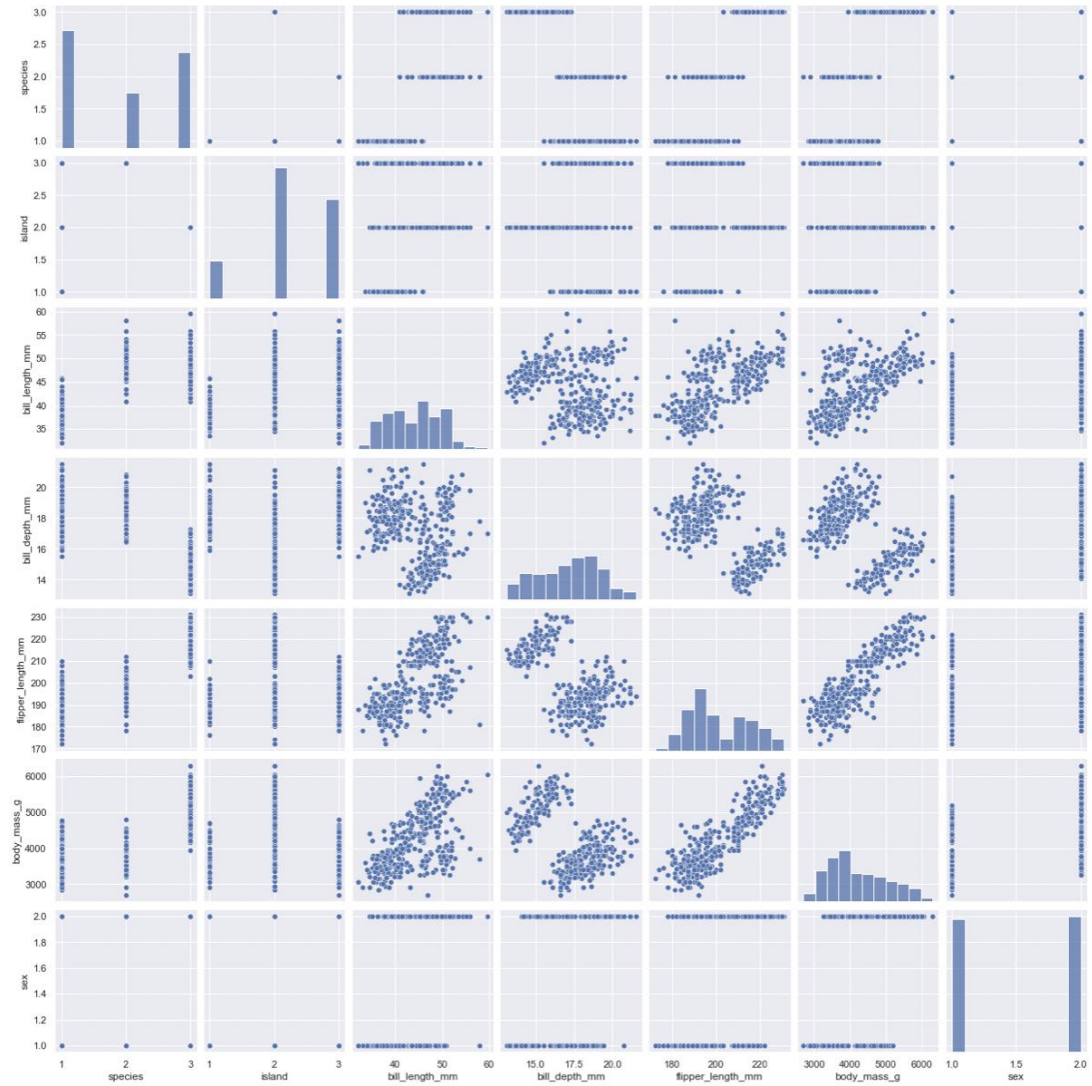
1. Decision Tree
   (1) Data preprocessing
      ① we use isna() to check whether there are “nan” or “Null”
      ② and use dropna() to drop the incomplete datas directly
   (2) Data statistics

