CS7641 Homework 1 – Supervised Learning

Junfei Xia
Jxia83@gatech.edu

Overview

There are two REAL dataset from UCI used in this classification research. The first one is the Algerian forest fires dataset. It is a dataset recording the forest situations (on fire or not on fire) from June 2012 to September 2012. There are 4 directly measured features and 5 processed indexes in this dataset. **The interesting question about this dataset is that whether it is possible to predict forest fire only based on directly measured features, and among the four features (Temperature, Relative Humidity, Wind speed and Rain fall), which one is the most important feature.** The identification of the most important feature might save a lot of observational and computational resource. Moreover, the importance of each observed features helps develop forest fire index.
The second dataset is the Room Occupancy Estimation Data Set, which is a dataset recording some environment features and number of people inside the room. There are four different features including temperature, light, sound and CO2. It is a multiclass dataset. **The interesting question about this dataset is about multiclass classification. Among the five algorithms, which one performs best. Another interesting thing is the accuracy of the prediction of the number of people based on the observed environment data.** The understanding of this question may help smart control of light and AC, which is related to energy saving and climate change.
The first dataset is a small dataset with 244 instances and the second one is much bigger, with 10129 instances. One more interesting thing is about the performance of each algorithm for different size of datasets.

Test case 1: Algerian forest fires dataset
https://archive.ics.uci.edu/ml/datasets/Algerian+Forest+Fires+Dataset++
This dataset includes 244 instances.
Features used in this study:

- **Temp:** temperature noon (temperature max) in Celsius degrees: 22 to 42
- **RH:** Relative Humidity in %: 21 to 90
- **Ws:** Wind speed in km/h: 6 to 29
- **Rain:** total day in mm: 0 to 16.8

Class: 1. Fire 2. Not Fire

Algorithms:

1. Support Vector Machines(default)

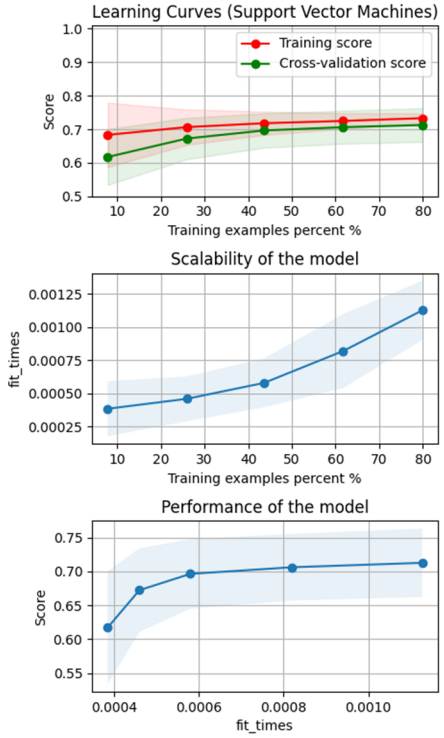

Figure 1. Learning curves of SVM.

The average learning score of SVM for the forest fire dataset is about 0.7, which is not a high score. The overall training score is higher than the cross-validation score. The training time exponentially increases with the training size. I think 70% is the best training size.

Since the dataset used has 4 features, SVM can only train two features at a time. So each time SVM choose a feature pair and start training. Since there are 4 features, there are 6 different pairs of features, which increase the training time.

I assume the weak performance of the SVM is because of the small size of the dataset and the way of training feature pairs.

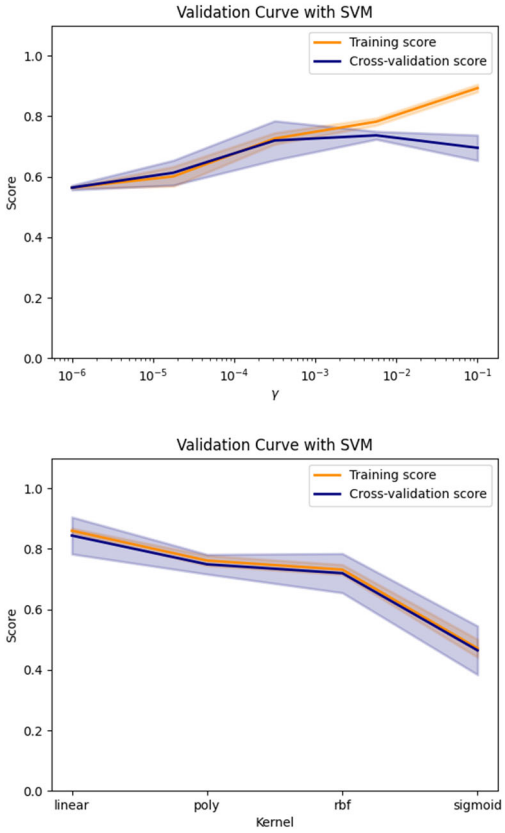

Figure 2. Validation curve of SVM for 'rbf' gamma(upper) and Kernel type(lower).

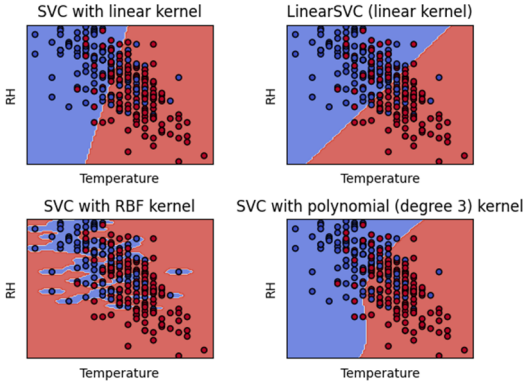

Figure 3. Result of one feature for different kernel types.

Figure 2 shows the validation curve for different kernel type and the kernel coefficient gamma for 'rbf'. For this dataset, linear kernel performs best, sigmoid performs worst. Gamma has a significant effect on the training result, when gamma is close to 0, the variance is small, when gamma is 0.001, the model start to overfit the training dataset, which means the training score increases but the validation score decreases.

