Supervised Learning

Kunming Zhu
kzhu81@gatech.edu

ABSTRACT

This paper applies five algorithms (Decision tree with pruning, Neural networks, Boosting, Support vector machines, and k-nearest neighbors) to analyze two datasets: digits and wine by using scikit-learn.

ABOUT DATASETS

Both datasets are from scikit-learn, so we do not need downloading from other websites, and they are well-formatted, so we can focus on our works.

Digits is a dataset about human being's hand-writing digits, and there are 1797 samples with 10 classes (from 0 to 9) and 8x8 features. Training a model to read something is written by human beings is very interesting for me.

Wine is a dataset of 178 samples with 3 classes and 13 features. Comparing wine to digits, wine is smaller and with fewer classes, so wine should behave very differently in models. Besides, I love wine so much.

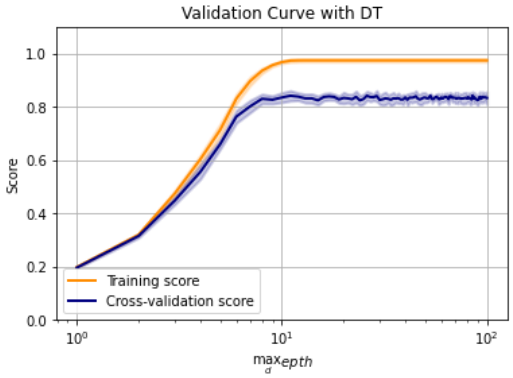
To reduce bias, I scaled datasets, and I also transferred digits dataset from 8x8 into 64x1.

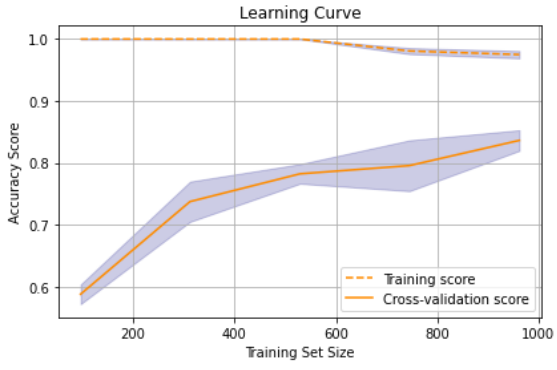
DECISION TREE

Dataset: Digits

At first, we find our best model with pruning, then the validation curve and learning curve as below.

From the validation curve graph, we can see both training score and validation score stop growing because the value of max depth reaches 10 which is the value





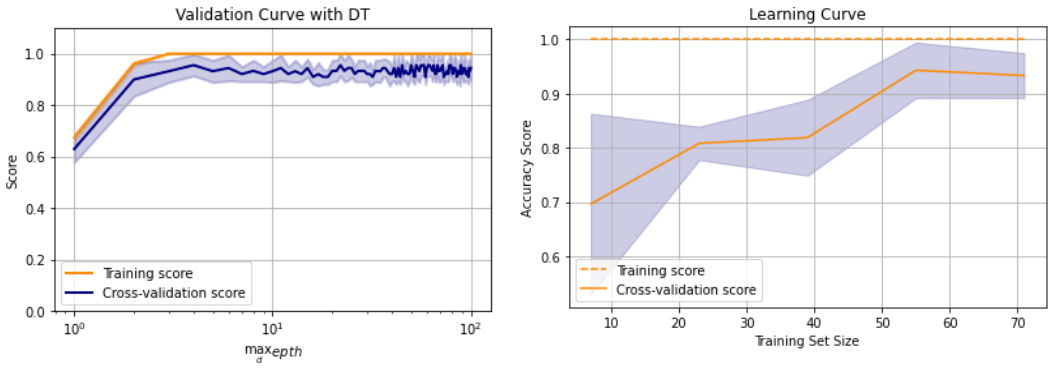
of classes in this dataset, so we do not need to set the value of max depth higher than 10.