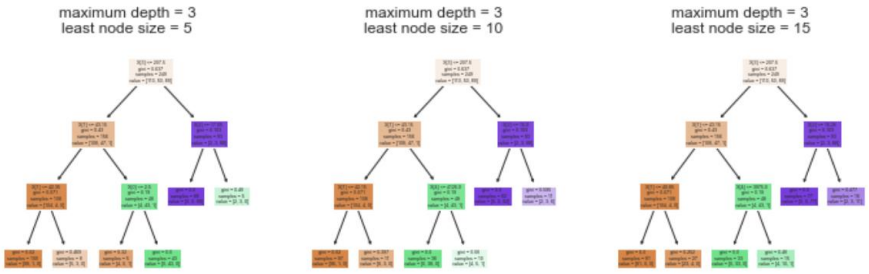
	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g
count	333.000000	333.000000	333.000000	333.000000
mean	43.992793	17.164865	200.966967	4207.057057
std	5.468668	1.969235	14.015765	805.215802
min	32.100000	13.100000	172.000000	2700.000000
25%	39.500000	15.600000	190.000000	3550.000000
50%	44.500000	17.300000	197.000000	4050.000000
75%	48.600000	18.700000	213.000000	4775.000000
max	59.600000	21.500000	231.000000	6300.000000

      ① check the statistics infomation
      ② picture some histograms to be visualized



   (3) Decision tree
      ① we try three different maximum depth and least node size, and find that the maximum depth changes, the shape of the tree may be not changed; the least node size changes, the accuracy may be not changed.

```
The maximum depth is 3, and the least node size is 5:
Train Accuracy : 0.9718875502008032
Test  Accuracy : 0.9642857142857143
The maximum depth is 3, and the least node size is 10:
Train Accuracy : 0.9437751004016064
Test  Accuracy : 0.9523809523809523
The maximum depth is 3, and the least node size is 15:
Train Accuracy : 0.9437751004016064
Test  Accuracy : 0.9523809523809523
The maximum depth is 4, and the least node size is 5:
Train Accuracy : 0.9718875502008032
Test  Accuracy : 0.9642857142857143
The maximum depth is 4, and the least node size is 10:
Train Accuracy : 0.9437751004016064
Test  Accuracy : 0.9523809523809523
The maximum depth is 4, and the least node size is 15:
Train Accuracy : 0.9437751004016064
Test  Accuracy : 0.9523809523809523
The maximum depth is 5, and the least node size is 5:
Train Accuracy : 0.9718875502008032
Test  Accuracy : 0.9642857142857143
The maximum depth is 5, and the least node size is 10:
Train Accuracy : 0.9437751004016064
Test  Accuracy : 0.9523809523809523
The maximum depth is 5, and the least node size is 15:
Train Accuracy : 0.9437751004016064
Test  Accuracy : 0.9523809523809523
```

② 

③

④ the principle of Decision Tree is that

1) find the best attribute to split by Information Gain and so on
2) split nodes until some conditions are satisfied such as maximum depth and so on

(4) Bagging of trees

① Bagging is to choose some benches of datas randomly

② To make many decision trees, and let them make decisions together, which will make the result more exact.

```
The maximum depth is 3, and the number of trees is 4:
Train Accuracy : 0.9879518072289156
Test  Accuracy : 0.9642857142857143
The maximum depth is 3, and the number of trees is 8:
Train Accuracy : 0.9879518072289156
Test  Accuracy : 0.9642857142857143
The maximum depth is 3, and the number of trees is 12:
Train Accuracy : 0.9919678714859438
Test  Accuracy : 0.9642857142857143
The maximum depth is 4, and the number of trees is 4:
Train Accuracy : 0.9879518072289156
Test  Accuracy : 0.9642857142857143
The maximum depth is 4, and the number of trees is 8:
Train Accuracy : 0.9879518072289156
Test  Accuracy : 0.9642857142857143
The maximum depth is 4, and the number of trees is 12:
Train Accuracy : 0.9959839357429718
Test  Accuracy : 0.9642857142857143
The maximum depth is 5, and the number of trees is 4:
Train Accuracy : 0.9879518072289156
Test  Accuracy : 0.9642857142857143
The maximum depth is 5, and the number of trees is 8:
Train Accuracy : 0.9879518072289156
Test  Accuracy : 0.9642857142857143
The maximum depth is 5, and the number of trees is 12:
Train Accuracy : 0.9959839357429718
Test  Accuracy : 0.9642857142857143
```

③

④ the accuracy is better than single decision tree.