2. Decision trees with some form of pruning

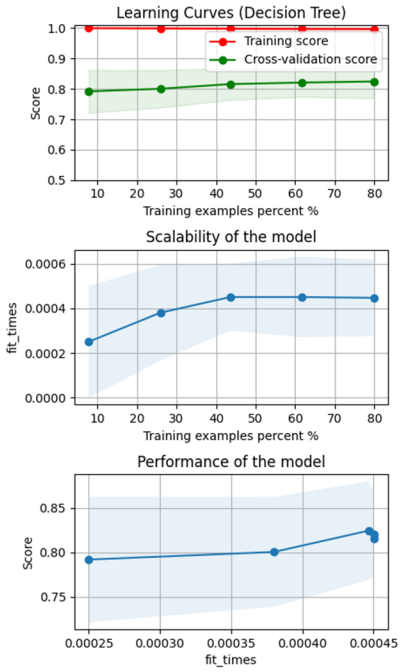

Figure 4. Learning Curves of Decision Tree.

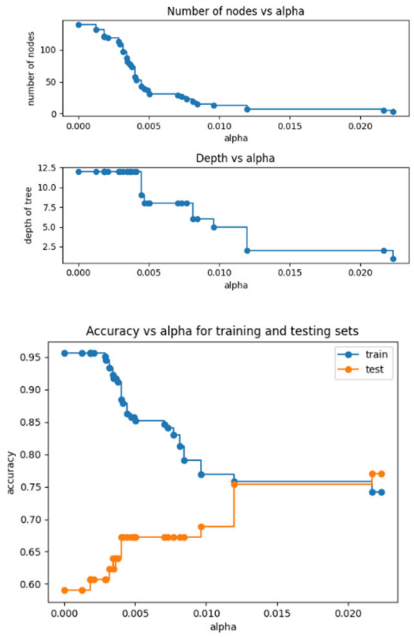

Figure 5. Effect of pruning.

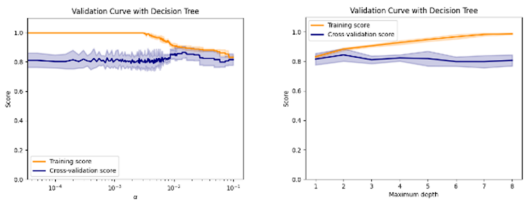

Figure 6. Validation curve of pruning(left) and maximum depth(right).

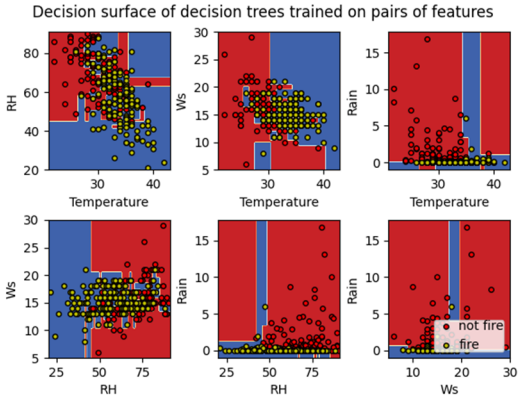

Figure 7. Results for different pairs of features.

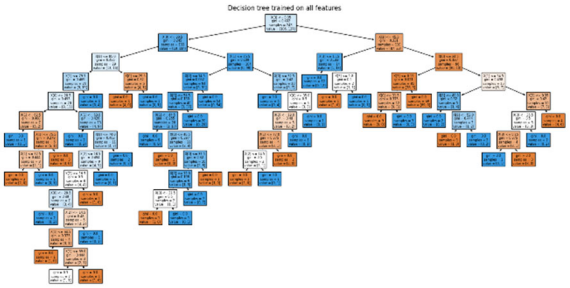

Figure 8. Decision Tree view.

No matter what the training size is, the training score of the decision tree is always 1, which means that decision tree always overfit the training dataset. However, the cross-validation score increases with the training sampling percent.

Figure 5 shows the effect of pruning. Alpha is the parameter regarding pruning, 0 means no pruning. The upper figure shows the relationship between alpha and the number of nodes and depth of tree. The lower figure shows the relationship between alpha and scores. The increase of pruning reduces the training scores but increase the testing scores, which means, pruning helps reducing overfitting. However, when alpha is bigger than 0.02, the testing score exceeds the training score, which means the model is underfitting.

Figure 6 shows the validation curve of pruning and maximum depth. The increase of pruning help reduces the overfitting and the increase of depth help increase the training score but does not help the improvement of model.

The overall performance of decision tree is about 0.8, which is acceptable. Figure 7 shows the classification figure for different pair of features and figure 8 shows the whole decision tree.

3. Neural networks

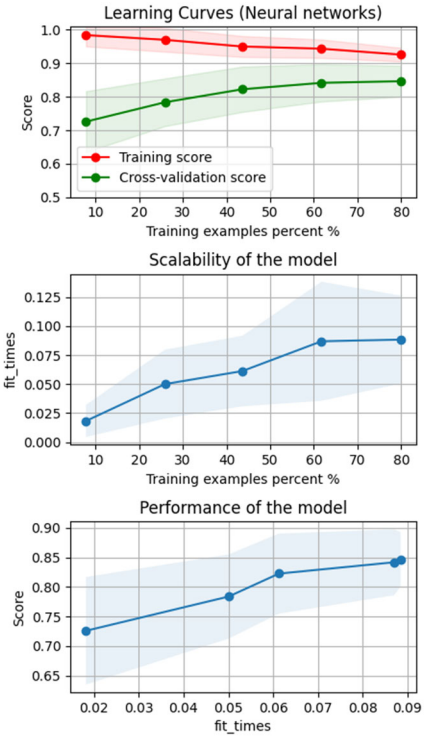

Figure 9. Learning Curves of Neural Networks.

