数据挖掘与数据融合编程实验

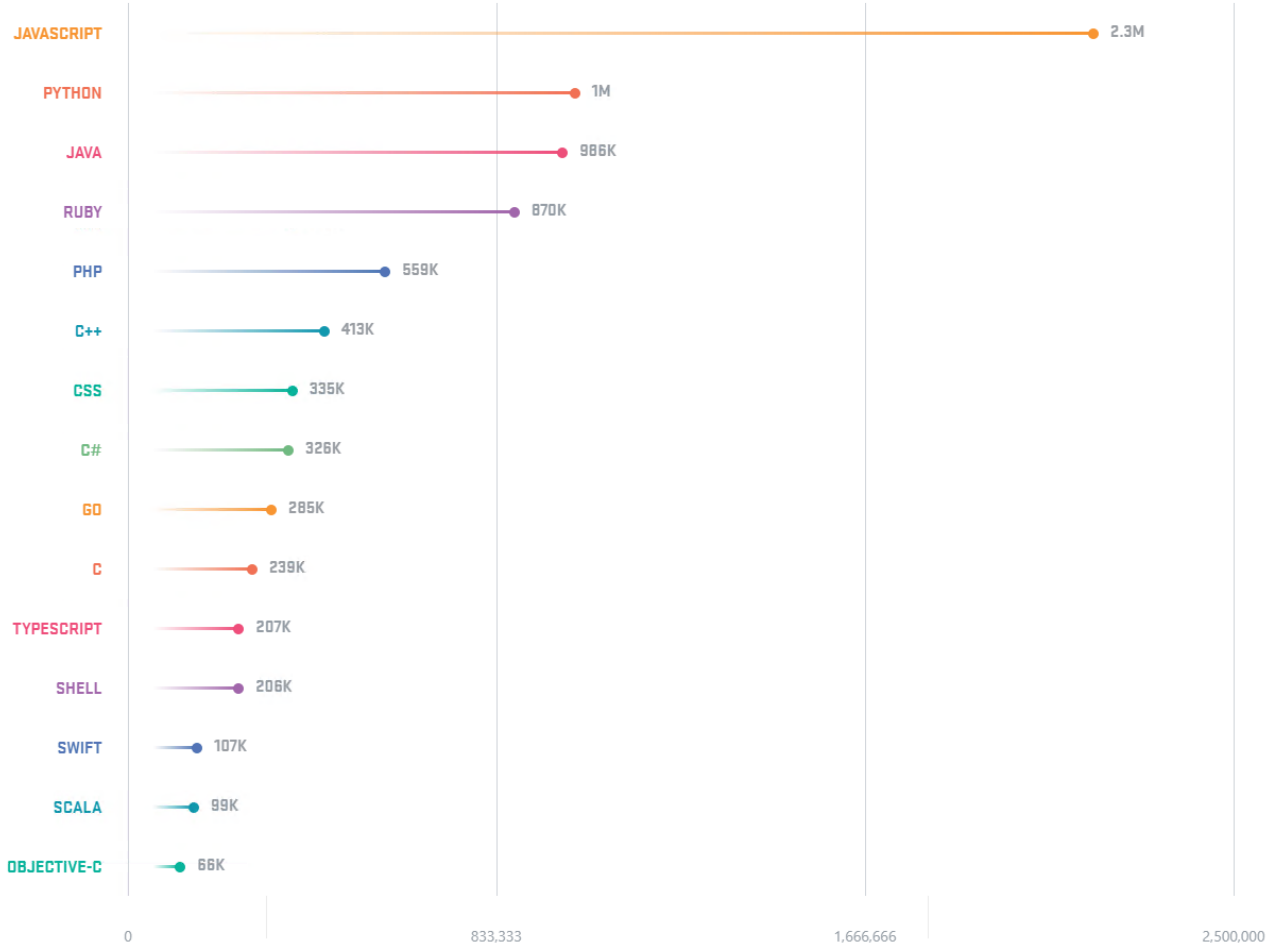
助课讲师： 潘 戈
指导老师： 荣 冈

Github: The State of the Octoverse 2017

https://octoverse.github.com/

第一阶段： Python基础

Github: The State of the Octoverse 2017



The fifteen most popular languages on GitHub by opened pull request.

Python replaced Java as the second-most popular language on GitHub, with 40 percent more pull requests opened this year than last.

