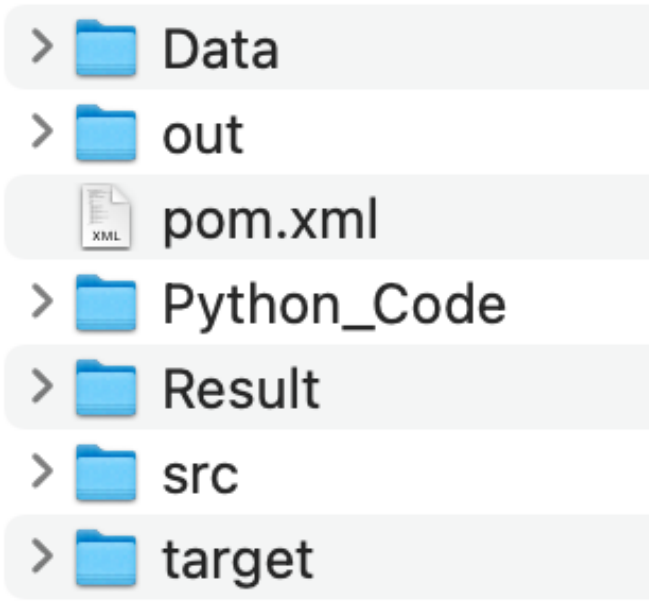
FBDP 作业7

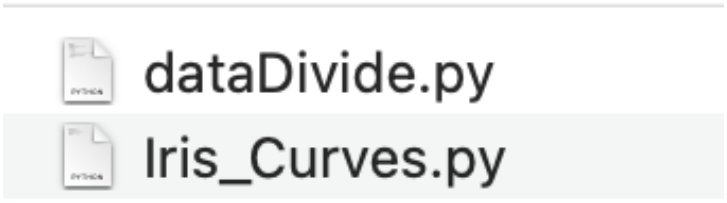
191840265 吴偲羿

0.文件夹目录



Data储存输入文件 为已经分好测试集训练集的iris数据集子集

Python_Code内：



dataDivide为通过sklearn中train_test_split库函数划分训练集测试集代码

Iris_Curves为绘制Iris数据集相关图表的函数

Result储存输出结果

src为java源码

2.设计思路

mapreduce实现KNN算法现成思路挺多的 我参考了https://blog.csdn.net/qq_39009237/article/details/86346762的算法实现

总体思路为：

在map中完成测试集合本地训练集的距离计算，reduce端完成排序和挑选。但是由于数据是巨量的，在reduce中完成排序是不实际的。通过自定义数据类型Rose(Iris鸢尾花 Rose玫瑰 挺好)，利用shuffle过程完成自动的排序。实验本质上是一个top N问题，在选择top N的算法上可以压缩到O(N)的算法复杂度，要充分利用map端的combiner来减少mapreduce的网络通信量。具体做法是对于每一个测试数据在本地只发送前k个数据，即将本地距离最近的k个发送出去。
map中完成距离的计算和发送特定的键值对，使其自动排序

combiner是在排好序后，在本地进行对每个id值发送k条数据的限制。有效地较少数据量。

Reduce完成的是将从各个map中接受到的数据和combiner类似，对每个测试数据id只处理前k个，因为使用了自定义数据类型，数据会按照距离排好序，相同id会聚集在一起。所以在处理时会非常方便。因为此时处理的顺序是按原数据行处理的，同时使用了Verify文件进行验证，最终计算出正确率