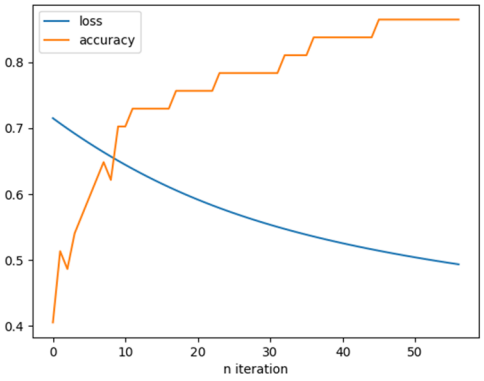

Figure 10. Loss function of Neural Networks.

The overall performance of neural networks is about 0.85 when the training dataset increased to 80%, which is good. The increase of training dataset reduces the variance and greatly increase the performance of model. In addition, as shown in figure 10, the increase of iteration increases the model accuracy.

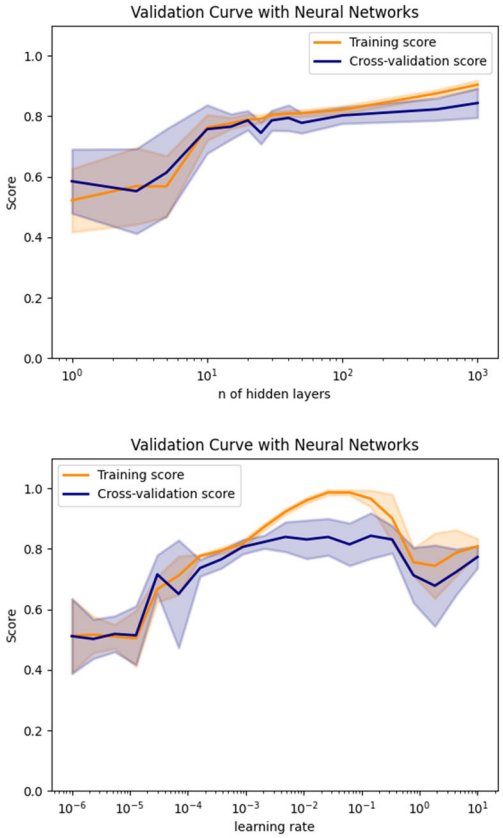

Figure 11. Validation curve of number of hidden layers(upper) and learning rate(lower).

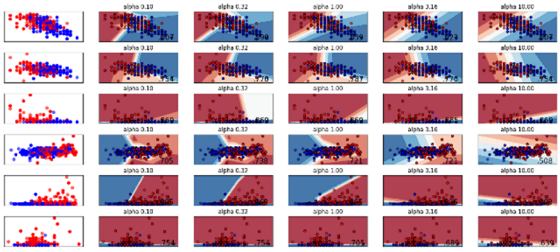

Figure 12. Result for different pairs of features and different alphas.

Figure 11 shows the performance effected by the number of hidden layers and the learning rate. The increase of hidden layers significantly increases the score and reduce the variance. The learning rate reaches the best performance at about 10^{-3}.
Alphas in figure 12 indicate the strength of the L2 regularization term. The L2 regularization term is divided by the sample size when added to the loss.

4. Ada Boost

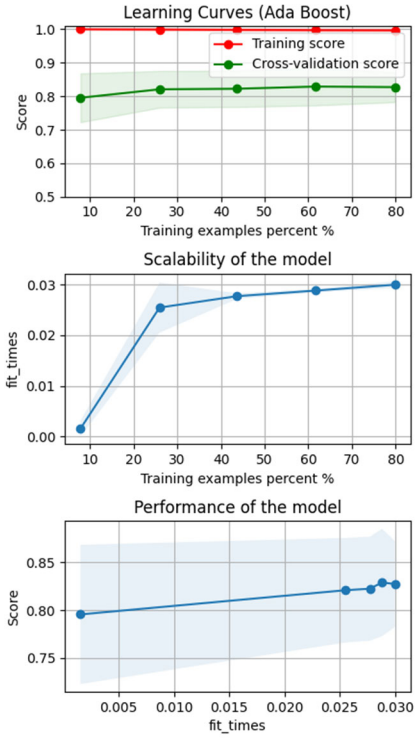

Figure 13. Learning Curves of Ada Boost.

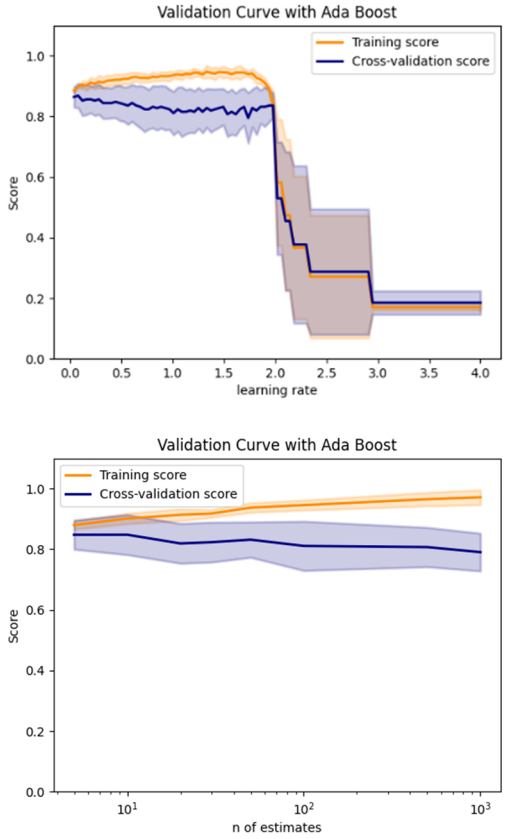

Figure 14. Validation curve of Ada Boost for learning rate(upper) and number of estimates(lower).

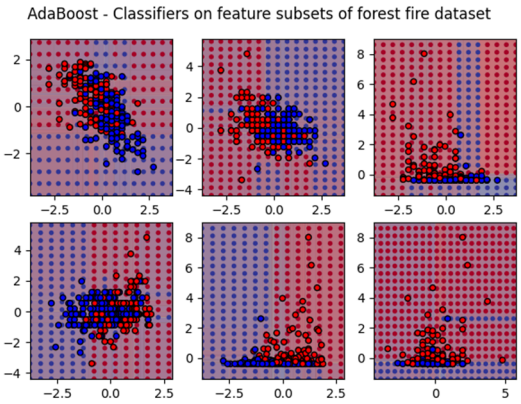

Figure 15. Results for different pairs of features.