From the learning curve graph, we can see the learning curve begins to curve down when the training set is more than 600, but the cross-validation score keeps growing after that, and both scores are in the reasonable range, so the test size set should be fine.

At last, the accuracy is around 85.86%.

Dataset: Wine

The graphs of this dataset:



Because the number of classes is 3, the training score stops growing and reaches 1 when the value of max depth reaches 3, and the validation score is stable before the max depth more than 10, so the max depth should be 3.

The learning curve keeps very well, and the cross-validation curve begins to fall after the training set size is greater than 55. There is no overfitting happened.

At last, the accuracy is around 89.89%

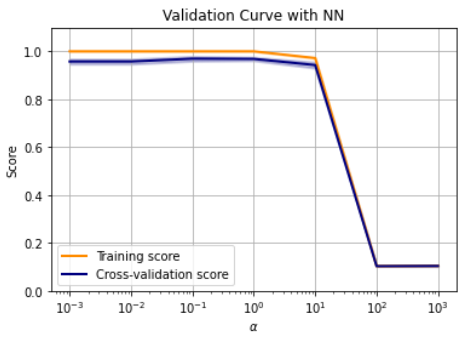
Decision Tree Summary

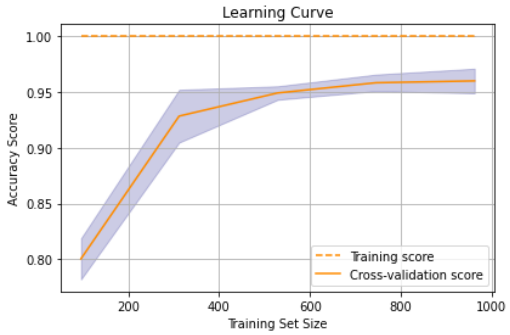
The decision tree model has better accuracy on the wine dataset; it reaches the max depth faster, and it needs a much smaller training set size to work well.

At all, we can say that the decision tree model works better when the dataset is smaller and simpler.

NEURAL NETWORK

Dataset: Digits



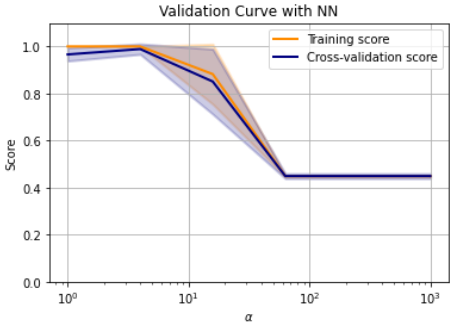


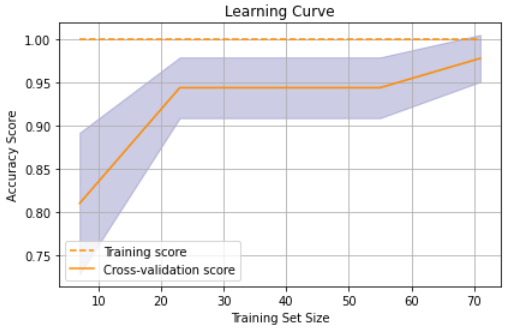
We can see the training score begins with 1 and the validation score is almost 1, but both of them begin to fall when the value of alpha is 8, and when alpha is 10, both of them fall so rapidly, so the best value should be 10.

In the learning curve graph, we can see the training score keeps every good, and the cross-validation score keeps growing when we increase the training set size, but the total number of samples in this dataset is 1797, so we already reach the edge of our dataset, but the model still has space to be improved. There is no overfitting happened.

The accuracy is about 95.45%

Dataset: Wine





This dataset is much smaller than digits, so the model runs much faster. In the validation curve graph, we can see both training score and validation score begin to fall when the value of alpha is around 6, so the best alpha value should be 6. For the learning curve, we can see the cross-validation score keeps increasing when we increase the training set size, and the error is getting narrow, so in this dataset, there is no overfitting.