3.划分数据

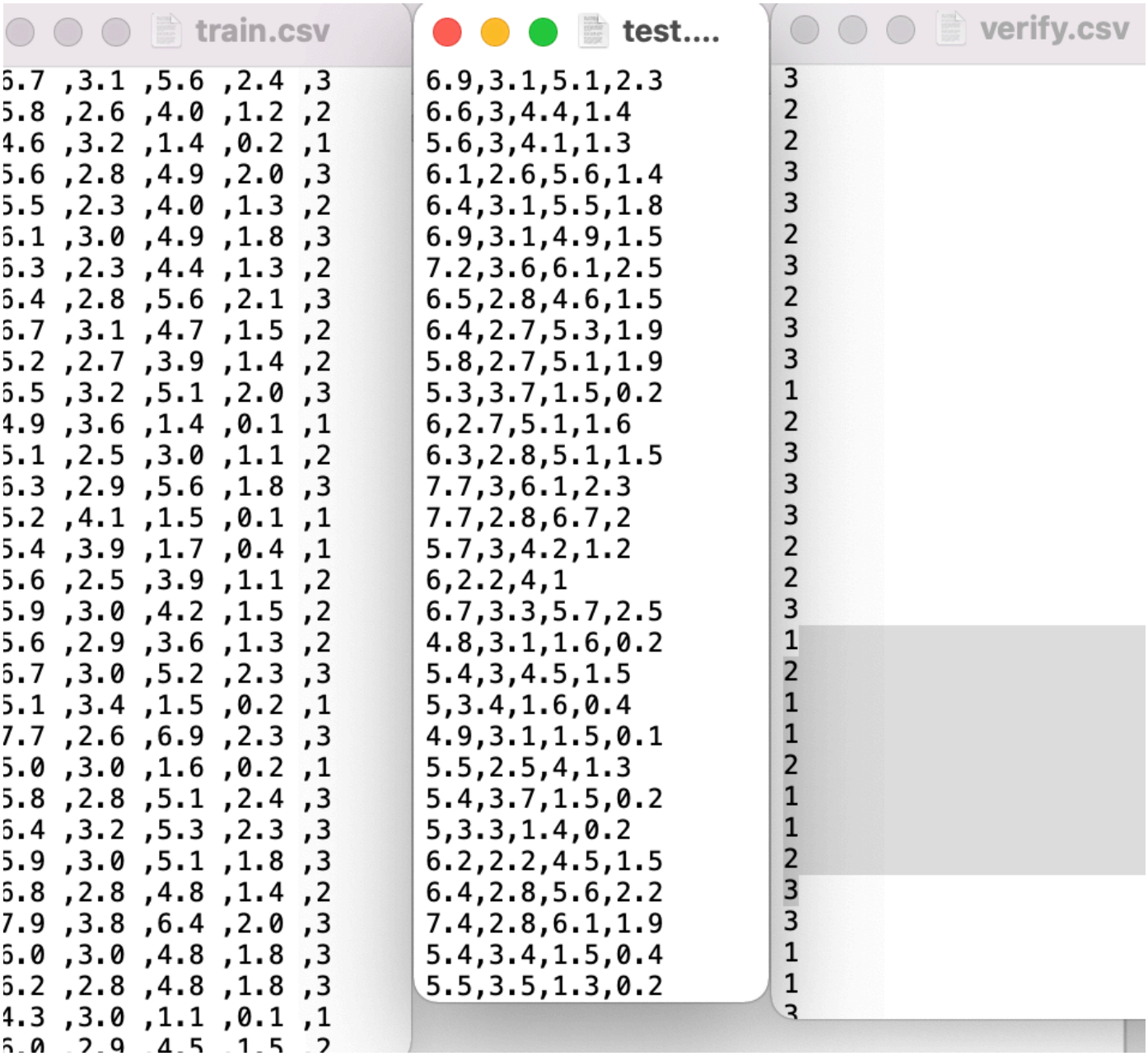
我们采用了python sklearn库中train_test_split函数 对iris数据集进行划分 为了处理数据方便 我们将鸢尾花的label抽象为 1.2.3 如下：

6.7 ,	3.1 ,	5.6 ,	2.4 ,	3
5.8 ,	2.6 ,	4.0 ,	1.2 ,	2
4.6 ,	3.2 ,	1.4 ,	0.2 ,	1
5.6 ,	2.8 ,	4.9 ,	2.0 ,	3
5.5 ,	2.3 ,	4.0 ,	1.3 ,	2
6.1 ,	3.0 ,	4.9 ,	1.8 ,	3
6.3 ,	2.3 ,	4.4 ,	1.3 ,	2
6.4 ,	2.8 ,	5.6 ,	2.1 ,	3
6.7 ,	3.1 ,	4.7 ,	1.5 ,	2
5.2 ,	2.7 ,	3.9 ,	1.4 ,	2
6.5 ,	3.2 ,	5.1 ,	2.0 ,	3
4.9 ,	3.6 ,	1.4 ,	0.1 ,	1

