大数据作业7

徐冰清 **191098273**

作业要求

Iris数据集是常用的分类实验数据集，由Fisher, 1936收集整理。Iris也称鸢尾花卉数据集，是一类多重变量分析的数据集。数据集包含150个数据，分为3类，每类50个数据，每个数据包含4个属性。可通过花萼长度，花萼宽度，花瓣长度，花瓣宽度4个属性预测鸢尾花卉属于（Setosa，Versicolour，Virginica）三个种类中的哪一类。在MapReduce上任选一种分类算法（**KNN**，朴素贝叶斯或决策树）对该数据集进行分类预测，采用留出法对建模结果评估，70%数据作为训练集，30%数据作为测试集，评估标准采用精度**accuracy**。可以尝试对结果进行可视化的展示。

附件：

1. iris.names：数据集简介
2. iris.csv：数据集

实现思路

预处理

需要将数据分为两个部分：训练集和测试集，训练集占70%，测试集占30%。采用`random.sample(range(1, 150), 105)`生成105个随机而不重复的index用于训练集，其余的为预测集。因为每一次的样本都是不一样的，所以我就只取一次随机，用于以后所有的训练。

```
train_list = random.sample(range(1, 150), 105)
train = iris[iris['index'].isin(train_list)]
predict = iris[~ iris['index'].isin(train_list)]
```

三个种类setosa、versicolor、virginica取代为1、2、3，在储存时略去`index`和`header`。

```
predict.loc[predict['Species'] == 'setosa', 'Species'] = 1
predict.loc[predict['Species'] == 'versicolor', 'Species'] = 2
predict.loc[predict['Species'] == 'virginica', 'Species'] = 3
predict.to_csv('./data/predict.csv', sep=' ', index=None,
header=None)
train.to_csv('./data/train.csv', sep=' ', index=None, header=None)
```

最后，手动将`.csv`改成`.txt`文件，就可以作为输入文件了！

train

数据格式：用空格分割，三个种类setosa、versicolor、virginica分别对应1、2、3。

```
5.1 3.5 1.4 0.2 1
4.9 3.0 1.4 0.2 1
7.0 3.2 4.7 1.4 2
6.4 3.2 4.5 1.5 2
5.8 2.7 5.1 1.9 3
7.1 3.0 5.9 2.1 3
...共105条数据
```

predict

为了方便之后与预测结果相互比对，确定精度accuracy，没有将Species都设置为-1，而是保留了原有的种类，这对结果没有影响。

```
4.7 3.2 1.3 0.2 1
5.4 3.9 1.7 0.4 1
6.5 2.8 4.6 1.5 2
5.7 2.8 4.5 1.3 2
6.7 2.5 5.8 1.8 3
6.5 3.2 5.1 2.0 3
...共45条数据
```