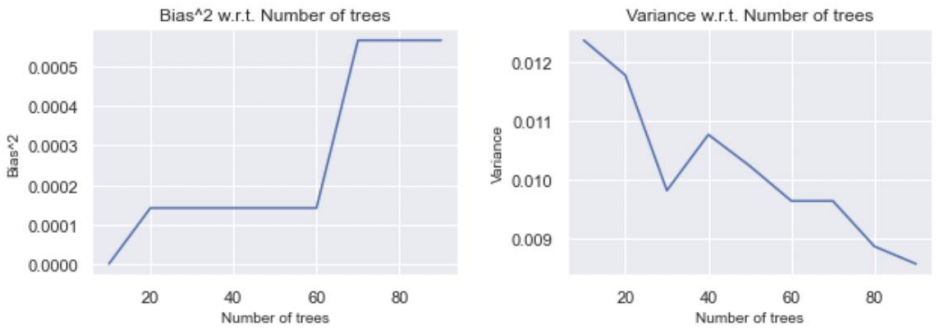
(5) Random forests

① based on bagging, we have some random choices of attributes, which make the decision trees more different between each other.

```
The number of trees is 5, and the value of m is 3:
Train Accuracy : 0.9879518072289156
Test  Accuracy : 0.9642857142857143
The number of trees is 5, and the value of m is 4:
Train Accuracy : 0.9879518072289156
Test  Accuracy : 0.9642857142857143
The number of trees is 5, and the value of m is 5:
Train Accuracy : 0.9879518072289156
Test  Accuracy : 0.9642857142857143
The number of trees is 10, and the value of m is 3:
Train Accuracy : 0.9959839357429718
Test  Accuracy : 0.9761904761904762
The number of trees is 10, and the value of m is 4:
Train Accuracy : 0.9959839357429718
Test  Accuracy : 0.9642857142857143
The number of trees is 10, and the value of m is 5:
Train Accuracy : 0.9959839357429718
Test  Accuracy : 0.9642857142857143
The number of trees is 15, and the value of m is 3:
Train Accuracy : 0.9959839357429718
Test  Accuracy : 0.9761904761904762
The number of trees is 15, and the value of m is 4:
Train Accuracy : 0.9959839357429718
Test  Accuracy : 0.9761904761904762
The number of trees is 15, and the value of m is 5:
Train Accuracy : 0.9959839357429718
Test  Accuracy : 0.9761904761904762
```

②

(6) bias^2 and variance



①

② bias^2 increases on [20,30] and [60,70], and on [20,60] and [70,100] the bias keeps constant, which indicates the Random Forests generally do not necessarily change the complexity of model.

③ variance generally decreases on [0,100] except [20,30], which indicates that ensemble model can avoid the overfitting to the fixed original data set as did in single decision trees.

2. Fashion-MNIST Recognition using sklearn

(1) Due to my slow computer, we do not use all the data, instead, we use 6000 datas to train, 1000 datas to test

(2) we use the function “MLPClassifier” of sklearn to construct a fully connected network to classify the Fashion-MNIST data.

(3) we introduce the following three hyperparameters

① hidden_layer_sizes: the number of neurons present in each hidden layer.

1) compute hidden_layer_sizes = (numHiddenNodes,)*numHiddenLayers

2) numHiddenNodes is choosed from (1,2,3)

3) numHiddenLayers is choosed from (50,200,784)

② solver: the solver employed for weight optimization.