大势所趋　众望所归

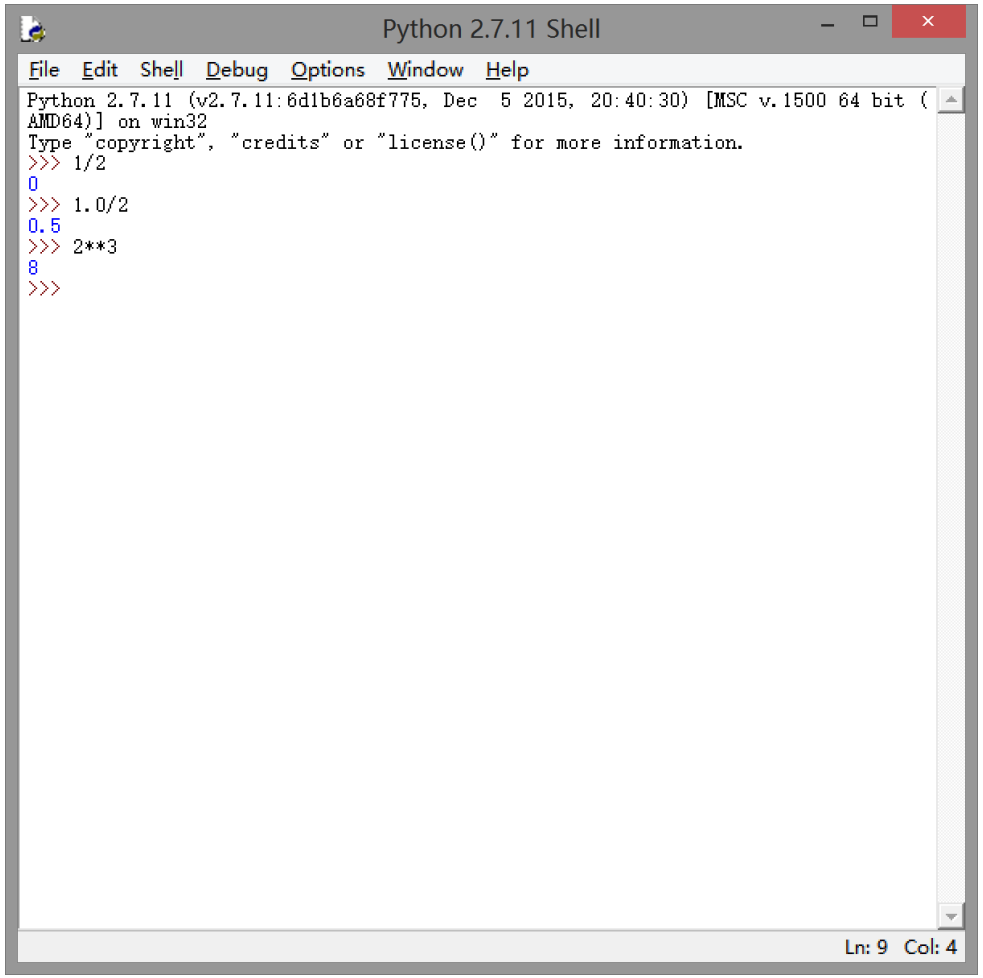
https://octoverse.github.com/

Getting Started

- https://www.python.org/
- https://wiki.python.org/moin/BeginnersGuide/Download
- https://docs.python.org/2/tutorial/index.html

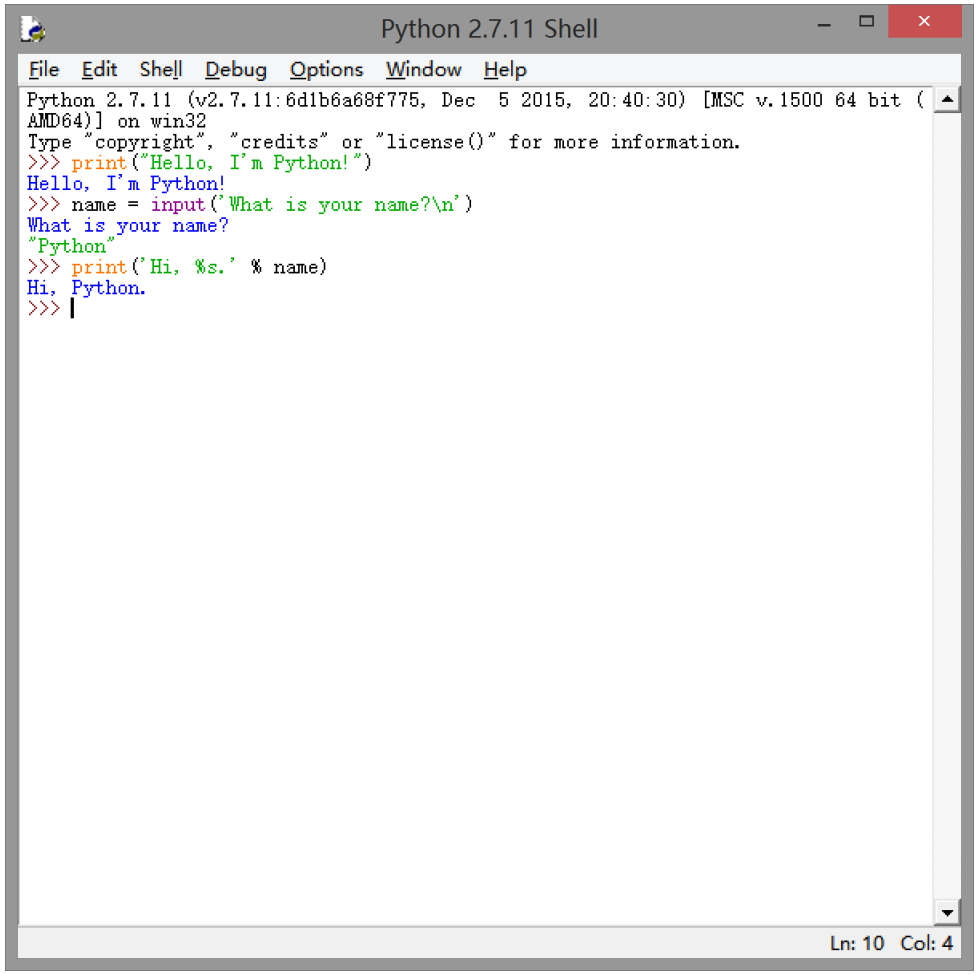
Getting Started

```
>>> 1/2
0
>>> 1.0/2
0.5
>>> 2**3
8
```



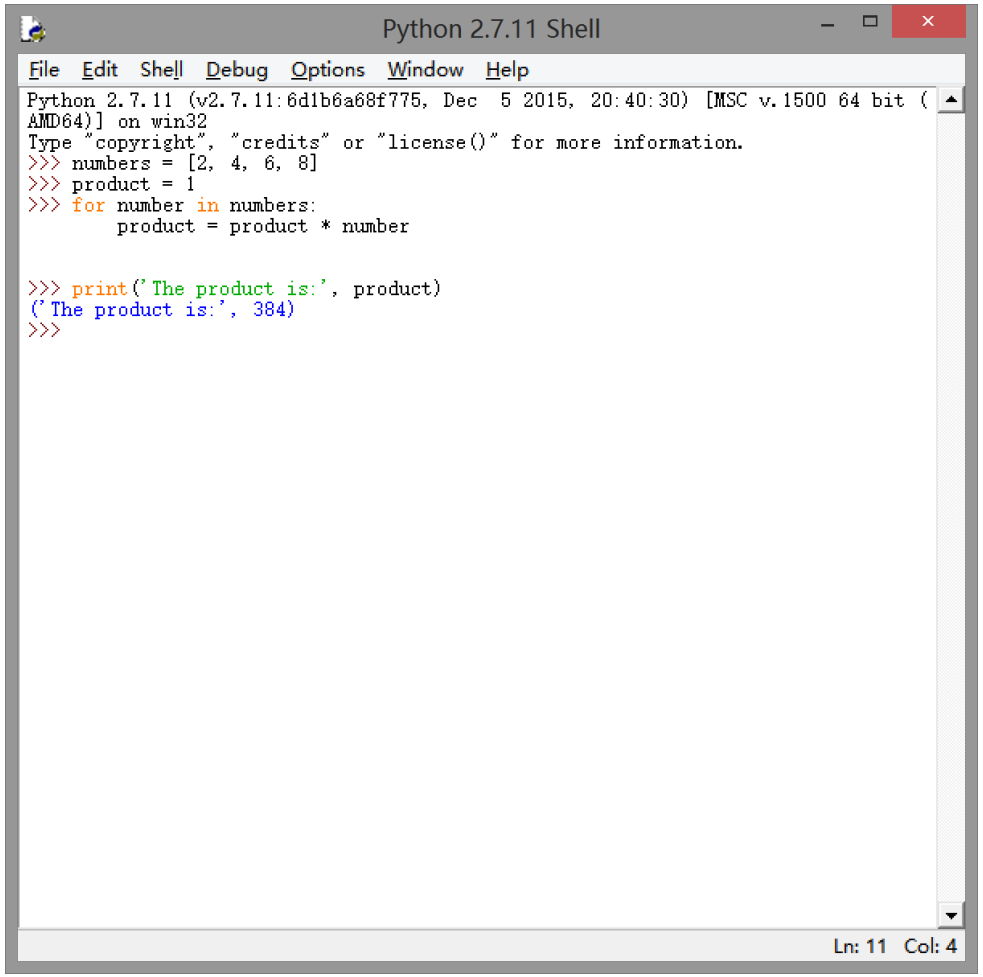
Getting Started

```
>>> print("Hello, I'm Python!")
Hello, I'm Python!

>>> name = input('What is your name?\n')
What is your name?
"Python"

print('Hi, %s.' % name)
Hi, Python.
```