5.1 ,	2.5 ,	3.0 ,	1.1 ,	2
6.3 ,	2.9 ,	5.6 ,	1.8 ,	3
5.2 ,	4.1 ,	1.5 ,	0.1 ,	1
5.4 ,	3.9 ,	1.7 ,	0.4 ,	1
5.6 ,	2.5 ,	3.9 ,	1.1 ,	2
5.9 ,	3.0 ,	4.2 ,	1.5 ,	2
5.6 ,	2.9 ,	3.6 ,	1.3 ,	2
6.7 ,	3.0 ,	5.2 ,	2.3 ,	3
5.1 ,	3.4 ,	1.5 ,	0.2 ,	1
7.7 ,	2.6 ,	6.9 ,	2.3 ,	3
5.0 ,	3.0 ,	1.6 ,	0.2 ,	1
5.8 ,	2.8 ,	5.1 ,	2.4 ,	3

第五列即为label。

由于训练集与测试集划分的随机性，我们重复执行dataDivide.py 5次，得到五组数据，并划分成train.csv,test.csv与verify.csv 供程序读取：



4.程序执行及结果

对于每一组数据以及固定的k值 我们执行一次程序 获取结果 运行截图及结果如下（以k=5为例）（正好night_shift mac的ui变成了深色模式）

```
quinton_541@MacBook-Air-de-541 bin % ./hadoop jar Iris.jar input/train.csv output_5.0 input/test.csv input/verify.csv
```

```
        File System Counters
                FILE: Number of bytes read=139153874
                FILE: Number of bytes written=141532603
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=6750
                HDFS: Number of bytes written=985
                HDFS: Number of read operations=65
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
                HDFS: Number of bytes read erasure-coded=0
        Map-Reduce Framework
                Map input records=105
                Map output records=4725
                Map output bytes=70875
                Map output materialized bytes=3831
                Input split bytes=121
                Combine input records=4725
                Combine output records=225
                Reduce input groups=221
                Reduce shuffle bytes=3831
                Reduce input records=225
                Reduce output records=49
                Spilled Records=450
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=8
                Total committed heap usage (bytes)=1306525696
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=2520
        File Output Format Counters
                Bytes Written=985
```

结果如下：

```
Sepal.Length,Sepal.Width,Petal.Length,Petal.Width,Predicted Value,True Value
```