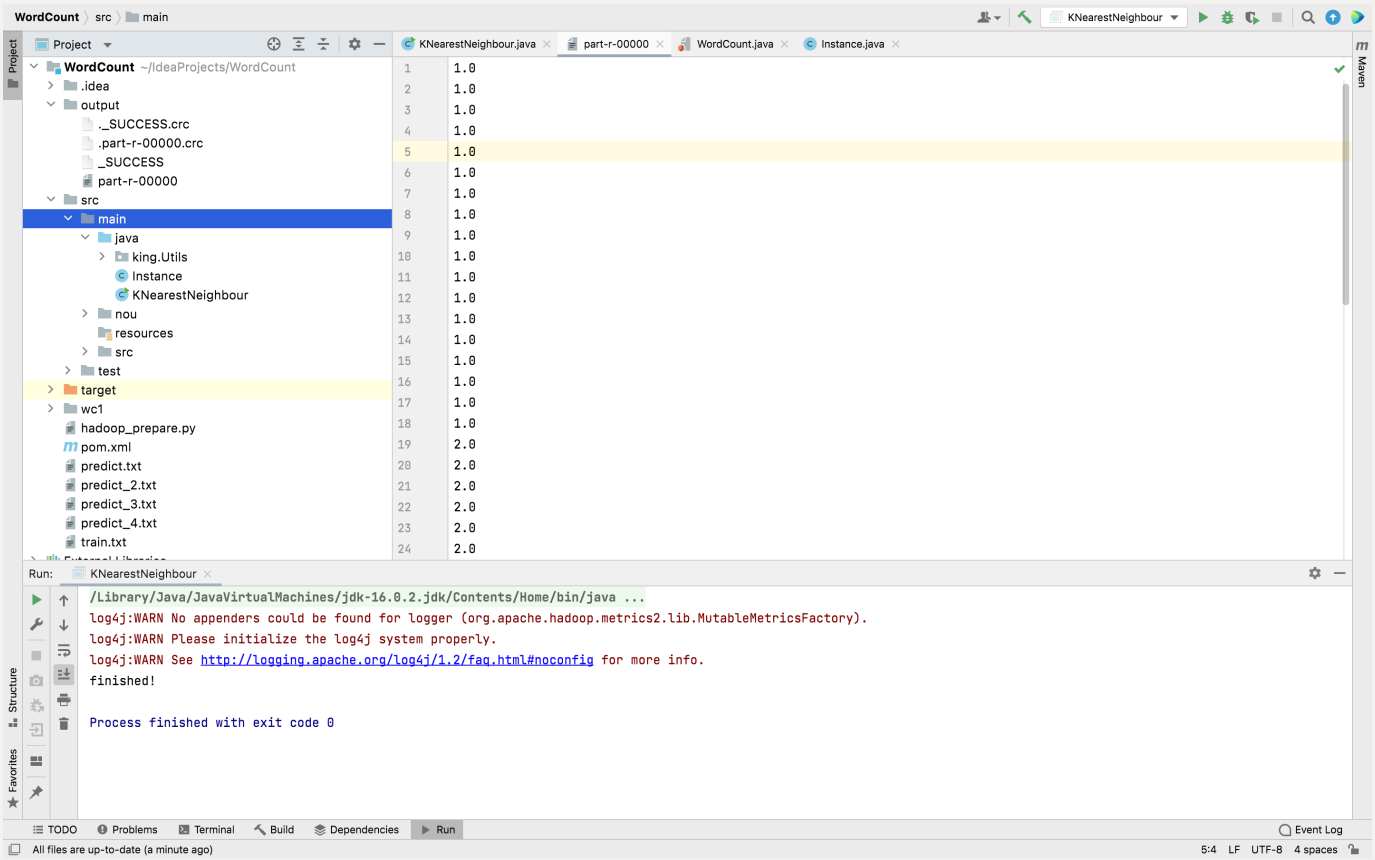
mapreduce实现

在KNN代码的基础上稍做修改即可。

输出的`result`格式为

```
1
1
2
3
2
3
3
...共45条数据
```

每一行与输入的`predict.txt`相对应。



精度如下：

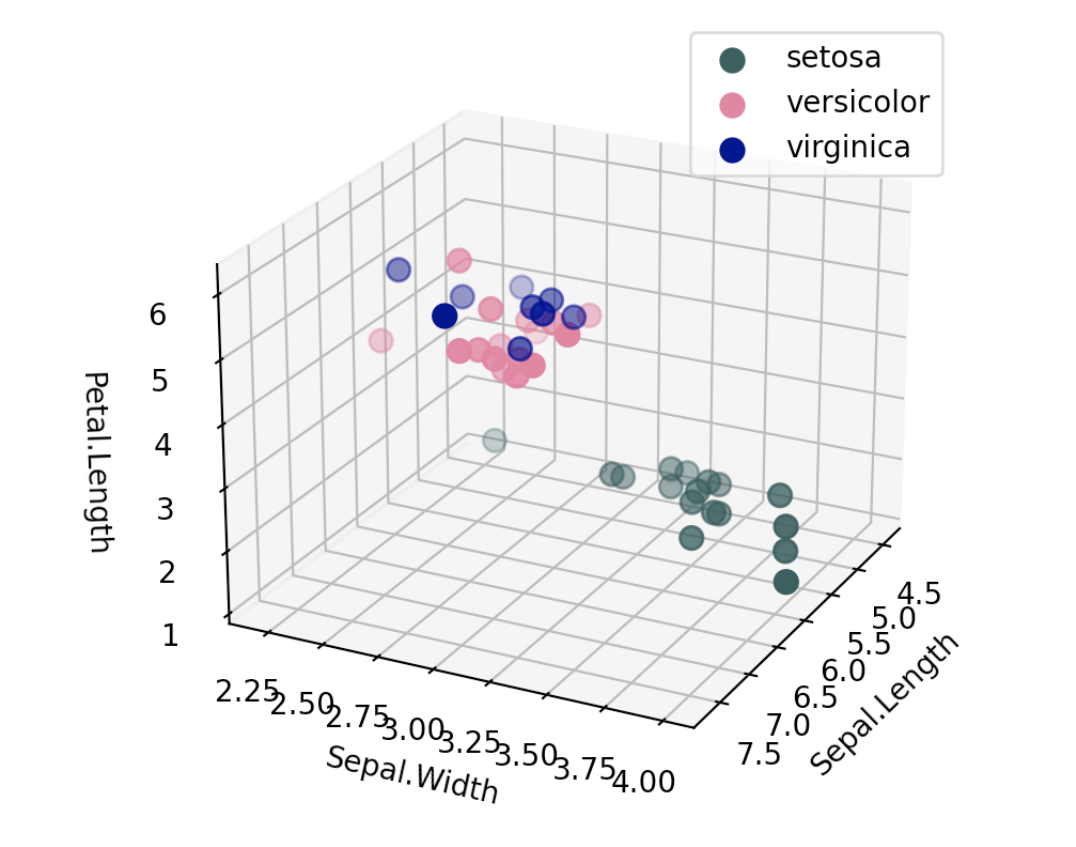
NEIGHBOUR NUM	2	3	4
accuracy	91.1%	97.8%	97.8%