Getting Started

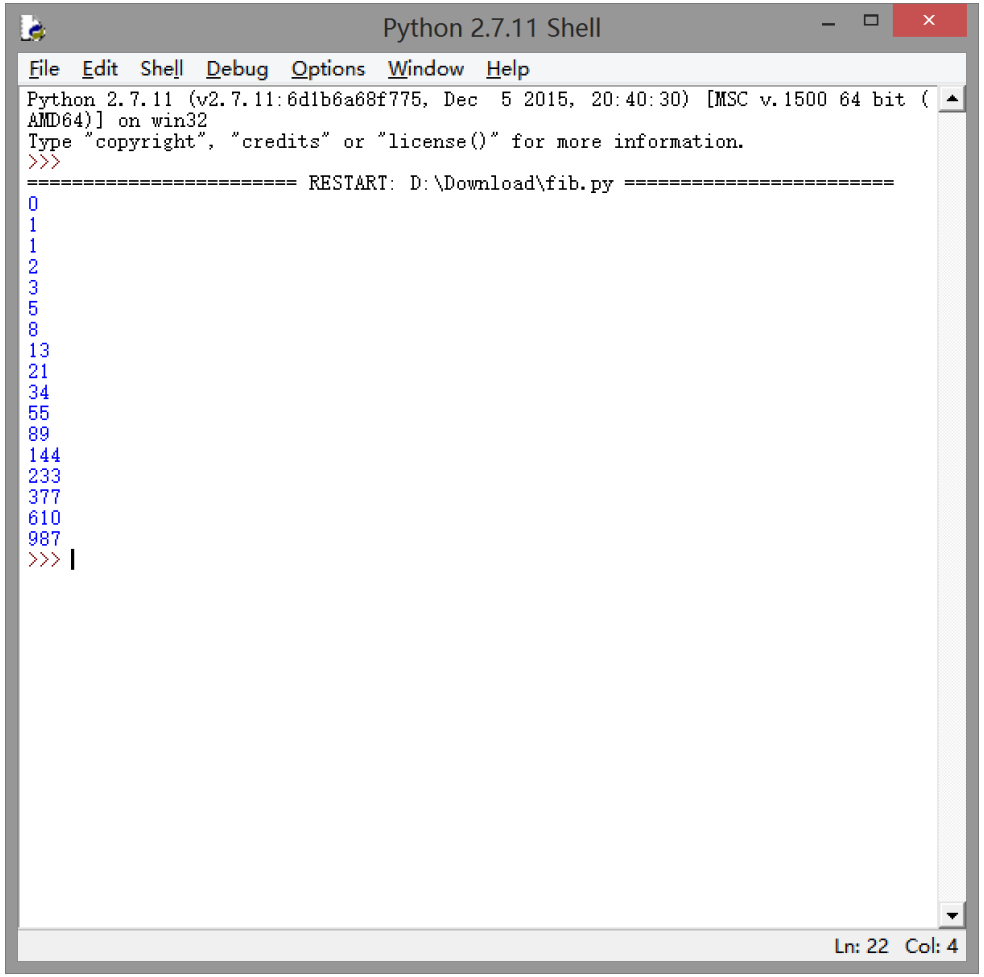
```
>>> numbers = [2, 4, 6, 8]
>>> product = 1
>>> for number in numbers:
...     product = product * number
...
>>> print('The product is:', product)
('The product is:', 384)
```



Getting Started

fib.py

```
def fib(n):
    a, b = 0, 1
    while a < n:
        print(a)
        a, b = b, a+b
fib(1000)
```



论调包侠的自我修养

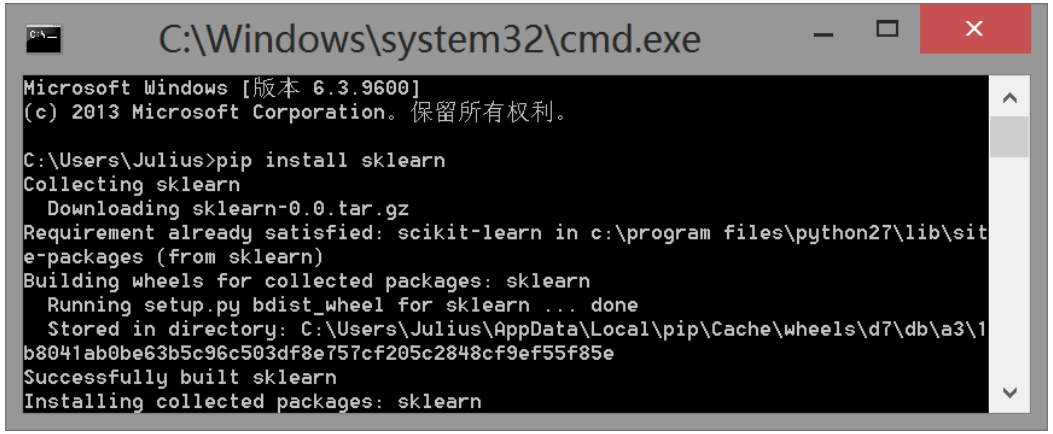
- numpy
- http://www.numpy.org/

- pandas
- http://pandas.pydata.org

- sklearn
- http://scikit-learn.org/

- Windows: 命令提示符
- Unix/Linux: Shell

- pip install sklearn



伟大的防火墙

File "C:\Python27\lib\site-packages\pip_vendor\requests\packages\urllib3\re

sponse.py", line 227, in read

raise ReadTimeoutError(self._pool, None, 'Read timed out.')