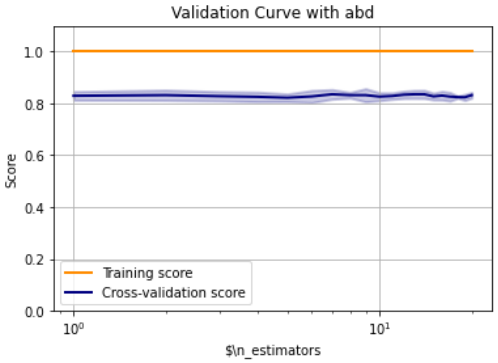
The accuracy is about 98.87%

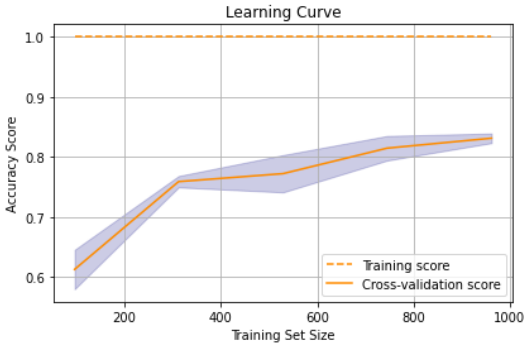
Neural Network Summary

The accuracies for both datasets are good, and both of them are not overfitting, so the model works very well for both datasets, on the other side, that means both datasets cannot tap the potential of this model.

BOOSTING

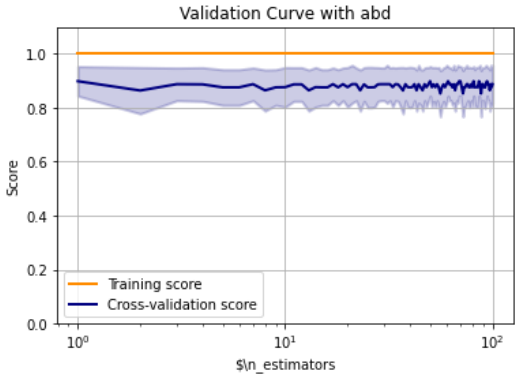
Dataset: Digits

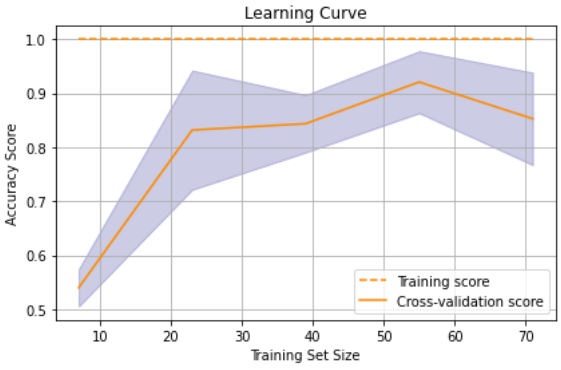




From the validation curve graph, we can see the value of n_estimators cannot affect the training score. That could be the reason for the dataset.
From the learning curve graph, we can see there is no overfitting, and the training score keeps 1, at the same time, the cross-validation score keeps increasing when we increase the training set size. If the dataset is bigger, we can have a better result.
The accuracy is around 84.68%.

Dataset: Wine





When the value of n_estmators increased, the validation score begins to be unstable, so the best value of n_estmators should be 1.
From the learning curve, we can see the model is overfitting when the training set size is greater than 55.

