## 一、iris 数据集特点分析

iris 数据集的中文名是安德森鸢尾花卉数据集，英文全称是 Anderson's Iris data set。iris 包含 150 个样本，对应数据集的每行数据。每行数据包含每个样本的四个特征和样本的类别信息，所以 iris 数据集是一个 150 行 5 列的二维表。

### （一）iris 数据集

表 1　iria 数据集各属性值最小值 四分位点 最大值列表

| | Min | Q1 | Q2 | Q3 | Max |
|---|---|---|---|---|---|
| 萼片长度（cm） | 4.3 | 5.1 | 5.8 | 6.4 | 7.9 |
| 萼片宽度（cm） | 2.2 | 2.8 | 3 | 3.3 | 4.4 |
| 花瓣长度（cm） | 1 | 1.6 | 4.35 | 5.1 | 6.9 |
| 花瓣宽度（cm） | 0.1 | 0.3 | 1.3 | 1.8 | 2.5 |

### （二）iris 数据集各属性箱图

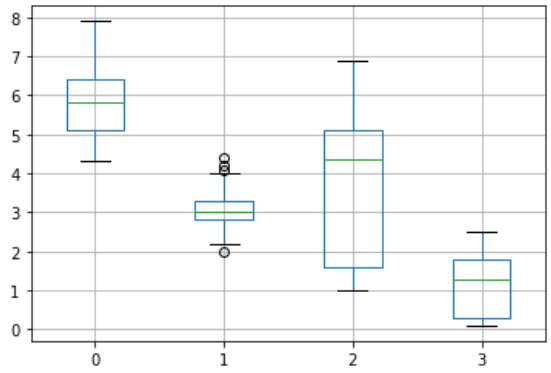


图 2 iris 数据集各属性箱图

### （三）iris 数据集直方图

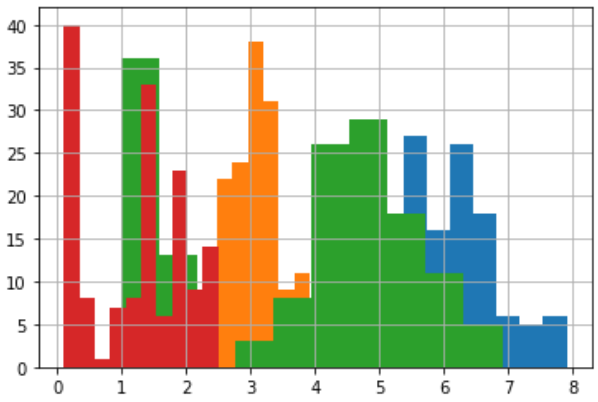


图 3 iris 数据集直方图

## （四）iris 数据集散点图

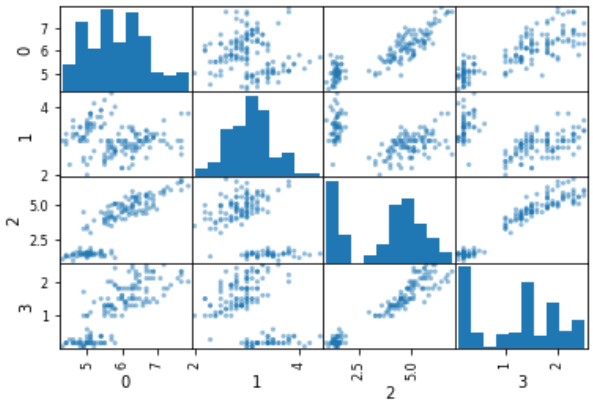


图 4 iris 数据集散点图 1

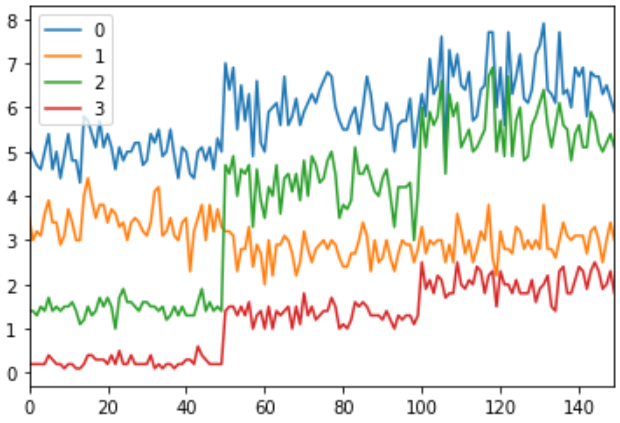


图 5 iris 数据集散点图 2

## （五）数据集特点分析

1、萼片长度（cm）该属性尾部较重，自由度较小
2、萼片宽度（cm） 该属性上明显有一些样本点是异常值 同时高属性略微成左偏态
3、花瓣长度（cm）该属性成右偏态
4、花瓣宽度（cm）该属性成右偏态
该数据样本数据点较少，各属性值分布正常。