ReadTimeoutError: HTTPSConnectionPool(host='pypi.python.org', port=443): Read

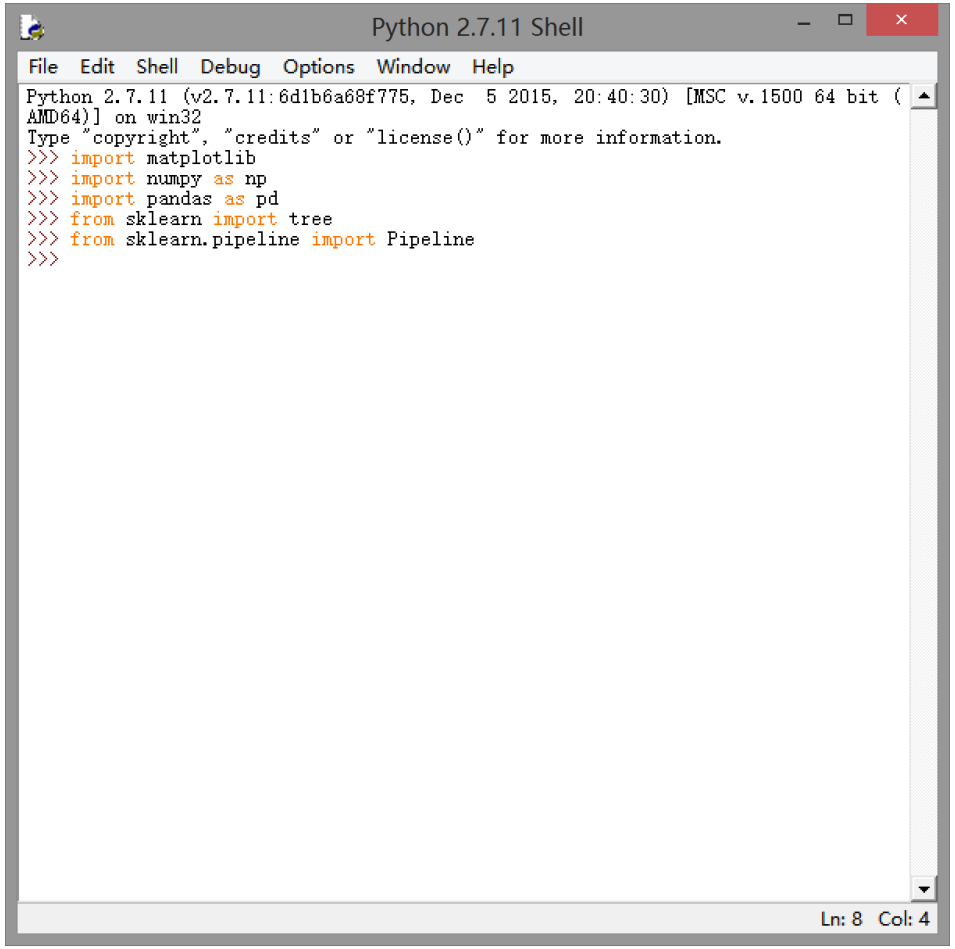
timed out.

伟大的防火墙

- 方案一：下载安装包
  - pip install /path/package
- 方案二：更换Python默认软件镜像源
  - 豆瓣：http://pypi.douban.com/simple/
  - 清华：https://pypi.tuna.tsinghua.edu.cn/simple/
- 方案三：吾知子之所以距我，吾不言。——《墨子》

论调包侠的自我修养

```
>>> import matplotlib
>>> import numpy as np
>>> import pandas as pd
>>> from sklearn import tree
>>> from sklearn.pipeline import Pipeline
```



实验任务

- 安装python 2.7
- 熟悉pip的基本使用方法
- 学习python的基本语法
- 简单使用几个常用库

第二阶段： 机器学习

鸢尾花数据集

- 鸢尾花的三个分类
  - Iris-setosa
  - Iris-versicolor
  - Iris-virginica
- 每类提供50个数据集
- 共150个数据集
- 取80%做训练，20%做测试

- 数据集描述的四个属性
  - sepal length, 花萼长度
  - sepal width, 花萼宽度
  - petal length, 花瓣长度
  - petal width, 花瓣宽度
- 根据四大属性
  预测鸢尾花的分类

特征抽取与特征选择（降维）

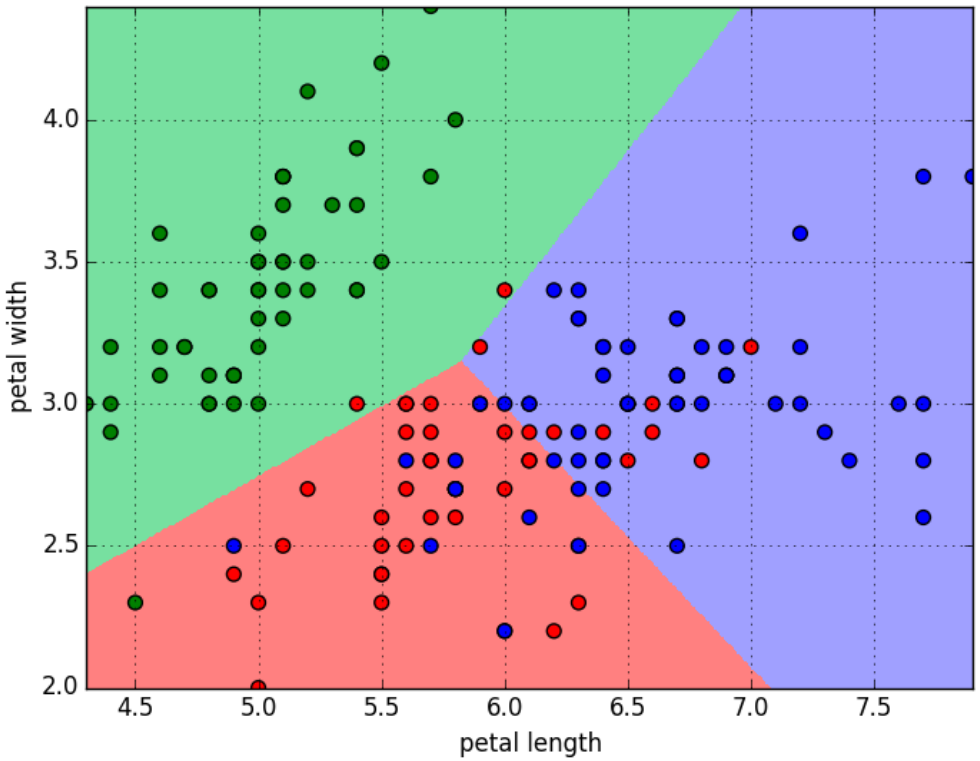
- 特征抽取（Feature Extraction）
- Creatting a subset of new features by combinations of the exsiting features.
- 特征抽取后的新特征是原来特征的一个映射。