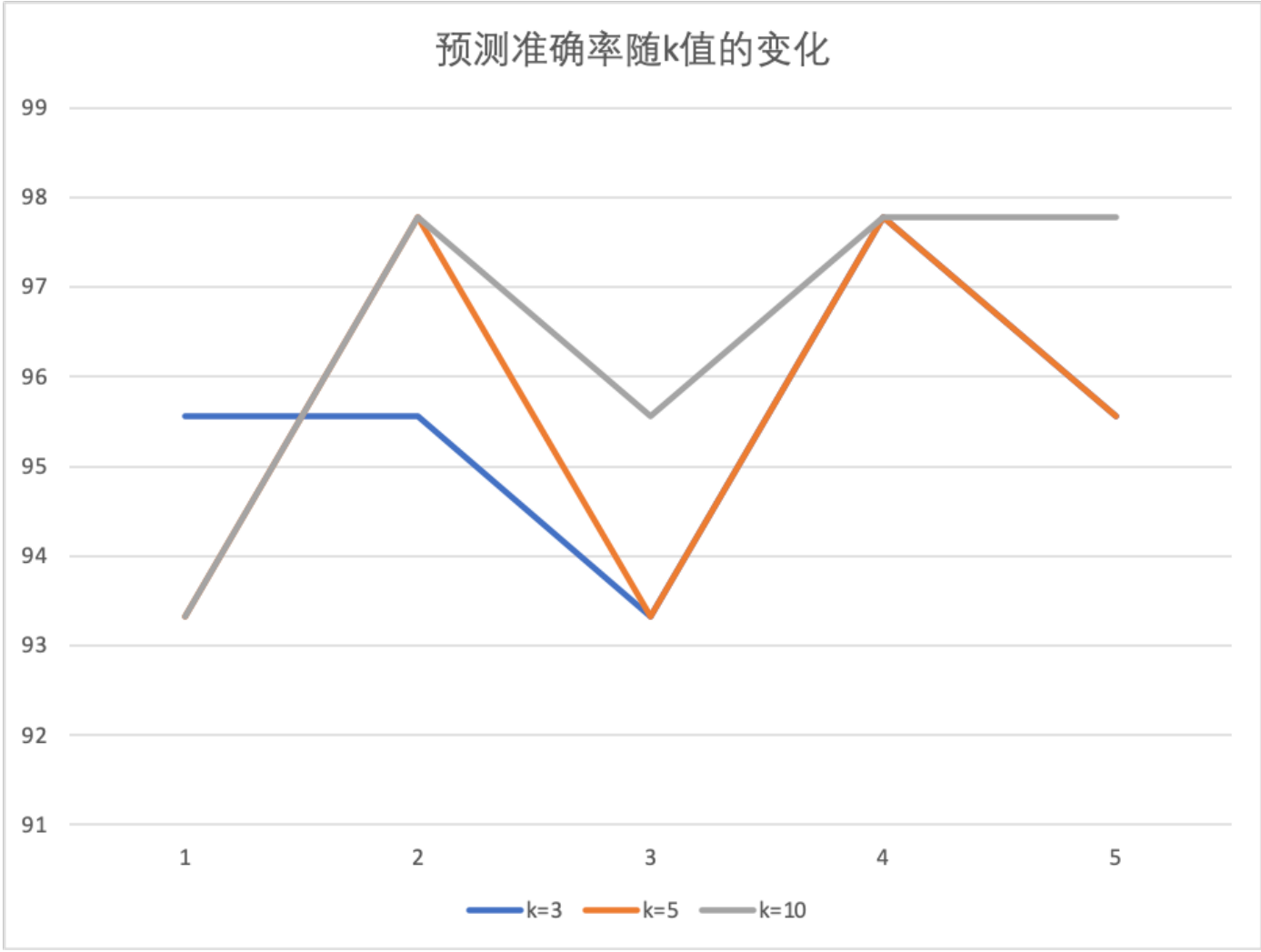
```
6.9,3.1,5.1,2.3,3,3
6.6,3,4.4,1.4,2,2
5.6,3,4.1,1.3,2,2
6.1,2.6,5.6,1.4,3,3
6.4,3.1,5.5,1.8,3,3
6.9,3.1,4.9,1.5,2,2
7.2,3.6,6.1,2.5,3,3
6.5,2.8,4.6,1.5,2,2
6.4,2.7,5.3,1.9,3,3
5.8,2.7,5.1,1.9,3,3
5.3,3.7,1.5,0.2,1,1
6,2.7,5.1,1.6,3,2
6.3,2.8,5.1,1.5,3,3
7.7,3,6.1,2.3,3,3
7.7,2.8,6.7,2,3,3
5.7,3,4.2,1.2,2,2
6,2.2,4,1,2,2
6.7,3.3,5.7,2.5,3,3
4.8,3.1,1.6,0.2,1,1
5.4,3,4.5,1.5,2,2
5,3.4,1.6,0.4,1,1
4.9,3.1,1.5,0.1,1,1
5.5,2.5,4,1.3,2,2
5.4,3.7,1.5,0.2,1,1
5,3.3,1.4,0.2,1,1
6.2,2.2,4.5,1.5,2,2
6.4,2.8,5.6,2.2,3,3
7.4,2.8,6.1,1.9,3,3
5.4,3.4,1.5,0.4,1,1
5.5,3.5,1.3,0.2,1,1
6.7,2.5,5.8,1.8,3,3
5.2,3.4,1.4,0.2,1,1
4.6,3.6,1,0.2,1,1
4.6,3.1,1.5,0.2,1,1
7.3,2.9,6.3,1.8,3,3
6.7,3.3,5.7,2.1,3,3
6.7,3,5,1.7,3,2
6.5,3,5.2,2,3,3
5.7,4.4,1.5,0.4,1,1
4.8,3,1.4,0.1,1,1
5.7,2.5,5,2,3,3
4.7,3.2,1.6,0.2,1,1
5.1,3.7,1.5,0.4,1,1
4.4,3.2,1.3,0.2,1,1
5.1,3.8,1.6,0.2,1,1
Distance calculation method:osjl
k:5
accuracy:93.333333333333333%
```