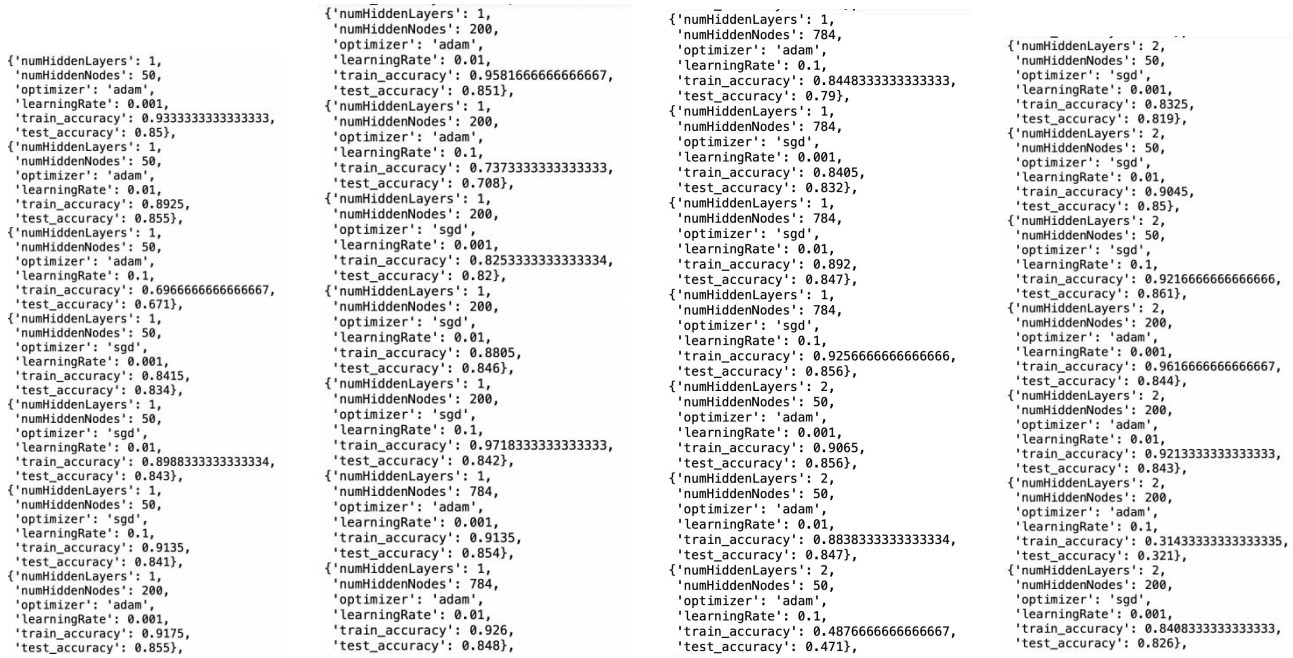
1) solver=’adam’ means the Adam optimizer is utilized to perform the gradient descent process, while solver=’sgd’ means stochastic gradient descent is employed

③ learning_rate_init: the initial learning rate employed for weight updates.

1) it means like step size, we choose it from (0.001,0.01,0.1)

(4) the result is

(5)



```
{'numHiddenLayers': 3,
 'numHiddenNodes': 784,
 'optimizer': 'adam',
 'learningRate': 0.01,
 'train_accuracy': 0.9196666666666666,
 'test_accuracy': 0.853},
{'numHiddenLayers': 3,
 'numHiddenNodes': 784,
 'optimizer': 'adam',
 'learningRate': 0.1,
 'train_accuracy': 0.8018333333333333,
 'test_accuracy': 0.77},
{'numHiddenLayers': 3,
 'numHiddenNodes': 784,
 'optimizer': 'sgd',
 'learningRate': 0.001,
 'train_accuracy': 0.8433333333333334,
 'test_accuracy': 0.833},
{'numHiddenLayers': 3,
 'numHiddenNodes': 784,
 'optimizer': 'sgd',
 'learningRate': 0.01,
 'train_accuracy': 0.885,
 'test_accuracy': 0.855},
{'numHiddenLayers': 3,
 'numHiddenNodes': 784,
 'optimizer': 'sgd',
 'learningRate': 0.1,
 'train_accuracy': 0.9221666666666667,
 'test_accuracy': 0.874}]
```


(6) I find that the performance of SGD and Adam varied depending on the learning rate. SGD outperforms Adam at high learning rates, while Adam showed slightly better performance at low learning rates.

(7) I observe that increasing model complexity by adding more hidden nodes and layers did not always improve test accuracy and could sometimes lead to overfitting.