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \xrightarrow{\text{feature extraction}} \begin{bmatrix} y_1 \\ y_2 \\ \\ y_M \end{bmatrix} = f\left(\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}\right)$$

- 特征选择（Feature Selection）
- choosing a subset of all the features (the ones more informative)
- 特征选择后的特征是原来特征的一个子集。

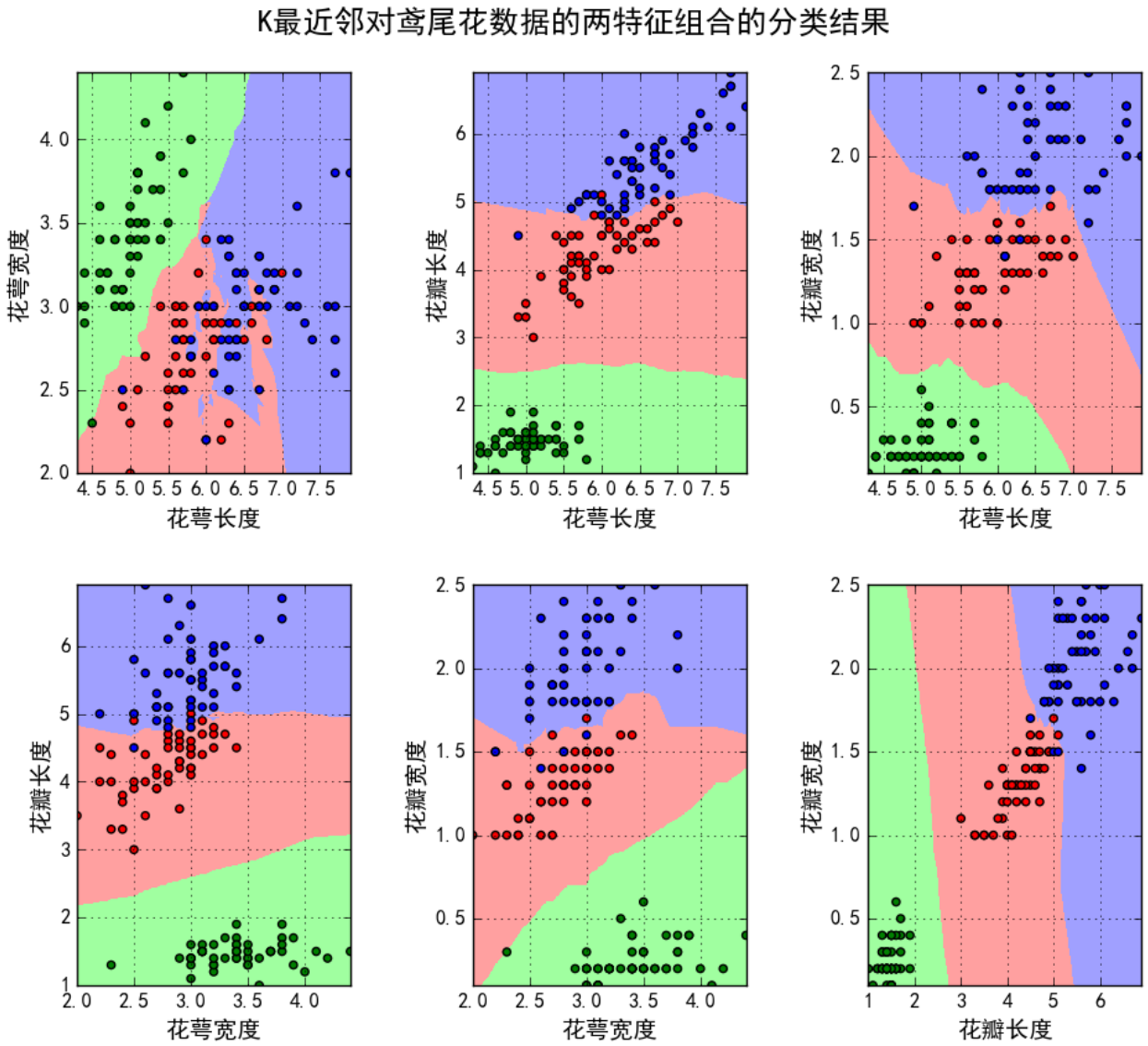
$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \xrightarrow{\text{feature selection}} \begin{bmatrix} x_{i_1} \\ x_{i_2} \\ \\ x_{i_M} \end{bmatrix}$$