The overall performance of Ada Boost is around 0.82. The training score is 1, which means the model overfit the training data. To save the training cost, I think 50% is the best training size.
Figure 14 shows the validation curve of learning curve and number of estimates. The increase of learning rate slightly increases the training score but decrease the cross-validation score. When the learning rate increase to 2, the score sharply decreases to 0.2, and the variance increases a lot. The best learning rate range should be between 0 and 2. The increase of number of estimates increase the training scores but decrease the cross-validation scores.

5. k-nearest neighbors

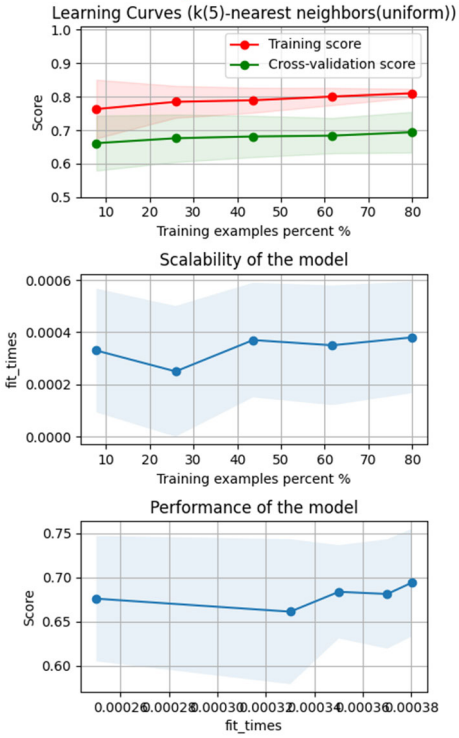

Figure 16. Learning Curves of k-Nearest Neighbors.

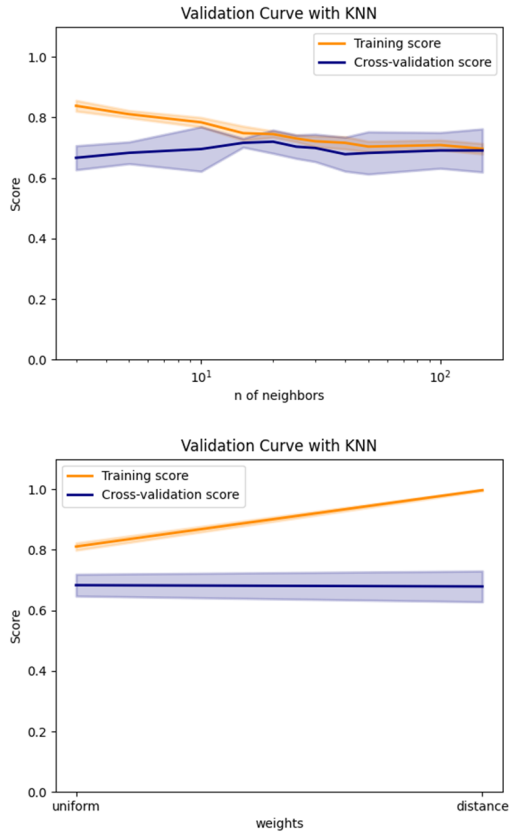

Figure 17. Validation curve of KNN for the number of neighbors (upper) and weights(lower).

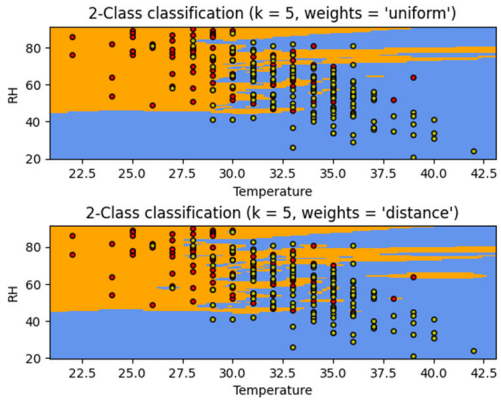

Figure 18 KNN results for RH and Temperature pair.

The overall performance of KNN is not good. According to the learning curves, training score is around 0.8 and cross-validation score is about 0.7, which means the model is overfitted. The increase of training samples does not have a great effect on the training time.

'Uniform' means all points in each neighborhood are weighted equally.

'Distance' means weight points by the inverse of their distance. In this case, closer neighbors of a query point will have a greater influence than neighbors which are further away.

'Uniform' has a lower training score but smaller cross-validation score variance, which means it is less overfitted than 'Distance'.

Brief conclusion for forest fire dataset

It is possible to predict forest fire directly based on observational data. Neural networks, Ada boost and Decision Tree have a performance score higher than 0.8, SVM and KNN have a weaker performance around 0.7.

Among the four features (Temperature, Relative Humidity, Wind speed and Rain), the most important feature is Rain. When there is rain, it is almost unlikely to have forest fire. Because rain is related with relative humidity, RH is also an important feature to predict forest fire. Rain and RH has negative effect on forest fire, Temperature and Wind speed has positive effect. Temperature is more important than Wind speed.