## 二、维度灾难的直观感受

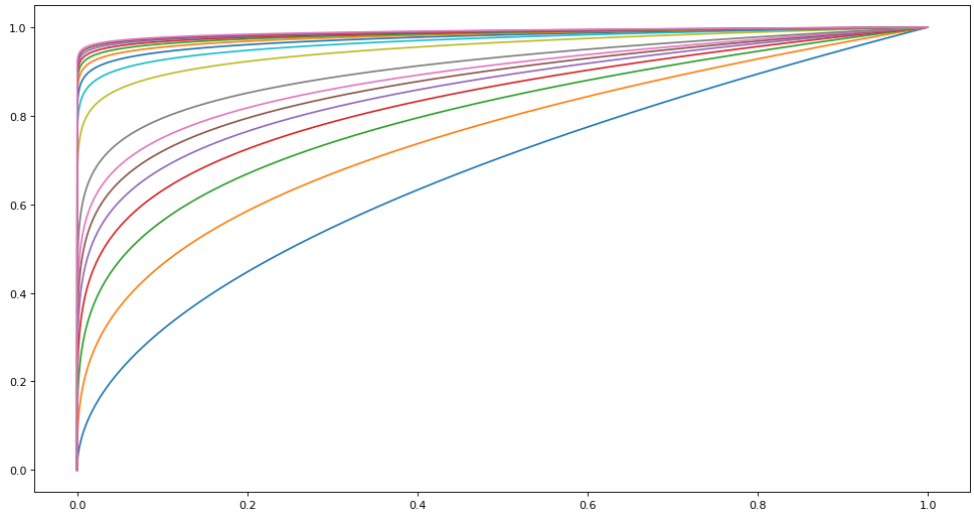


图 6 维度灾难图

相关代码均有本人书写，具见 https://github.com/TianhaoFu/-

## 附录一：iris 数据集统计量计算代码

```
#include<iostream>
#include<string>
#include<vector>
#include<fstream>
#include<sstream>
#include<typeinfo>
#include<algorithm>
#include<cstring>
#define N 150
using namespace std;
template<class outT , class inT>
outT convert(const inT &in)
{
	stringstream ss;
	outT out;
	ss << in;
	ss >> out;
	return out;
}
string array[151][5];
double array_double[151][4];
//csv 文件其实就是文本文件，每行字段用逗号分隔。
int main()
{
	ifstream inFile("iris_data.csv", ios::in);
	string lineStr;
	vector < vector    < string > > strArray;
	int i , j ;
	i = 0    ;
	char * end ;
	if ( inFile.fail() )
		cout << "读取文件失败" << endl;
	while ( getline(inFile,lineStr))
	{
//		打印整行字符串
		j = 0 ;
//		cout << lineStr << endl;
		stringstream ss(lineStr) ;
		string str;
		vector < string > lineArray ;
		while ( getline(ss,str,','))
		{
```

```
                array[i][j] = str;
                j++;
            }
            i++;
            strArray.push_back(lineArray);
        }
//      检查数据格式
////    cout<<"bibu"<<strArray[6][4]<<endl;
        for(int i = 0 ; i < 150 ; i    ++ )
        {
            for ( int j = 0 ; j < 4 ; j ++ )
            {
//                  cout << "----" <<array[i][j]<<typeid(array[i][j]).name();
                    array_double[i][j] = convert<double,string>(array[i][j]);
//                  cout << "++++" <<array_double[i][j]<<typeid(array_double[i][j]).name();
            }
            cout << endl;
        }

        double min[5],max[5],media[5],q1[5],q3[5];
        double a[N+1];
        for(int j = 0 ; j <4    ; j ++)
        {
            memset(a,0,sizeof(a));
            for(int i = 0 ; i < 150 ; i ++ )
            {
                a[i+1] =array_double[i][j] ;
            }
            sort(a+1,a+N);
//          for(i=1;i<150;i++)
//                  cout<<"Ddaa"<<a[i]<<endl;
            min[j] = a[1];
            max[j] = a[N];
            media[j] = (a[N/2] + a[N/2+1])/2;
            int loc1 = (N/2)/2 + 1;
            //不是整数
            int loc3 = (N/2+1) + (( N - (N/2+1) + 1) / 2);
            //不是整数
//          长度 ： 末-初+1
            //中位数 首 + 长度/2 （此题中非整数）
            q1[j] = a[loc1] ;
            q3[j] = a[loc3] ;
        }
        ofstream outFile;
```

```
    outFile.open("iris_data_processed.csv",ios::out);
    outFile<<"attribute_name"<<','<<"min"<<','<<"q1"<<','<<"mean"<<','<<"q3"<<','<<"max"<
<endl;
    outFile << "萼片长度(cm)"<<','<<min[0]<<','<<q1[0]<<','<<media[0]<<','<<q3[0]<<','<<max[0]<<endl;
    outFile << "萼片宽度(cm)"<<','<<min[1]<<','<<q1[1]<<','<<media[1]<<','<<q3[1]<<','<<max[1]<<endl;
    outFile << "花瓣长度(cm)"<<','<<min[2]<<','<<q1[2]<<','<<media[2]<<','<<q3[2]<<','<<max[2]<<endl;
    outFile << "花瓣宽度(cm)"<<','<<min[3]<<','<<q1[3]<<','<<media[3]<<','<<q3[3]<<','<<max[3]<<endl;
    outFile.close();
    return 0;
}
```