Logistic Regression



准确率（4个特征） = 预测正确的数目/测试集样本总数 = 86.67%

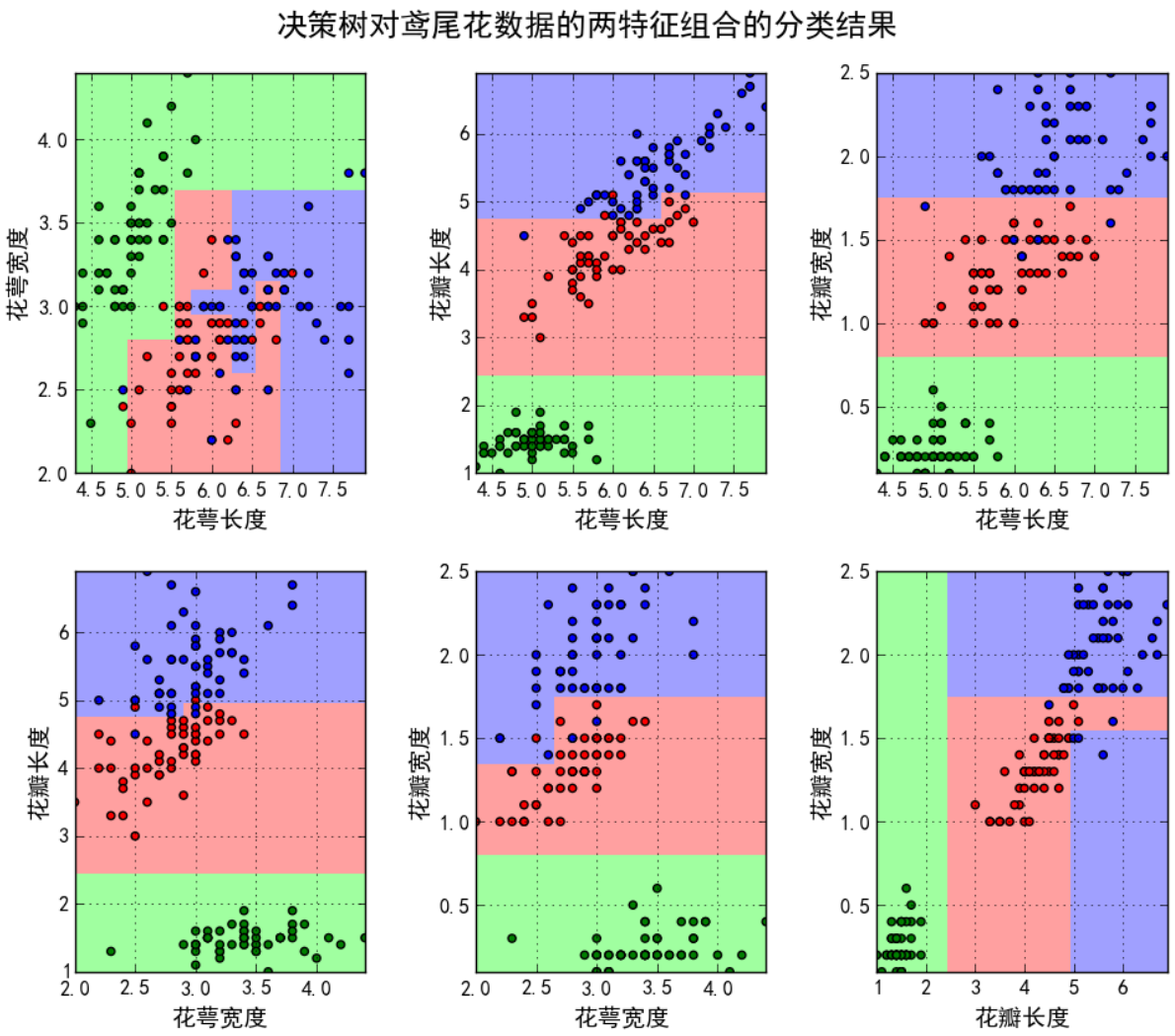
KNN



准确率（4个特征） = 96.67%

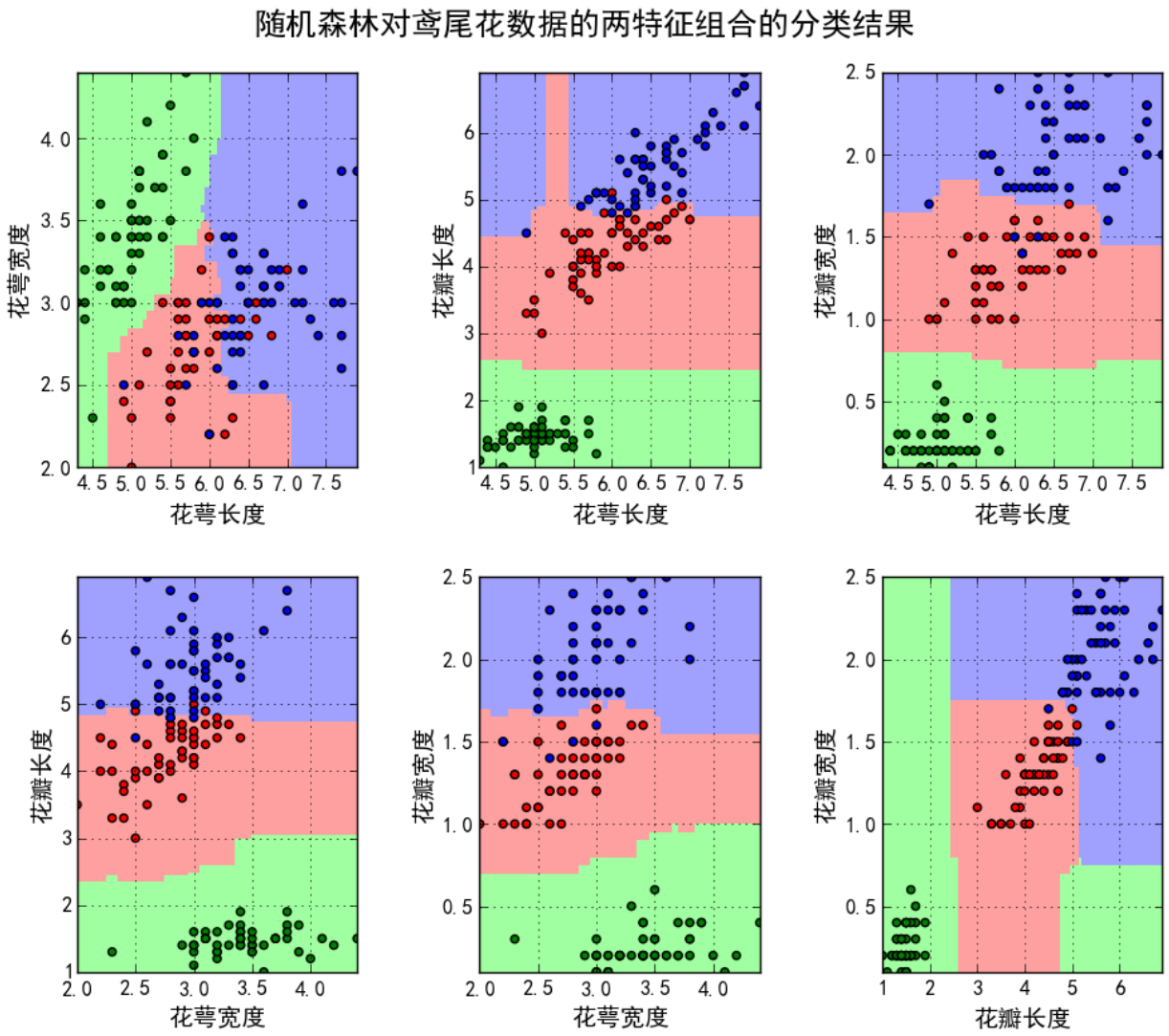
Decision Tree

- 特征选择后的训练集准确率
  - 花萼长度 + 花萼宽度 = 82.00%
  - 花萼长度 + 花瓣长度 = 96.67%
  - 花萼长度 + 花瓣宽度 = 96.00%
  - 花萼宽度 + 花瓣长度 = 95.33%
  - 花萼宽度 + 花瓣宽度 = 96.67%
  - 花瓣长度 + 花瓣宽度 = 98.00%
- 4个特征下的测试集准确率
  - 96.67%