Test case 2: Room Occupancy Estimation Data Set
url:
https://archive.ics.uci.edu/ml/datasets/Room+Occupancy+Estimation
This dataset includes 10129 instances.
Features used in this study:

- **Average Temperature:** In degree Celsius
- **Average light:** In Lux
- **Average sound:** In Volts
- **CO2 slopes:** Slope of CO2 values taken in a sliding window

Class: number of people inside the room from **0 to 3**

Algorithms:

1. Support Vector Machines

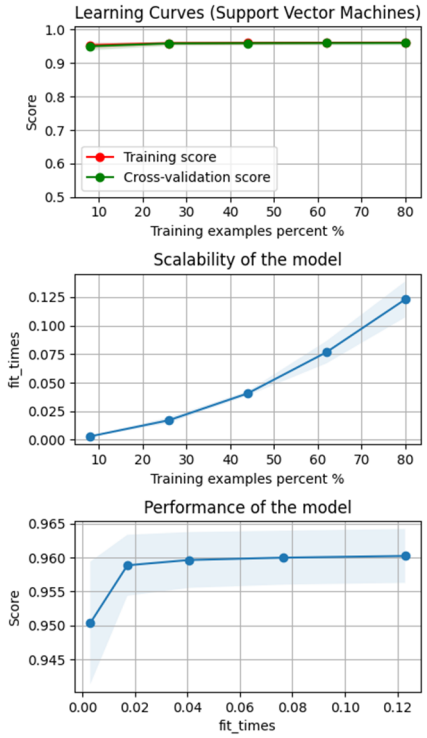

Figure 19. Learning Curves of SVM.

The average learning score of SVM for the Room Occupancy dataset is about 0.95, which is a high score. The overall training score is close to cross-validation score. The training time exponentially increases with the training size. I think 20% is the best training size because it is a large dataset.

Since the dataset used has 4 features, SVM can only train two features at a time. So each time SVM choose a feature pair and start training. Since there are 4 features, there are 6 different pairs of features, which increase the training time.

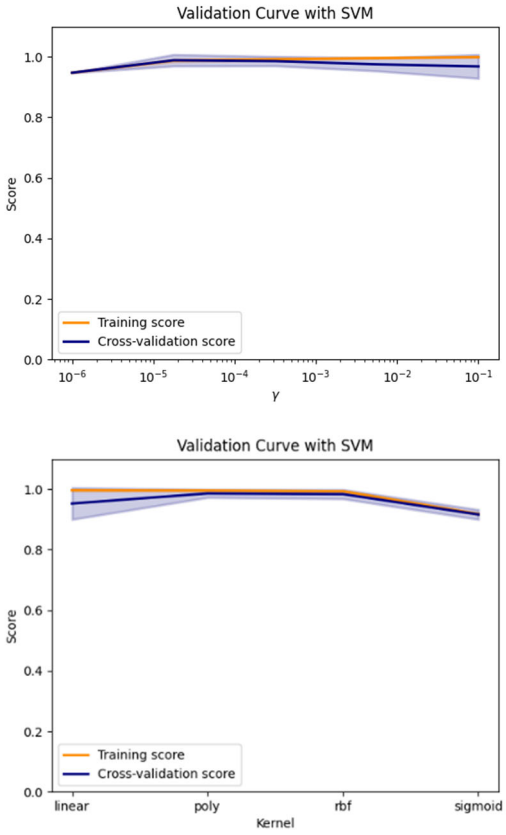

Figure 20. Validation curve of SVM for 'rbf' gamma from (upper) and Kernel type(lower).

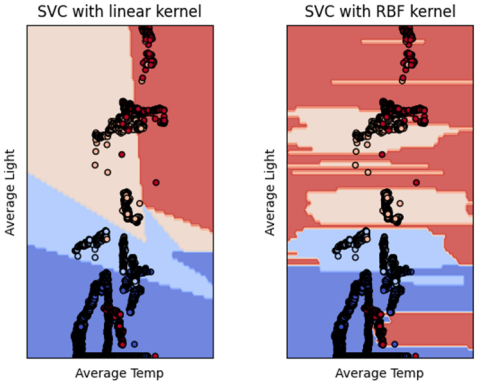

Figure 21. Result of one feature for different kernel types.

Figure 20 shows the validation curve for different kernel type and the kernel coefficient gamma for 'rbf'. For this dataset, 'poly' and 'rbf kernel' performs best. Gamma has some effect on the training result, when gamma is close to 0, the variance is small, when gamma is 0.001, the model start to overfit the training dataset, which means the training score increases but the validation score decreases.

2. Decision trees with some form of pruning

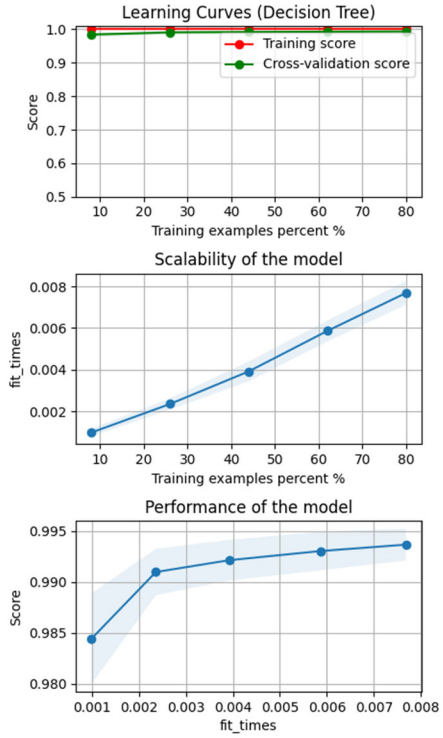

Figure 22. Learning Curves of Decision Tree.

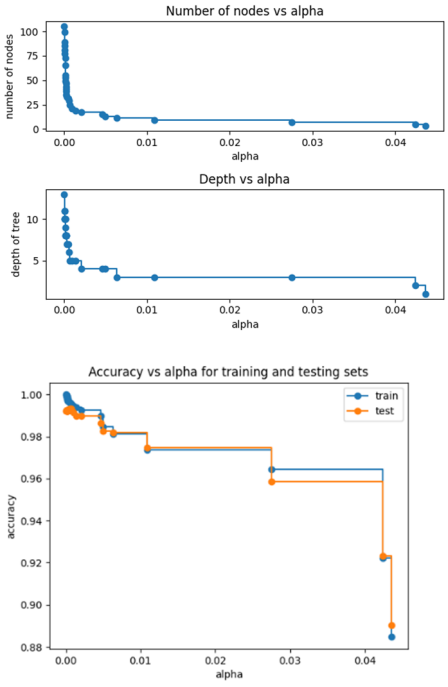

Figure 23. Effect of pruning.