The accuracy is 95.51%

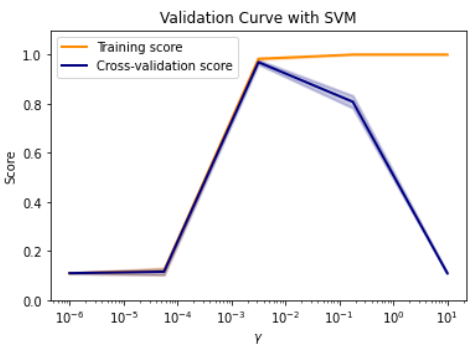
Boosting Summary

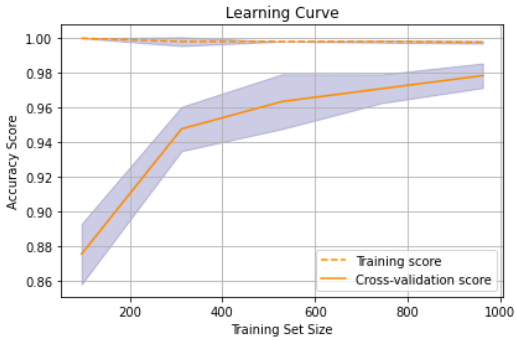
This model is interesting. Our bigger dataset is not overfitting, but our smaller dataset is overfitting, and we can figure it out by comparing the accuracies. If I reduce the test size for dataset wine, the accuracy can be improved.

After all, I think this model is very good for the dataset which is small and has few classes and features.

SVM

Dataset: Digits



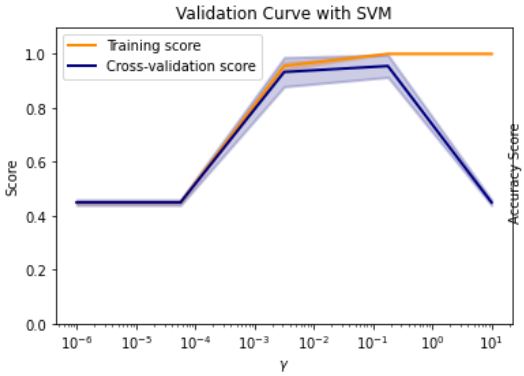


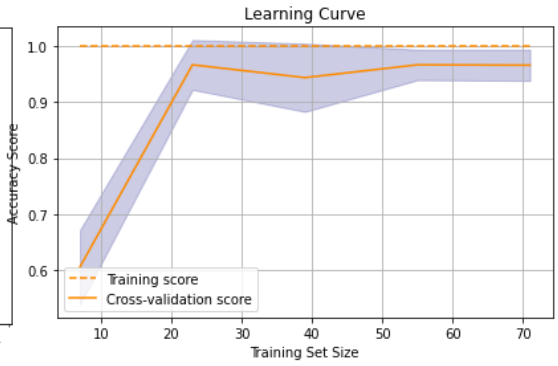
From the validation curve graph, we can find the best value of gamma is around 0.005 for this dataset.

We do not see any overfitting in this model for the dataset, but the training score shakes a little bit then is fine. The cross-validation score keeps increase and the error is decreasing when we increase the training set size.

The accuracy is 97.81%.

Dataset: Wine





On the validation curve graph, both curves begin with a low value and increase to 0.9 by increasing the value of gamma, but the validation begins to fall when the value of gamma is 0.1, so the best value of gamma is 0.1.

For the learning curve graph, we can see the training score is stable, and the cross-validation score keeps increasing when we increase training set size, but it gets stable after the value of training set size is 55, so although we do not see overfitting, but adding more training set should not affect this model.

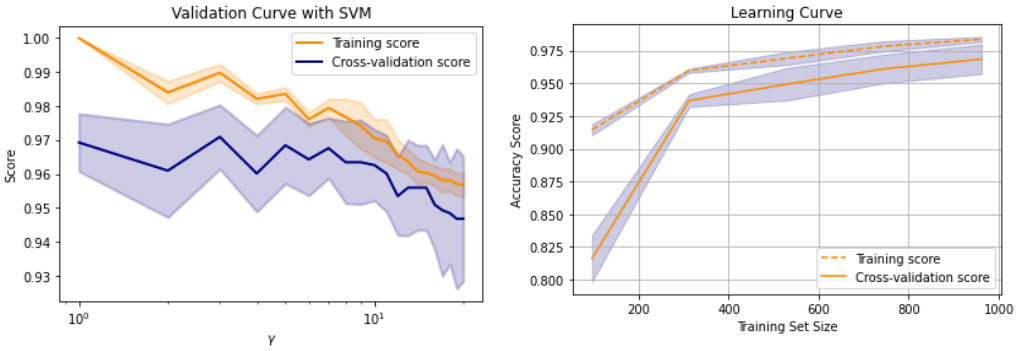
The average accuracy is 98.88%

SVM Summary

This model works very well on both datasets. We can find the best values of gamma, and there is no overfitting in both datasets. Comparing the two datasets, we can find the accuracy of wine is higher than the accuracy of digits. I believe the reason for it should be the cross-validation score in wine is higher than the one in digits. From this model, we can learn that if the graph is better, the final accuracy is better in this model.

