数据科学基础

作业报告

院　　系：信息与通信工程学院

班　　级：2019211112

学　　号：2021523016

类　　别：交流生

姓　　名：徐川峰

指导老师：刘芳

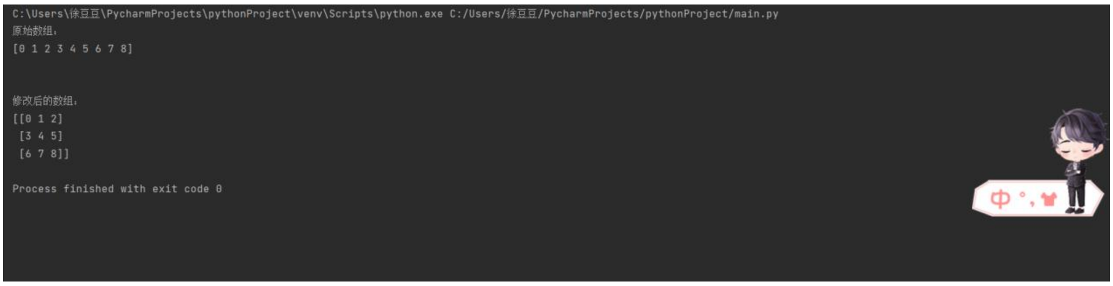
2021 年 10 月 19 日

任务 1：numpy 创建数组，数组形状修改结果截图

对应关键代码段：

```
import numpy as np

a = np.arange(9)

print ('原始数组：')

print (a)

print ('\n')

b = a.reshape(3,3)

print ('修改后的数组：')

print (b)
```

运行结果截图：



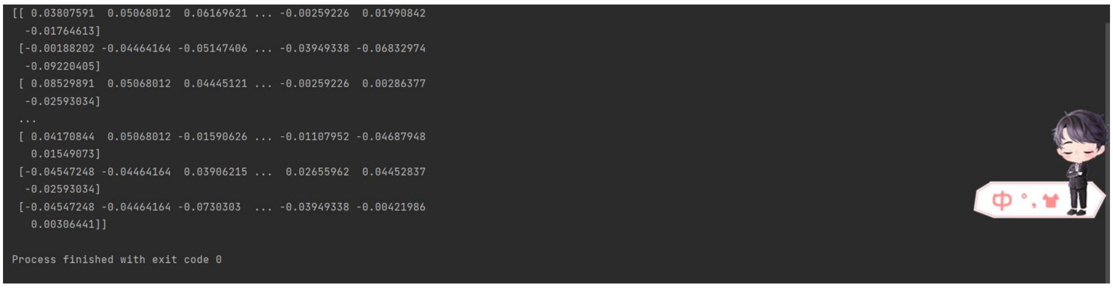
```
C:\Users\徐豆豆\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/徐豆豆/PycharmProjects/pythonProject/main.py
原始数组：
[0 1 2 3 4 5 6 7 8]


修改后的数组：
[[0 1 2]
 [3 4 5]
 [6 7 8]]

Process finished with exit code 0
```

任务 2：输出糖尿病数据集所有变量值及其数组形状

对应关键代码段：

```
import matplotlib.pyplot as plt

import numpy as np

from sklearn import datasets

from sklearn.datasets import load_diabetes

diabetes = datasets.load_diabetes()

print(diabetes.data)
```

运行结果截图：



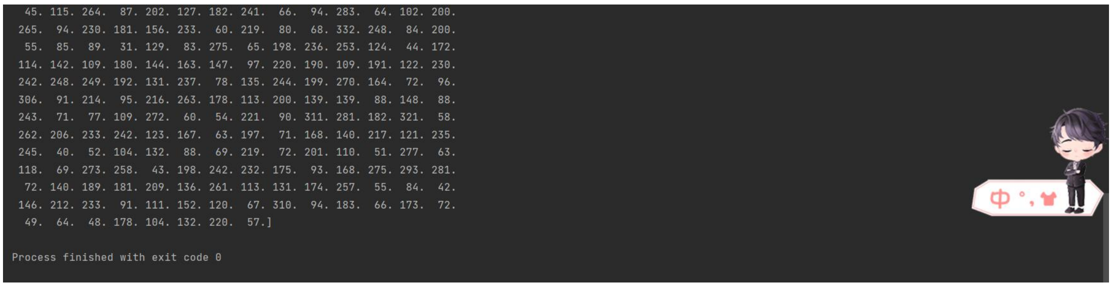
```
[[ 0.03807591  0.05068012  0.06169621 ... -0.00259226  0.01990842
  -0.01764613]
 [-0.00188202 -0.04464164 -0.05147406 ... -0.03949338 -0.06832974
  -0.09220405]
 [ 0.08529891  0.05068012  0.04445121 ... -0.00259226  0.00286377
  -0.02593034]
 ...
 [ 0.04170844  0.05068012 -0.01590626 ... -0.01107952 -0.04687948
   0.01549073]
 [-0.04547248 -0.04464164  0.03906215 ...  0.02655962  0.04452837
  -0.02593034]
 [-0.04547248 -0.04464164 -0.0730303  ... -0.03949338 -0.00421986
   0.00306441]]

Process finished with exit code 0
```