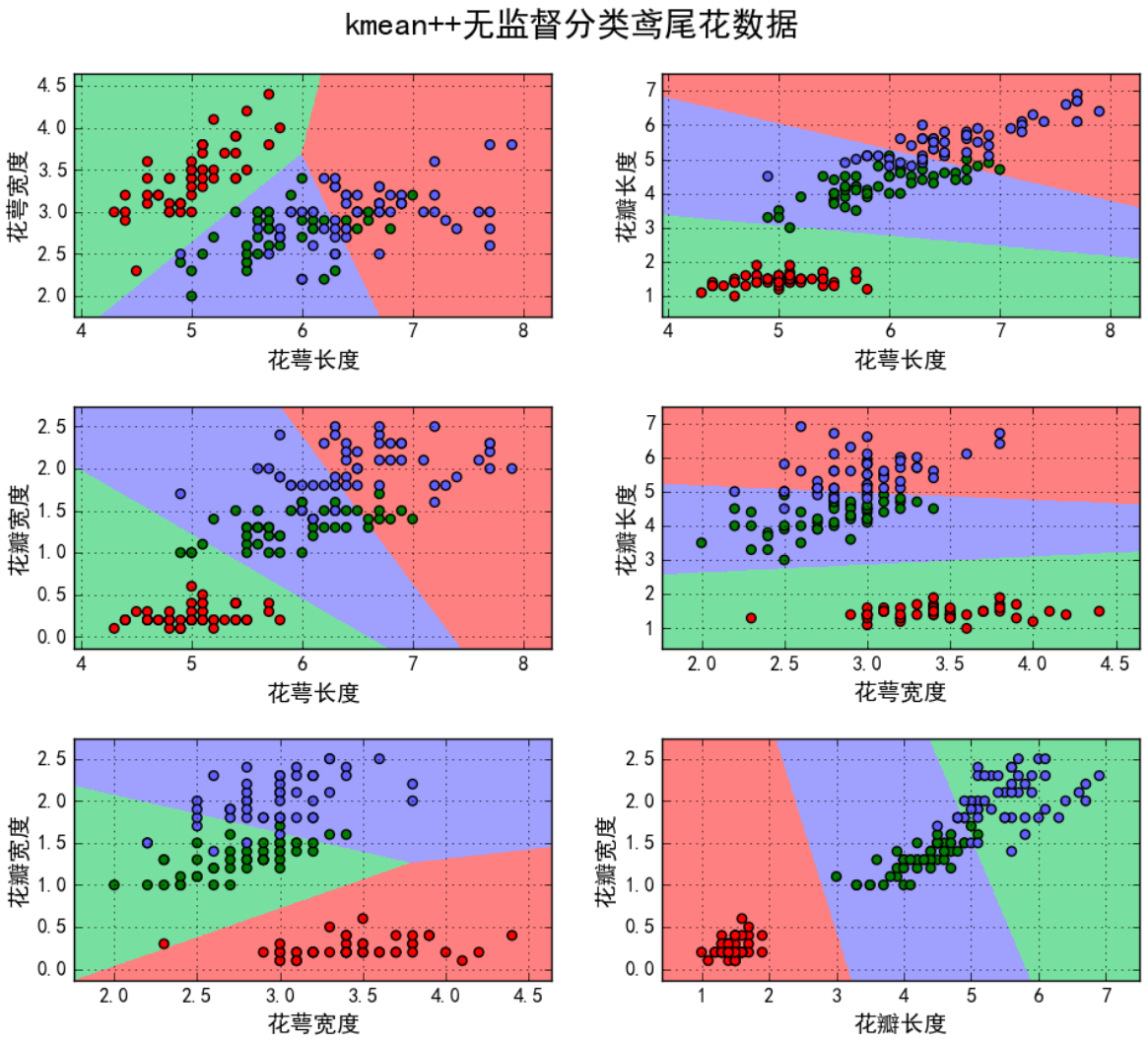
Random Forest

- 特征选择后的训练集准确率
  - 花萼长度 + 花萼宽度 = 84.00%
  - 花萼长度 + 花瓣长度 = 97.33%
  - 花萼长度 + 花瓣宽度 = 96.67%
  - 花萼宽度 + 花瓣长度 = 96.00%
  - 花萼宽度 + 花瓣宽度 = 96.67%
  - 花瓣长度 + 花瓣宽度 = 96.67%
- 4个特征下的测试集准确率
  - 100%