同时，在程序中更改默认k值为3，10，对5组数据分别执行程序，获得输出至不同文件，此时hdfs文件系统内容如下：

☐	drwxr-xr-x	quinton_541	supergroup	0 B	Nov 14 15:44	0	0 B	input
☐	drwxr-xr-x	quinton_541	supergroup	0 B	Nov 14 16:14	0	0 B	output_10.0
☐	drwxr-xr-x	quinton_541	supergroup	0 B	Nov 14 16:15	0	0 B	output_10.1
☐	drwxr-xr-x	quinton_541	supergroup	0 B	Nov 14 16:16	0	0 B	output_10.2
☐	drwxr-xr-x	quinton_541	supergroup	0 B	Nov 14 16:17	0	0 B	output_10.3
☐	drwxr-xr-x	quinton_541	supergroup	0 B	Nov 14 16:18	0	0 B	output_10.4
☐	drwxr-xr-x	quinton_541	supergroup	0 B	Nov 14 16:00	0	0 B	output_3.0
☐	drwxr-xr-x	quinton_541	supergroup	0 B	Nov 14 16:06	0	0 B	output_3.1
☐	drwxr-xr-x	quinton_541	supergroup	0 B	Nov 14 16:07	0	0 B	output_3.2
☐	drwxr-xr-x	quinton_541	supergroup	0 B	Nov 14 16:08	0	0 B	output_3.3
☐	drwxr-xr-x	quinton_541	supergroup	0 B	Nov 14 16:09	0	0 B	output_3.4
☐	drwxr-xr-x	quinton_541	supergroup	0 B	Nov 14 15:45	0	0 B	output_5.0
☐	drwxr-xr-x	quinton_541	supergroup	0 B	Nov 14 15:47	0	0 B	output_5.1
☐	drwxr-xr-x	quinton_541	supergroup	0 B	Nov 14 15:47	0	0 B	output_5.2
☐	drwxr-xr-x	quinton_541	supergroup	0 B	Nov 14 15:48	0	0 B	output_5.3
☐	drwxr-xr-x	quinton_541	supergroup	0 B	Nov 14 15:48	0	0 B	output_5.4

获得不同k值下精确度，做表作图如下：

	k=3	**k=5**	**k=10**
1	95.56	93.33	93.33
2	95.56	97.78	97.78
3	93.33	93.33	95.56
4	97.78	97.78	97.78
5	95.56	95.56	97.78



我们可以看出，也许数据集较小，预测准确率随k值变化并不明显，查阅相关资料得知：在 KNN算法中，k的取值一一般不不超过训练样本数的平方方根。实际应用用中，可以采用用 交叉验证法 来选择最优的 k 值。