任务 3：输出糖尿病数据所有样本真实标签及其数组形状

对应关键代码段：

```
import matplotlib.pyplot as plt

import numpy as np

from sklearn import datasets

from sklearn.datasets import load_diabetes

diabetes = datasets.load_diabetes()

print(diabetes.data)

print(diabetes.target)
```

运行结果截图：



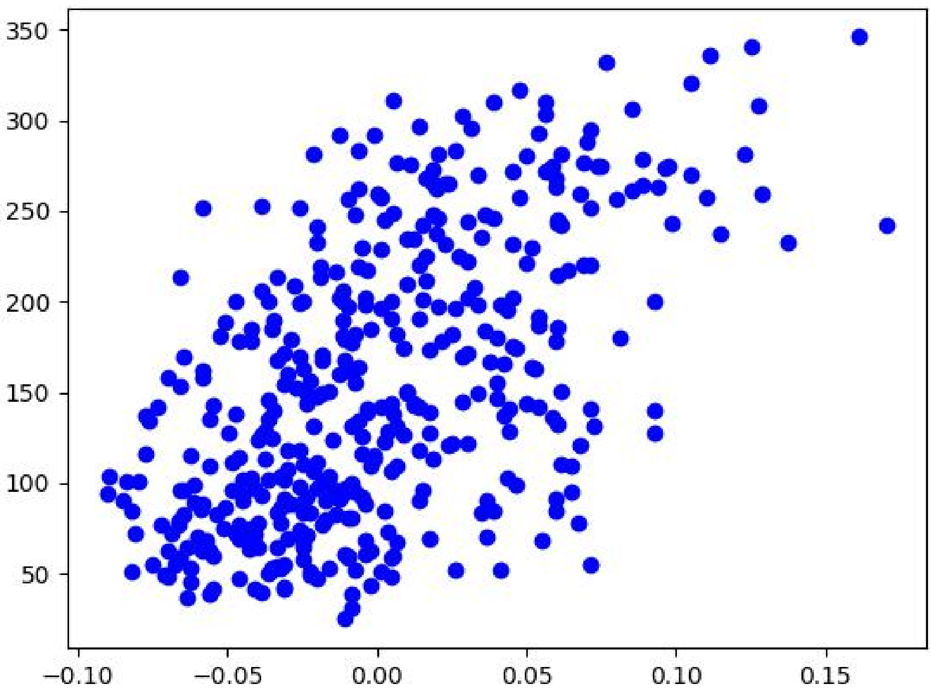
```
  45. 115. 264.  87. 202. 127. 182. 241.  66.  94. 283.  64. 102. 200.
 265.  94. 230. 181. 156. 233.  60. 219.  80.  68. 332. 248.  84. 200.
  55.  85.  89.  31. 129.  83. 275.  65. 198. 236. 253. 124.  44. 172.
 114. 142. 109. 180. 144. 163. 147.  97. 220. 190. 109. 191. 122. 230.
 242. 248. 249. 192. 131. 237.  78. 135. 244. 199. 270. 164.  72.  96.
 306.  91. 214.  95. 216. 263. 178. 113. 200. 139. 139.  88. 148.  88.
 243.  71.  77. 109. 272.  60.  54. 221.  90. 311. 281. 182. 321.  58.
 262. 206. 233. 242. 123. 167.  63. 197.  71. 168. 140. 217. 121. 235.
 245.  40.  52. 104. 132.  88.  69. 219.  72. 201. 110.  51. 277.  63.
 118.  69. 273. 258.  43. 198. 242. 232. 175.  93. 168. 275. 293. 281.
  72. 140. 189. 181. 209. 136. 261. 113. 131. 174. 257.  55.  84.  42.
 146. 212. 233.  91. 111. 152. 120.  67. 310.  94. 183.  66. 173.  72.
  49.  64.  48. 178. 104. 132. 220.  57.]

Process finished with exit code 0
```