KNN

Dataset: Digits



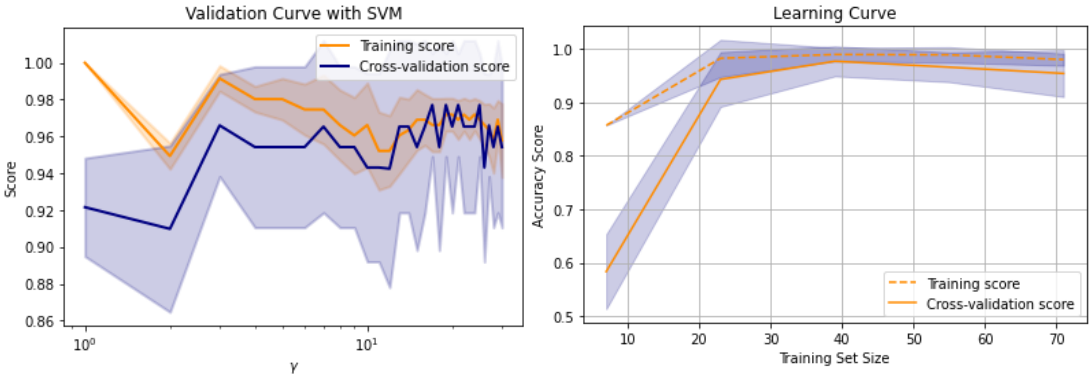
The training score and the cross-validation score all begin to fall when the value of n_neighbors increases and the errors of them are getting higher too, so the best value of n_neighbors should be 1.

The learning curve shows us that by increasing the training set size, both scores are getting better, but the errors are getting larger too, and there is no overfitting.

Comparing the two scores, I think if we can increase the training set, the model could behave better.

The accuracy is around 96.97%

Dataset: Wine



The graph of the validation curve is too messy, but we can see both scores are better when the value of n_neighbors is around 5, but the errors are still not good.

From the learning curve graph, we can see the model begins to be overfitting when the training set size is more than 40, which means we already reach the best point in this model for the dataset.

The final accuracy is 97.75%

KNN Summary

This model has so many errors in both datasets, and learning curves are both fine.

Considering the errors and unreasonable scores in validation curve graphs, this model does not look good for these datasets, but the accuracies are good. That may be caused by the datasets are clear and almost do not have noises.

Comparing two datasets, this model behaves fairly, so we cannot say what kind of dataset this model is good at.

SUMMARY

Accuracy

Accuracies Table:

	DT	NN	BOOSTING	SVM	kNN
Digits	**85.86%**	**95.45%**	**84.68%**	**97.81%**	**96.97%**
Wine	**89.89%**	**98.87%**	**95.51%**	**98.88%**	**97.75%**

From the table, we can find the best model for both datasets is SVM, and even NN is a little worse than SVM.

If we compare the validation curve graphs, we can find out that only SVM's graphs have all three stages, so we can say only the SVM model fully worked in these datasets.

On the other hand, even NN does tap its full potential, but it still behaves very well. It seems that whatever the dataset is complex or simple, large or small, the NN model all works very well and accurately enough.

Time

Time Table:

	DT	NN	BOOSTING	SVM	kNN
Digits	**0.029s**	**2.578s**	**0.029s**	**0.177s**	**0.120s**
Wine	**0.009s**	**0.222s**	**0.009**	**0.010s**	**0.014s**

For both datasets, we know NN models are the ones that take the most time, and DT ones are the fastest ones on training and predicting.

Models consume time as the size of datasets, and there is no evidence that the models can be more accurate if they consume more time.