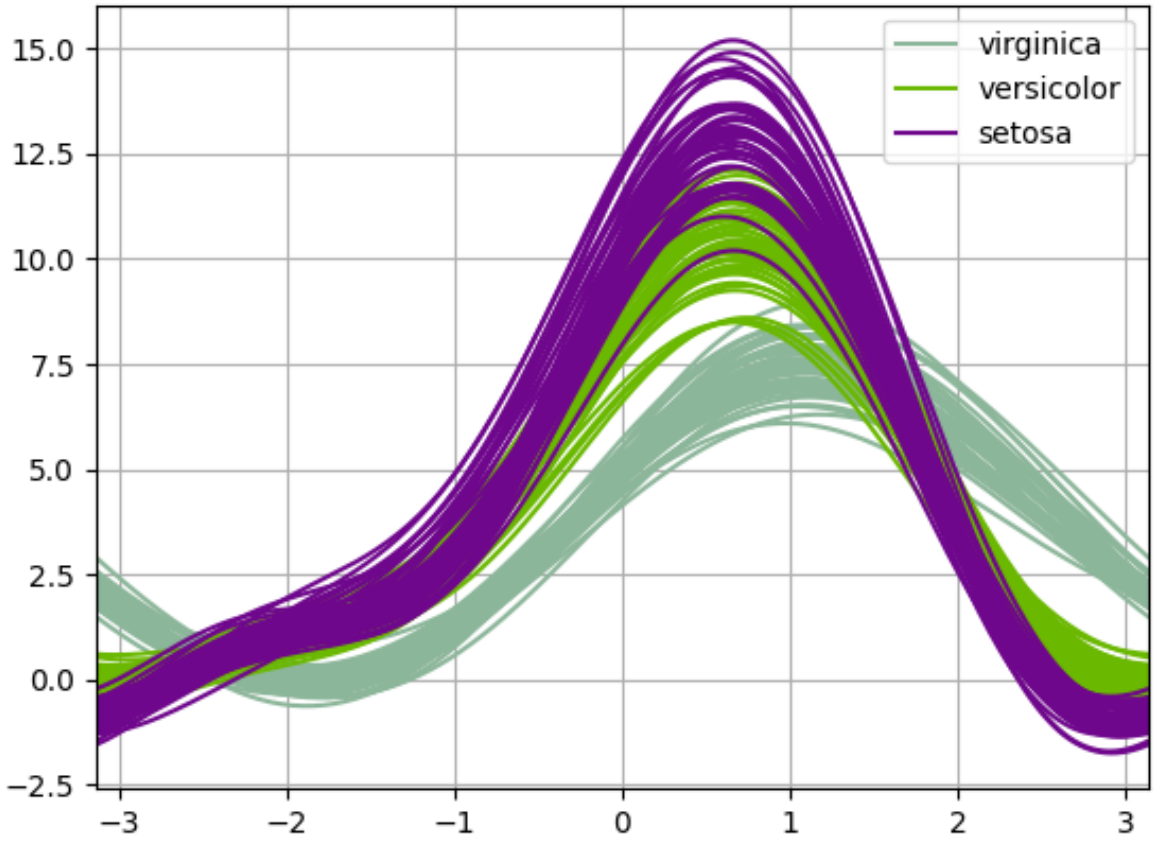
5.有关Iris鸢尾花数据集的一些作图与结论

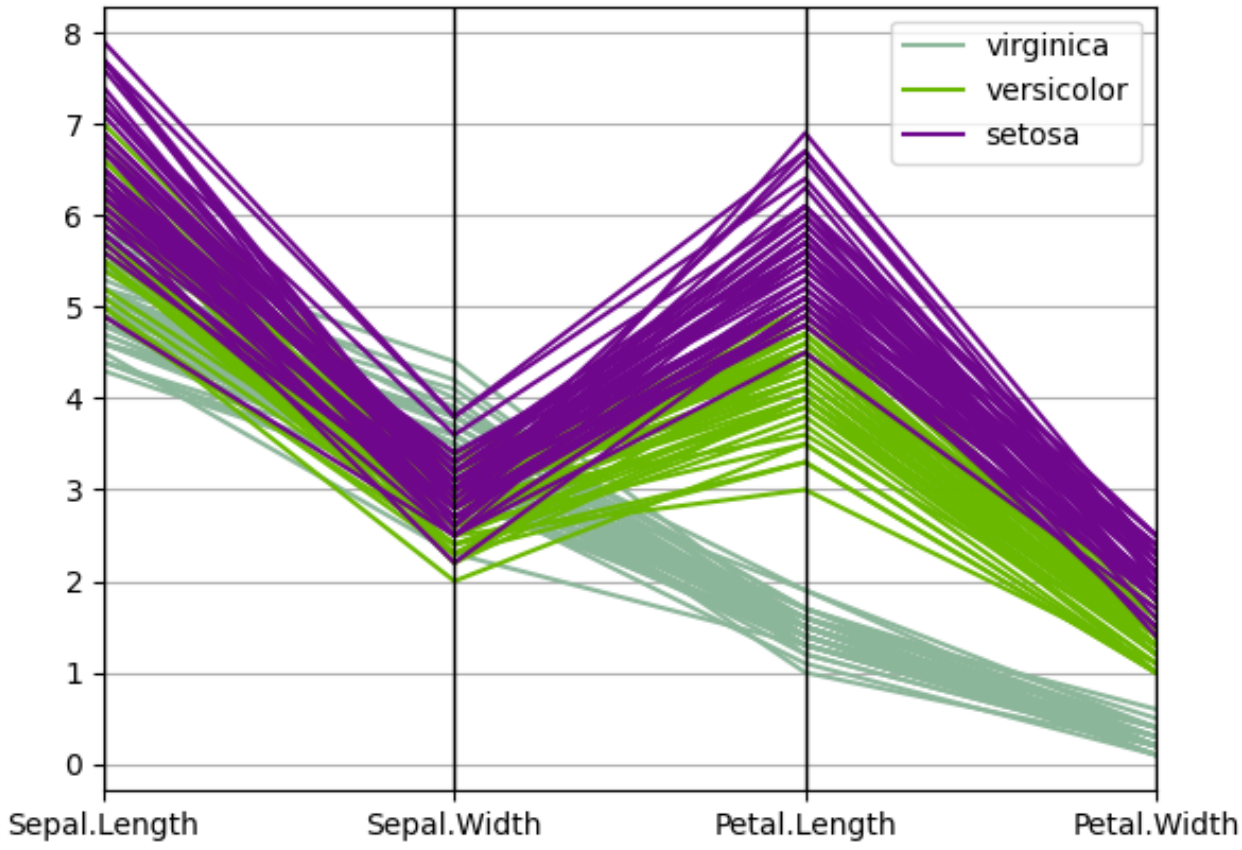
我们查看预测结果 发现预测失误的点集中在Setosa与Versicolor两类，因此猜想两类鸢尾花数据应当类似。