任务 4：输出测试数据散点图（学号尾号为基数散点图为红色方形，学号尾号为偶数散点图为蓝色圆形）

对应关键代码段：

```
import matplotlib.pyplot as plt

import numpy as np

from sklearn import datasets
from sklearn.datasets import load_diabetes

diabetes = datasets.load_diabetes()

print(diabetes.data)

print(diabetes.target)

diabetes = datasets.load_diabetes()

diabetes_X = diabetes.data[:, 2]

diabetes_X_train = diabetes_X[:-20]

diabetes_y_train = diabetes.target[:-20]

plt.scatter(diabetes_X_train, diabetes_y_train,

color='blue', marker='o')

plt.show()
```

运行结果截图：



任务 5：diabetes_X_train=np.array(diabetes_X_train).reshape(-1,1)句的意义？

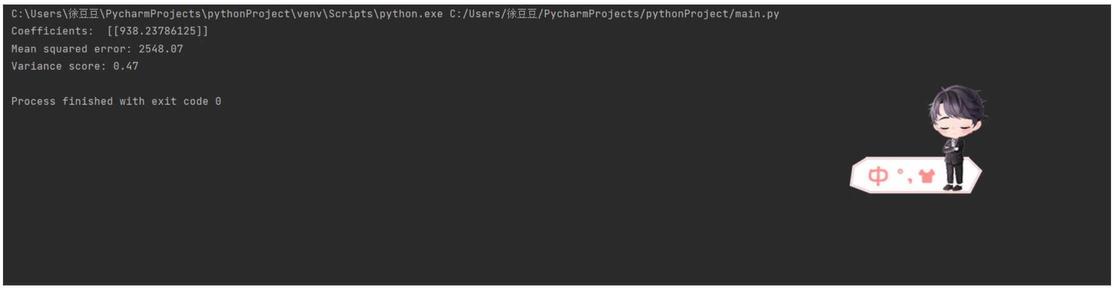
答：训练数据集转换成 1 列，numpy 根据剩下的维度计算行数。

任务 6：线性回归回归系数计算

对应关键代码段：

```
import matplotlib.pyplot as plt

import numpy as np

from sklearn.linear_model import LinearRegression

from sklearn import datasets, linear_model

from sklearn.datasets import load_diabetes

from sklearn.datasets import load_diabetes

from sklearn.metrics import mean_squared_error, r2_score

diabetes = datasets.load_diabetes()

diabetes_X = diabetes.data[:, 2]

diabetes_X_train = diabetes_X[:-20]

diabetes_X_test = diabetes_X[-20:]

diabetes_y_train = diabetes.target[:-20]

diabetes_y_test = diabetes.target[-20:]

regr = linear_model.LinearRegression()

regr.fit(np.array(diabetes_X_train).reshape(-1, 1),

np.array(diabetes_y_train).reshape(-1, 1))

diabetes_y_pred = regr.predict(np.array(diabetes_X_test).reshape(-1, 1))

print('Coefficients: ', regr.coef_)

print("Mean squared error: %.2f"
 % mean_squared_error(diabetes_y_test,diabetes_y_pred))

print('Variance score: %.2f' % r2_score(diabetes_y_test,diabetes_y_pred))
```

运行结果截图：



```
C:\Users\徐豆豆\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/徐豆豆/PycharmProjects/pythonProject/main.py
Coefficients:  [[938.23786125]]
Mean squared error: 2548.07
Variance score: 0.47

Process finished with exit code 0
```