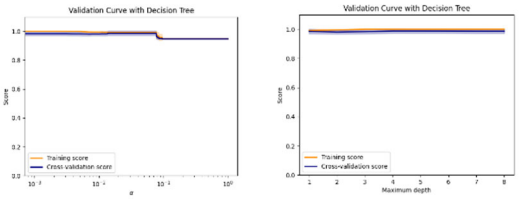

Figure 24. Validation curve of pruning(left) and maximum depth(right).

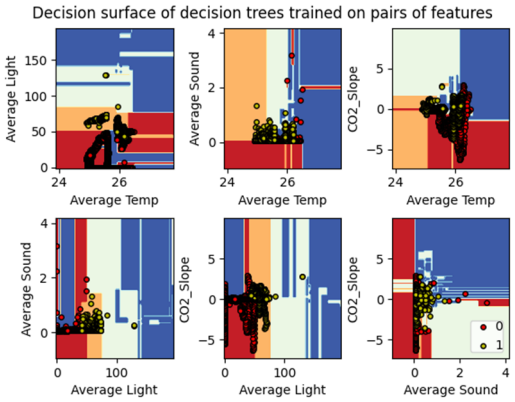

Figure 25. Results for different pairs of features.

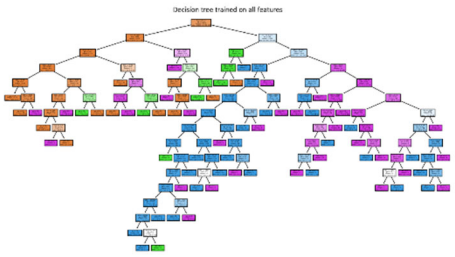

Figure 26. Decision Tree view.

No matter what the training size is, the training score of the decision tree is always 1, and the cross-validation score increases with the training sampling percent. I think about 10-20% is the best training size for the dataset.

Figure 23 shows the effect of pruning. Alpha is the parameter regarding pruning, 0 means no pruning. The upper figure shows the relationship between alpha and the number of nodes and depth of tree. The lower figure shows the relationship between alpha and scores. The increase of pruning reduces the training scores, which means pruning does not help the classification.

3. Neural networks

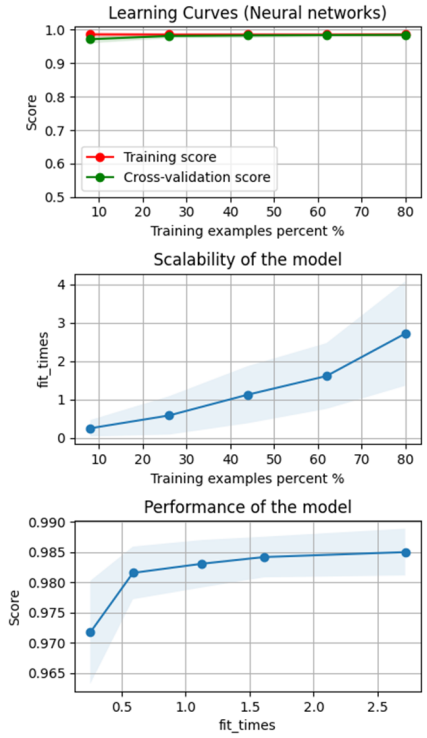

Figure 27. Learning Curves of Neural networks.

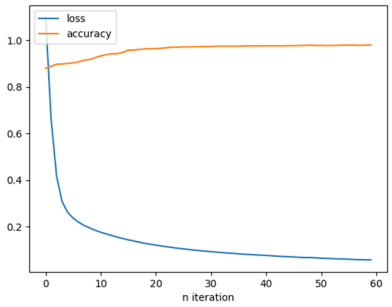

Figure 28. Loss function of Neural Networks.

The overall performance of neural networks is about 1, which is pretty good. The increase of training dataset increases the cross-validation score. In addition, as shown in figure 28, the increase of iteration also increases the model accuracy.

I think 20-30% is the best train size according to the cross validation size and training time.

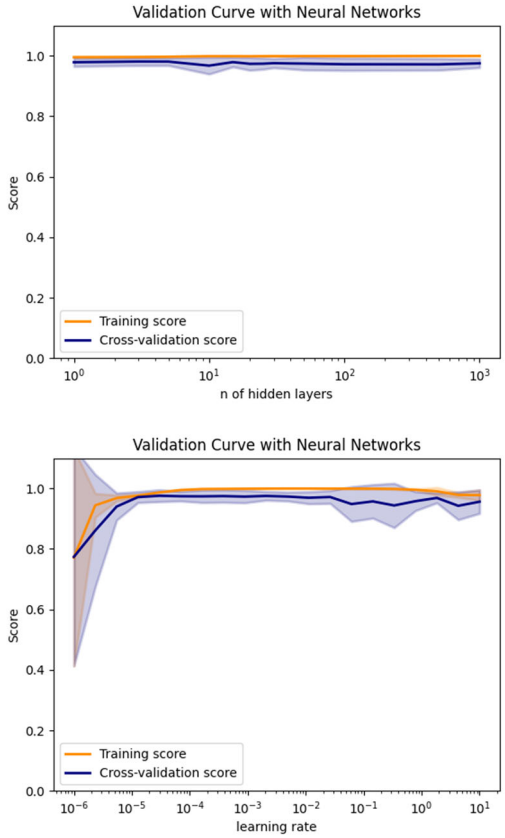

Figure 29. Validation curve of number of hidden layers(upper) and learning rate(lower).

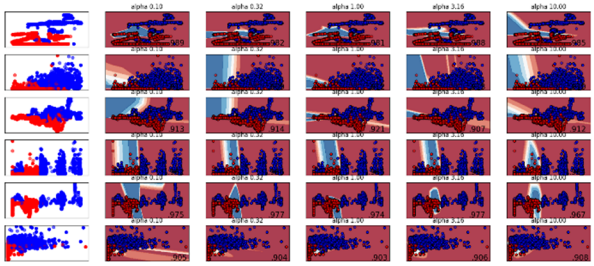

Figure 30. Result for different pairs of features and different alphas.