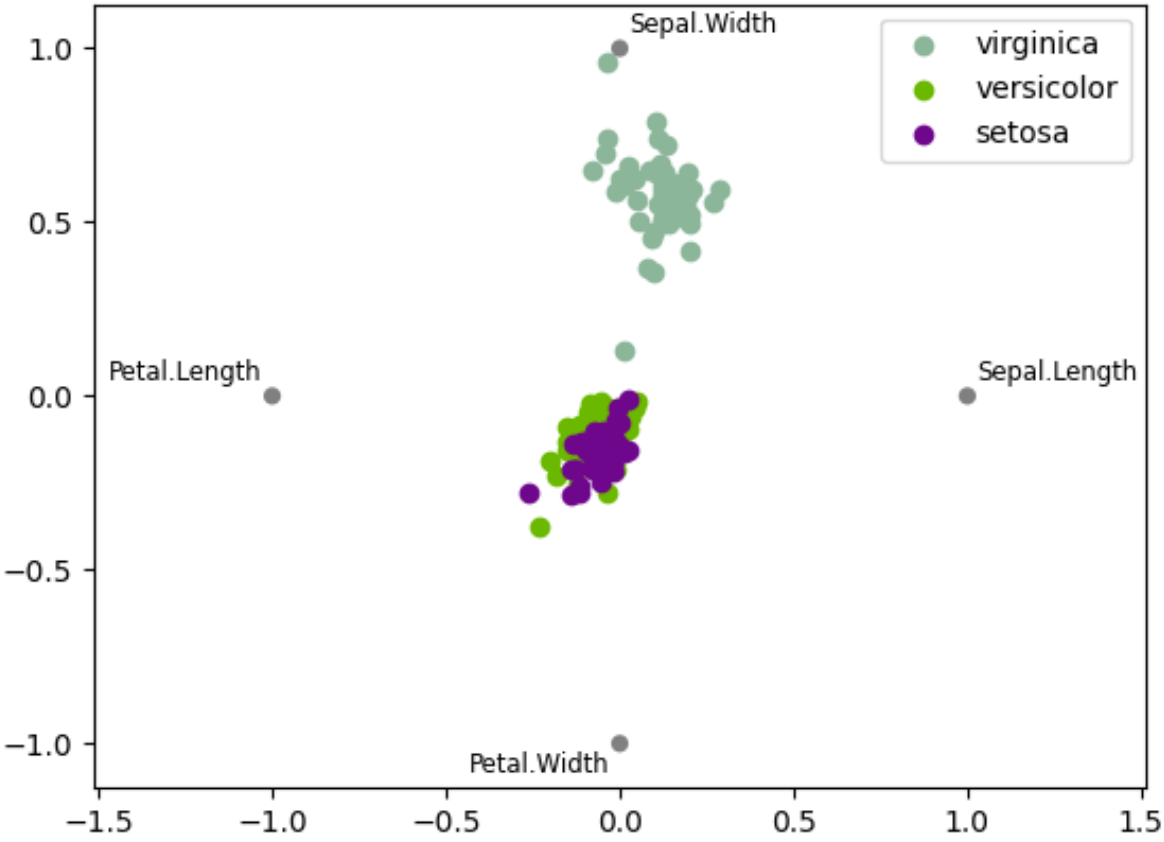
然而 由于有四个变量，传统的二维平面图与三维立体图均无法直观反应数据集的相似性，我们采用Andrews Curves 将每个样本的多变量量属性值转化为傅里里里叶级数的系数来创建曲线：