## 附录二：iris 数据集箱图绘制代码

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

df = pd.read_csv(r'C:\Users\Tianh\Desktop\iris_data.csv',header=None)
#将 4 列属性提取为 4 列列表便于后续处理
df1 = df.iloc[0:,0]
df2 = df.iloc[0:,1]
df3 = df.iloc[1:,2]
df4 = df.iloc[1:,3]

df.boxplot()
plt.show()
```

## 附录三：iris 数据集直方图绘制代码

```
df1.hist()
df2.hist()
df3.hist()
df4.hist()
```

## 附录四：iris 数据集散点图绘制代码

```
df.plot()
```

## 附录五：维度灾难示意图绘制代码

```
import matplotlib.pyplot as plt
import numpy as np
```

```
def f1(x):
    return pow(x,1/2)
def f2(x):
    return pow(x,1/3)
def f3(x):
    return pow(x,1/4)
def f4(x):
    return pow(x,1/5)
def f5(x):
    return pow(x,1/6)
def f6(x):
    return pow(x,1/7)
def f7(x):
    return pow(x,1/8)
def f11(x):
    return pow(x,1/10)
def f21(x):
    return pow(x,1/20)
def f31(x):
    return pow(x,1/30)
def f41(x):
    return pow(x,1/40)
def f51(x):
    return pow(x,1/50)
def f61(x):
    return pow(x,1/60)
def f71(x):
    return pow(x,1/70)
def f81(x):
    return pow(x,1/80)
def f91(x):
    return pow(x,1/90)
def f101(x):
    return pow(x,1/100)
g1 = np.frompyfunc(f1,1,1)
g2 = np.frompyfunc(f2,1,1)
g3 = np.frompyfunc(f3,1,1)
g4 = np.frompyfunc(f4,1,1)
g5 = np.frompyfunc(f5,1,1)
g6 = np.frompyfunc(f6,1,1)
g7 = np.frompyfunc(f7,1,1)
g11 = np.frompyfunc(f11,1,1)
g21 = np.frompyfunc(f21,1,1)
g31 = np.frompyfunc(f31,1,1)
```

```
g41 = np.frompyfunc(f41,1,1)
g51 = np.frompyfunc(f51,1,1)
g61 = np.frompyfunc(f61,1,1)
g71 = np.frompyfunc(f71,1,1)
g81 = np.frompyfunc(f81,1,1)
g91 = np.frompyfunc(f91,1,1)
g101 = np.frompyfunc(f101,1,1)

fig = plt.figure(num=1,figsize=(15,8),dpi=80)
a = np.arange(0,1,0.0001)
plt.plot(np.arange(0,1,0.0001),g1(a))
plt.plot(np.arange(0,1,0.0001),g2(a))
plt.plot(np.arange(0,1,0.0001),g3(a))
plt.plot(np.arange(0,1,0.0001),g4(a))
plt.plot(np.arange(0,1,0.0001),g5(a))
plt.plot(np.arange(0,1,0.0001),g6(a))
plt.plot(np.arange(0,1,0.0001),g7(a))
plt.plot(np.arange(0,1,0.0001),g11(a))
plt.plot(np.arange(0,1,0.0001),g21(a))
plt.plot(np.arange(0,1,0.0001),g31(a))
plt.plot(np.arange(0,1,0.0001),g41(a))
plt.plot(np.arange(0,1,0.0001),g51(a))
plt.plot(np.arange(0,1,0.0001),g61(a))
plt.plot(np.arange(0,1,0.0001),g71(a))
plt.plot(np.arange(0,1,0.0001),g81(a))
plt.plot(np.arange(0,1,0.0001),g91(a))
plt.plot(np.arange(0,1,0.0001),g101(a))
plt.show()
```