Figure 29 shows the performance effected by the number of hidden layers and the learning rate. The increase of hidden layers does not have a clear effect on the model performance. The learning rate reaches the best performance at about 10^{-3}.
Alphas in figure 30 indicate the strength of the L2 regularization term. The L2 regularization term is divided by the sample size when added to the loss.

4. Ada Boost

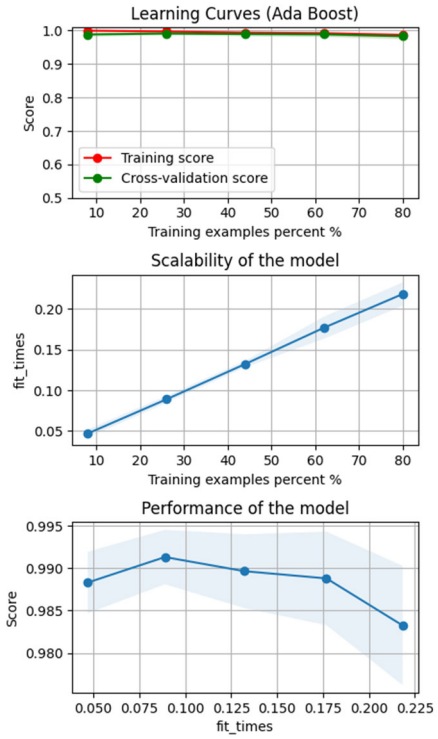

Figure 31. Learning Curves of Ada Boost.

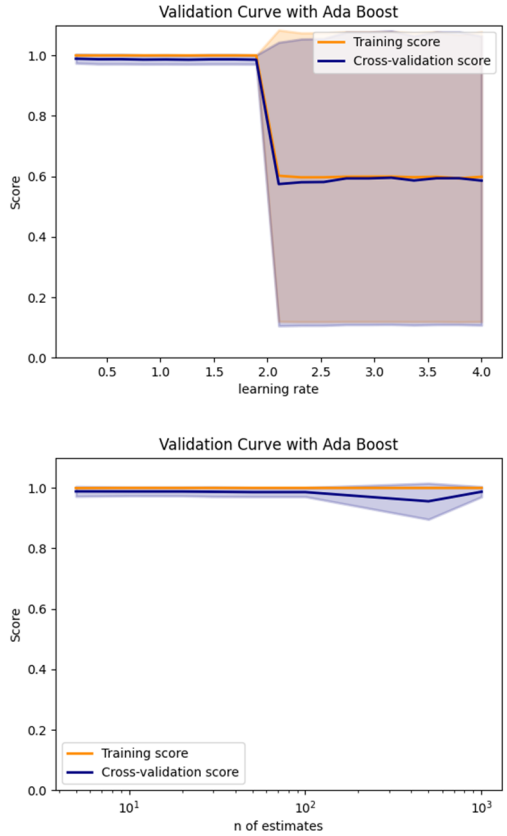

Figure 32. Validation curve of Ada Boost for learning rate(upper) and number of estimates(lower).

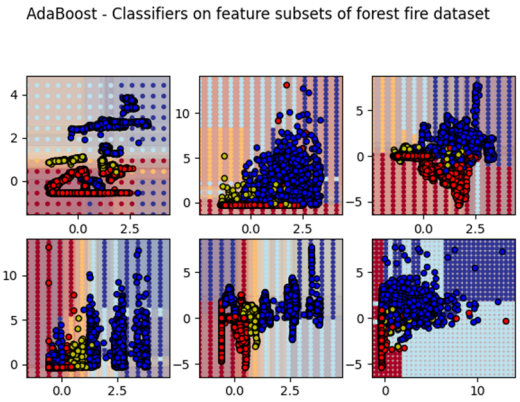

Figure 33. Results for different pairs of features.

The overall performance of Ada Boost is around 1. The training score slightly decreases with the increase of training size. The cross-validation score first slightly increase then decrease. I think 30 -40% is the best training size.

Figure 32 shows the validation curve of learning curve and number of estimates. The increase of learning rate does not affect the training score until 2. When the learning rate increased to 2, both training score and cross-validation score dropped, and the variance become extremely high. The best learning rate range should be between 0 and 2. The increase of number of estimates increase the variance of cross-validation scores.

5. k-nearest neighbors

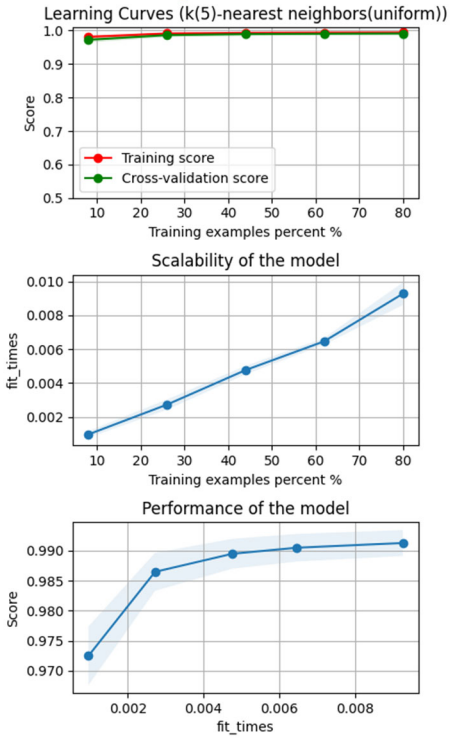

Figure 34. Learning Curves of KNN.