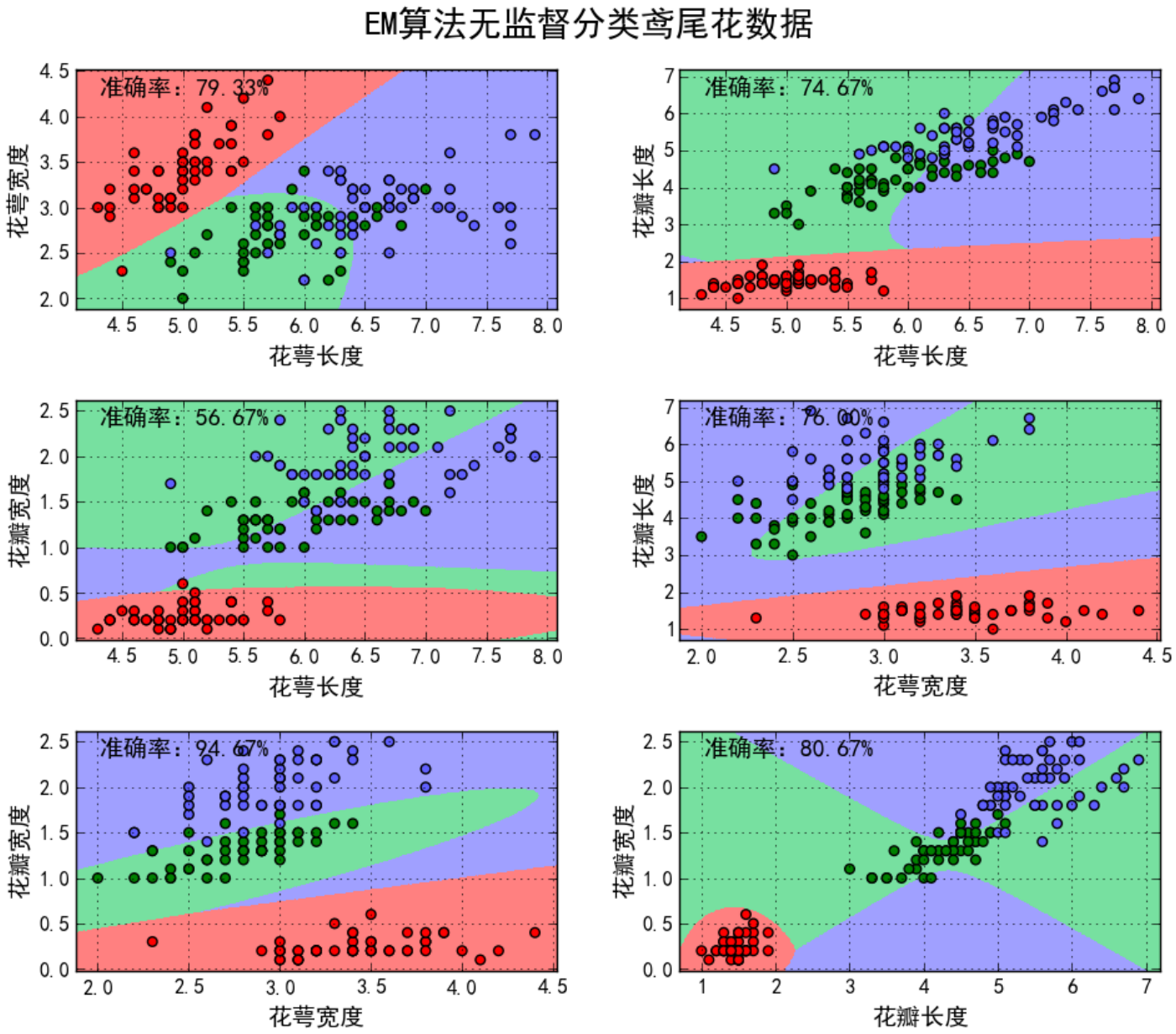


K-Means

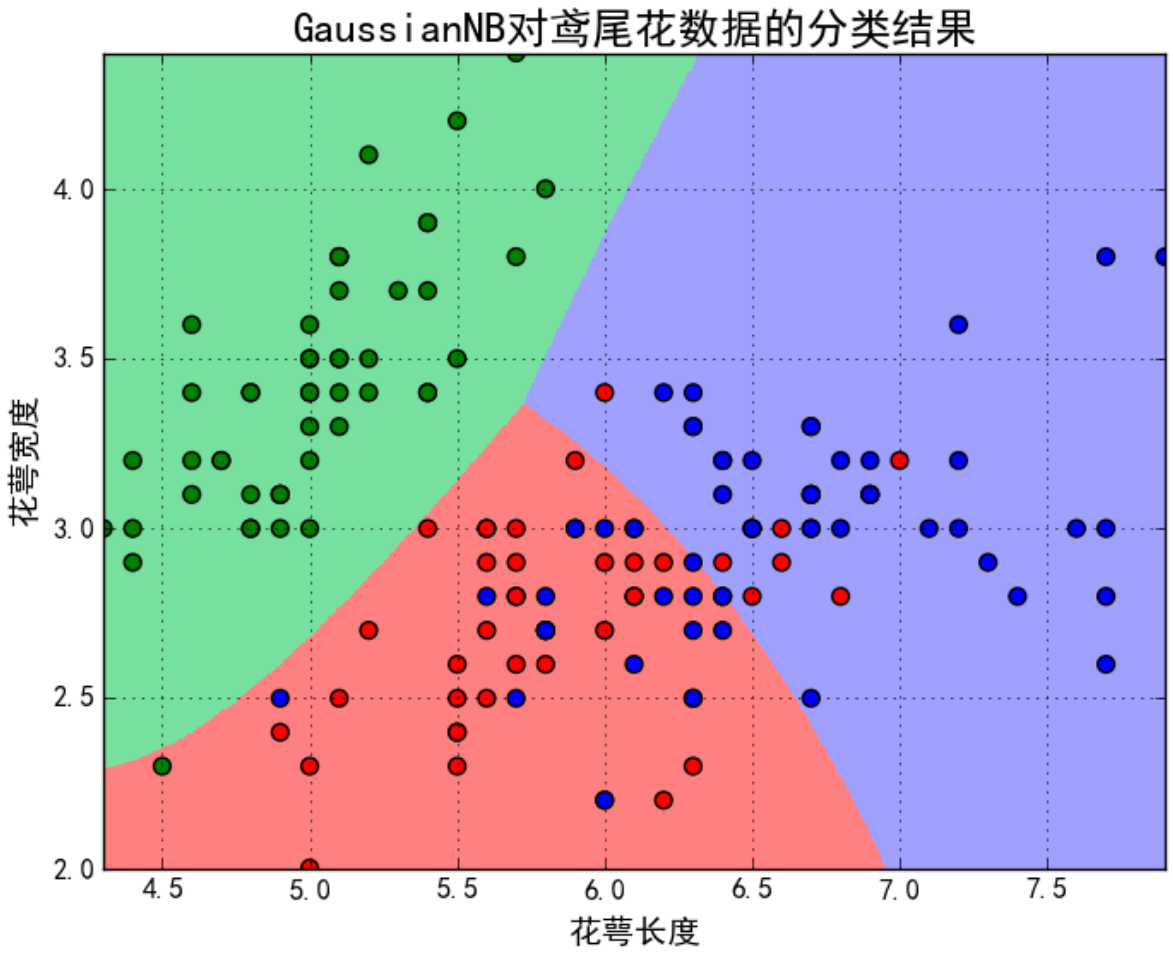
- 特征选择后的训练集准确率
  - 花萼长度 + 花萼宽度 = 82.00%
  - 花萼长度 + 花瓣长度 = 88.00%
  - 花萼长度 + 花瓣宽度 = 82.67%
  - 花萼宽度 + 花瓣长度 = 92.67%
  - 花萼宽度 + 花瓣宽度 = 92.67%
  - 花瓣长度 + 花瓣宽度 = 96.00%
- 4个特征下的测试集准确率
  - 90%



EM


EM算法无监督分类鸢尾花数据
准确率：79.33%
花萼宽度
花萼长度
准确率：74.67%
花瓣长度
花萼长度
准确率：56.67%
花瓣宽度
花萼长度
准确率：76.00%
花瓣长度
花萼宽度
准确率：94.67%
花瓣宽度
花萼宽度
准确率：80.67%
花瓣宽度
花瓣长度

Naive Bayes


GaussianNB对鸢尾花数据的分类结果
花萼宽度
花萼长度
4.0
3.5
3.0
2.5
2.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5

编程学习

- 阅读理解并运行Iris文件夹下的算法测试代码
  - Iris_LogisticRegression.py
  - Iris_KNN.py
  - Iris_DecisionTree.py
  - Iris_RandomForest.py
  - Iris_kMeans.py
  - Iris_DBSCAN.py

进阶任务

- 阅读理解algorithms文件夹下的算法实现代码
  - linear_model/logistic.py
  - neighbors/regression.py
  - tree/tree.py
  - ensemble/forest.py
  - cluster/k_means_.py
  - cluster/dbscan_.py