经过观测，我发现setosa总是能够被正确预测，但是versicolor和virginica总被混淆，不少virginica被预测为了versicolor。neighbour num为3和4的时候，预测结果完全相同。

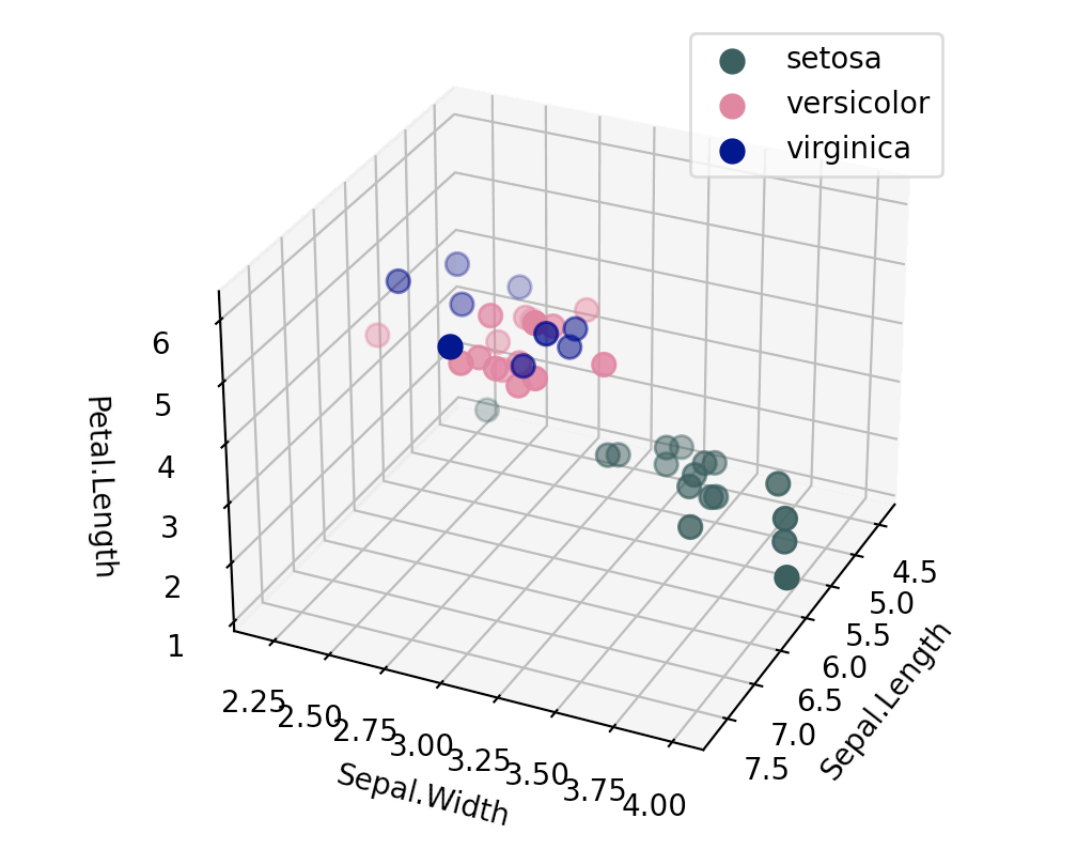
可视化

采用python中`matplotlib.pyplot`和`mpl_toolkits.mplot3d`进行可视化，可以画出三维图像，为此省略了第四维Petal.Width。分成三个部分绘图，使三个种类呈现出不同的颜色。其中，种类是预测的结果。

neighbour num = 2



neighbour num = 3, 4



从图中也可以看出，相比于与世独立的setosa，versicolor和virginica两组距离较近，的确不易分割，这与我之前的观察相符合。

实验反思

1. 在预处理时，多做一些处理，可以方便后续实验的进行，达到事半功倍的效果。

2. 暂未想出将四个维度都在图上可视化的方法，所以图上只展示了三个维度，这是有待改进的部分。
3. 实验中有不少手动实现的部分，例如output与predict聚合时，我手动把预测结果与原本特征结合在了一起，例如我手工将csv转化成txt，这都有待优化。