision Tree	**86%**	**76%**
Neural Network	**85%**	**56%**
Support Vector Machines	**RBF - 96%** **Linear - 70%**	**RBF - 56%** **Linear - 53%**
K Nearest Neighbors	**94%**	**73%**
Boosting	**96%**	**76%**