> Andrews curves have the functional form:
>
> **f(t) = x_1/sqrt(2) + x_2 sin(t) + x_3 cos(t) +**
> x_4 sin(2t) + x_5 cos(2t) + ...
>
> Where x coefficients correspond to the values of each dimension and t is linearly spaced between -pi and +pi. Each row of frame then corresponds to a single curve.

相应的 我们还有Parallel coordinates与radviz图对多属性的数据进行维度压缩的可视化处理，作图如下：


virginica
versicolor
setosa
15.0
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
−3
−2
−1
0
1
2
3


virginica
versicolor
setosa
8
7
6
5
4
3
2
1
0
Sepal.Length
Sepal.Width
Petal.Length
Petal.Width



这样，setosa和versicolor的相似性就十分直观了，这也造成了测试集在预测上的一些malfunction。

6.遇到的问题

总体上还是非常顺利的 除了一个折磨人的 IDE抽风问题

原因至今未知，如下：

541

老师 想请问一下为什么这个addCacheFile无法识别 QAQ

541

全网就没找到有关问题的信息

11:25

541

大不了下午我重装一下 idea 吧。。

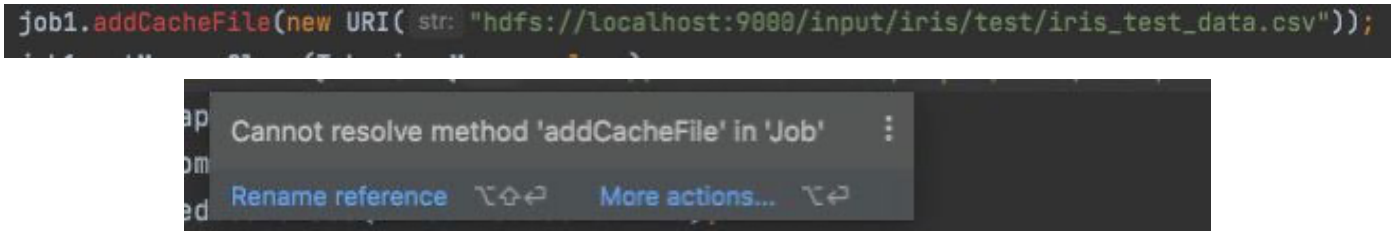
11:31

541

重装 idea

541

解决了



约莫就是 全世界人民都没有出现的问题 在541的电脑上出现了，无法解析addCacheFile与getCacheFile两个关键字 也大概就是尝试了一个多小时各种歪门邪道还是没法解决

一气之下 重装IDEA 问题就 莫名其妙解决了

于是541 有感而发：

重装ide解决一切魑魅魍魉 重装ide解决不了的就重装系统 重装系统解决不了的就重买电脑 重买电脑解决不了的建议重开。

完。