General

Neural network is the best and most general machine learning model at all. In some datasets or situations, other models could behave better than a neural network, but users need to figure out which model is the one.

In this test, the number of hidden layers is 3, and the size of each layer is equal to the number of features in each dataset.

There are still some ways to improve neural network model, such as adding more hidden layers, changing the size of layers, and do some grid searching, so the accuracy of neural network could be improved.

The only disadvantage of neural network is the execution time. Neural network models need much more time than other models to execute. If the dataset is more complex and bigger, the time of execution may be hours or days.

ALGORITHMS ANALYSIS

From our summaries, we know that only the SVM model is trained very well for the datasets, so in this part, we will see how other models work by adding grid search.

Decision Tree

After doing a grid search, the result of decision tree for Digits:

```
Best DT score:0.83
Best DT parameters:{'max_depth': 50, 'max_features': 10,
'min_samples_leaf': 1, 'min_samples_split': 2}
```

And the final accuracy is 83.00%. Comparing the result which we got above, this result is even worse, so it looks like that the decision tree is not good for this dataset, and it is easy to be overfitting.

Result of decision tree for Wine:

```
Best DT score:0.98
Best DT parameters:{'max_depth': 26, 'max_features': 4,
'min_samples_leaf': 1, 'min_samples_split': 3}
```

The accuracy on test is 83.15%. Comparing 89.89%, we can get the same result with the result for digits.

Neural Network

Digits:

```
Best NN score:0.98
Best NN parameters:{'alpha': 0.01, 'hidden_layer_sizes': 61,
'learning_rate_init': 0.07196856730011521}
```

Accuracy: 89.22% (95.45%)

Wine:

```
Best NN score:0.98
Best NN parameters:{'alpha': 1e-06, 'hidden_layer_sizes': 11,
'learning_rate_init': 0.07196856730011521}
```

Accuracy: 95.50% (98.87%)

We can see for both datasets, the model is a little overfitting, but the accuracies are still very good, so this model is stable and accurate.

Boosting

Digits:

```
Best Boosting score:0.70
Best Boosting parameters:{'learning_rate':
0.060000000000000005, 'n_estimators': 28}
```

Accuracy: 69.86% (84.68%)

Wine:

```
Best Boosting score:0.98
Best Boosting parameters:{'learning_rate': 0.5,
'n_estimators': 13}
```

Accuracy: 89.89% (95.51%)

We can see this model is overfitting, and when the dataset is big, such as digits, it works worse. In this case, this model is not good for handling a big dataset.

SVM

Digits:

```
Best SVM score:0.99
Best SVM parameters:{'C': 10.0, 'gamma': 0.01}
```

Accuracy: 97.98% (97.81%)

Wine:

```
Best SVM score:0.98
Best SVM parameters:{'C': 0.5754399373371566, 'gamma':
0.02089296130854041}
```

Accuracy:

98.88% (98.88%)

This model behaves so perfectly on both datasets, and we cannot see any overfitting.

KNN

Digits:

```
Best kNN score:0.97
Best kNN parameters:{'leaf_size': 1, 'n_neighbors': 1, 'p': 2}
```

Accuracy: 98.15% (96.97%)

Wine:

```
Best kNN score:0.98
Best kNN parameters:{'leaf_size': 1, 'n_neighbors': 9, 'p': 1}
```

Accuracy: 93.26% (97.75)

This model works well, and there is no overfitting.

Summary

All the best parameters of results are in the range of settings.

Comparing behaves, the SVM, NN, and kNN models are accurate with and without grid search, but NN consumes the longest time, and kNN consumes less time than NN, and SVM is fast.

After all, depends on the results we got so far, the SVM model is the best for both datasets. It is fast to train, and it does not need a big dataset training to be accurate. The NN and kNN models are good too, but they need more time on training to behave well. The decision tree and boosting models are not accurate enough for usages.

REFERENCES

Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.
API design for machine learning software: experiences from the scikit-learn project, Buitinck *et al.*, 2013.