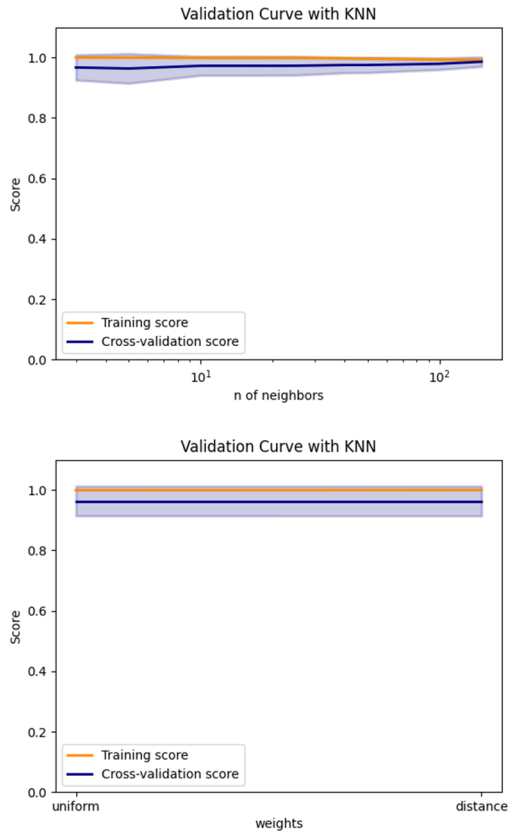

Figure 35. Validation curve of KNN for the number of neighbors (upper) and weights(lower).

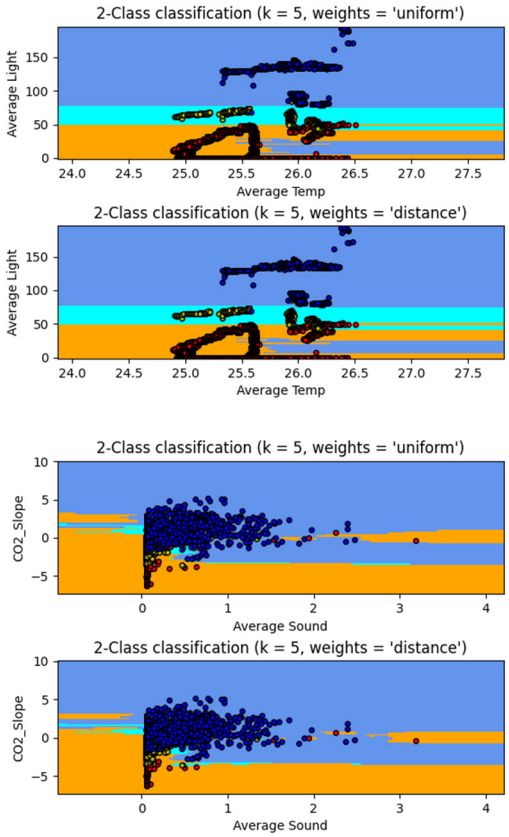

Figure 36. KNN results for Average light - Average temperature pair(upper) and CO2 – Average Sound pair(lower).

The overall performance of KNN is pretty good. According to the learning curves, both training score and cross-validation scores are close to 1. I think the best training size is around 20-30%.

The increasing number of neighbors help reduce the variance of cross-validation scores. There is no obvious difference between the 'uniform' and 'distance' weights.

Brief conclusion for Room Occupancy dataset

Because it is a large dataset with 10129 instances, all the algorithms perform well. The best algorithm should be identified by the training and predicting time. So, Decision Tree is the best algorithm in this case.

Among the four features (Temperature, Light, Sound and CO2), the clearest feature pair is the temperature and light pair. The most important feature is the average light.

Table 1 Fitting and testing time for different algorithms and datasets.

Algorithm / *Dataset (Instance)*	***Decision Tree*** *unit: s*	***Neural networks*** *(5,2) unit: s*	***Ada Boosting*** *unit: s*	***SVM*** *unit: s*	***KNN****(5 neibs, uniform) unit: s*
			Time to fit		
D1 (244)	0.0005	0.0185	0.0465	0.001	0.0005
D2 (10129)	0.003	0.9917	0.1985	0.057	0.004
			Time to test		
D1 (244)	0.0005	0.0005	0.005	0.0005	0.002
D2 (10129)	0.0005	0.002	0.019	0.1035	0.05

WHYs

- **Why did you get the results you did?**

✓ Because machine learning algorithms learns and improves upon the given dataset. It explores labeled data and identifies patterns. Based on the learned patterns, it can do prediction.

- **Compare and contrast the different algorithms.**

✓ SVM: SVM has several kernels, each kernel has different performance for different dataset. For a Boolean dataset(first), linear kernel performs better, for a multiclass dataset(second), others perform better. SVM has a better performance when the dataset is bigger.

✓ Decision Tree: The training score for decision tree is always 1, which means the model is always overfitted. For a small Boolean dataset, decision tree has a good performance.

✓ Neural networks: Neural networks have a better performance in both datasets. The disadvantage of neural networks is that it takes much longer than other algorithms to fit.

✓ Ada Boost: Ada Boost is based on Decision Tree (as weak estimate). It has a better performance than other algorithms when the dataset is small.

✓ KNN: KNN has the fastest fitting speed, and the performance is good when applied to a big dataset, which means KNN is better for classifying large dataset.

- **What sort of changes might you make to each of those algorithms to improve performance?**

✓ Testing and changing hyperparameters, like decreasing the learning rate, using different kernel. Using gird search to choose and set hyperparameters.

- **How fast were they in terms of wall clock time? Iterations?**

✓ The training speed and testing speed for each algorithm and dataset are shown in table 1. The speed of iterations is shown in figures above.

- **Would cross validation help (and if it would, why didn't you implement it?)?**

✓ Cross Validation really helps. It is implemented during the study for each algorithm.

- **How much performance was due to the problems you chose?**

✓ According to the instruction, I choose simple problems with small datasets. The performance is good, with a larger dataset, all the algorithms perform better.

- **How about the values you choose for learning rates, stopping criteria, pruning methods, and so forth (and why doesn't your analysis show results for the different values you chose? Please do look at more than one. And please make sure you understand it, it only counts if the results are meaningful)?**

✓ The choice of hyperparameters really influence the training results. See the validation figures for each algorithm.

- **Which algorithm performed best? How do you define best?**

✓ I would like to say neural networks, though it takes the longest time to fit. I think the most important thing for machine learning is accuracy, the second important thing is the time to predict. Neural network is the most accurate model among the five considering both small and large datasets and take a short time to predict.