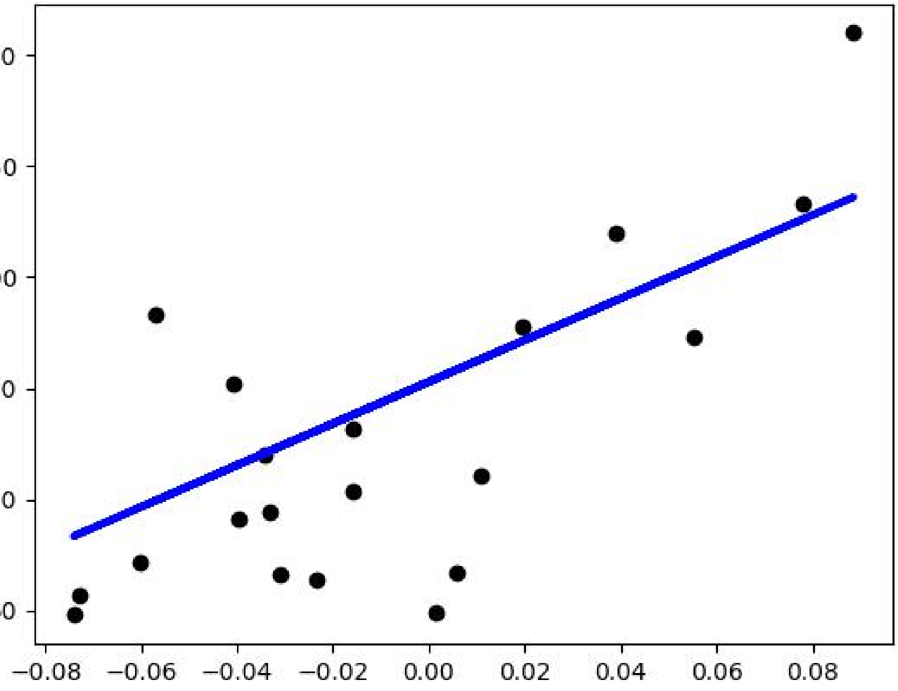
任务 7：线性回归的回归结果折线图及散点图展示

对应关键代码段：

```
import matplotlib.pyplot as plt

import numpy as np

from sklearn.linear_model import LinearRegression

from sklearn import datasets, linear_model

from sklearn.datasets import load_diabetes

from sklearn.datasets import load_diabetes

from sklearn.metrics import mean_squared_error, r2_score

diabetes = datasets.load_diabetes()

diabetes_X = diabetes.data[:, 2]

diabetes_X_train = diabetes_X[:-20]

diabetes_X_test = diabetes_X[-20:]

diabetes_y_train = diabetes.target[:-20]

diabetes_y_test = diabetes.target[-20:]

regr = linear_model.LinearRegression()
```

```
regr.fit(np.array(diabetes_X_train).reshape(-1, 1),

np.array(diabetes_y_train).reshape(-1, 1))

diabetes_y_pred = regr.predict(np.array(diabetes_X_test).reshape(-1, 1))

print('Coefficients: ', regr.coef_)

print("Mean squared error: %.2f"

 % mean_squared_error(diabetes_y_test, diabetes_y_pred))

plt.scatter(diabetes_X_test, diabetes_y_test,  color='black')

plt.plot(diabetes_X_test, diabetes_y_pred, color='blue', linewidth=3)

plt.show()
```

运行结果截图：

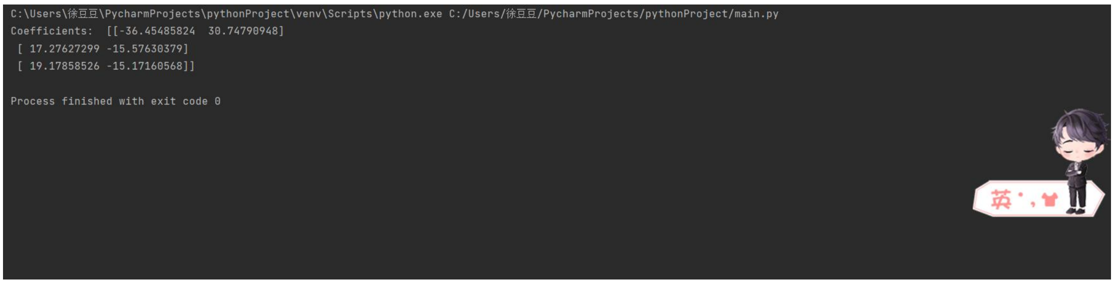


任务 8：逻辑回归回归系数计算

对应关键代码段：

```
import matplotlib.pyplot as plt

import numpy as np

from sklearn.datasets import load_iris

from sklearn.linear_model import LogisticRegression

iris = load_iris()

X = iris.data[:, :2]

Y = iris.target

lr = LogisticRegression(C=1e5)

lr.fit(X, Y)

print('Coefficients: ', lr.coef_)
```

运行结果截图：

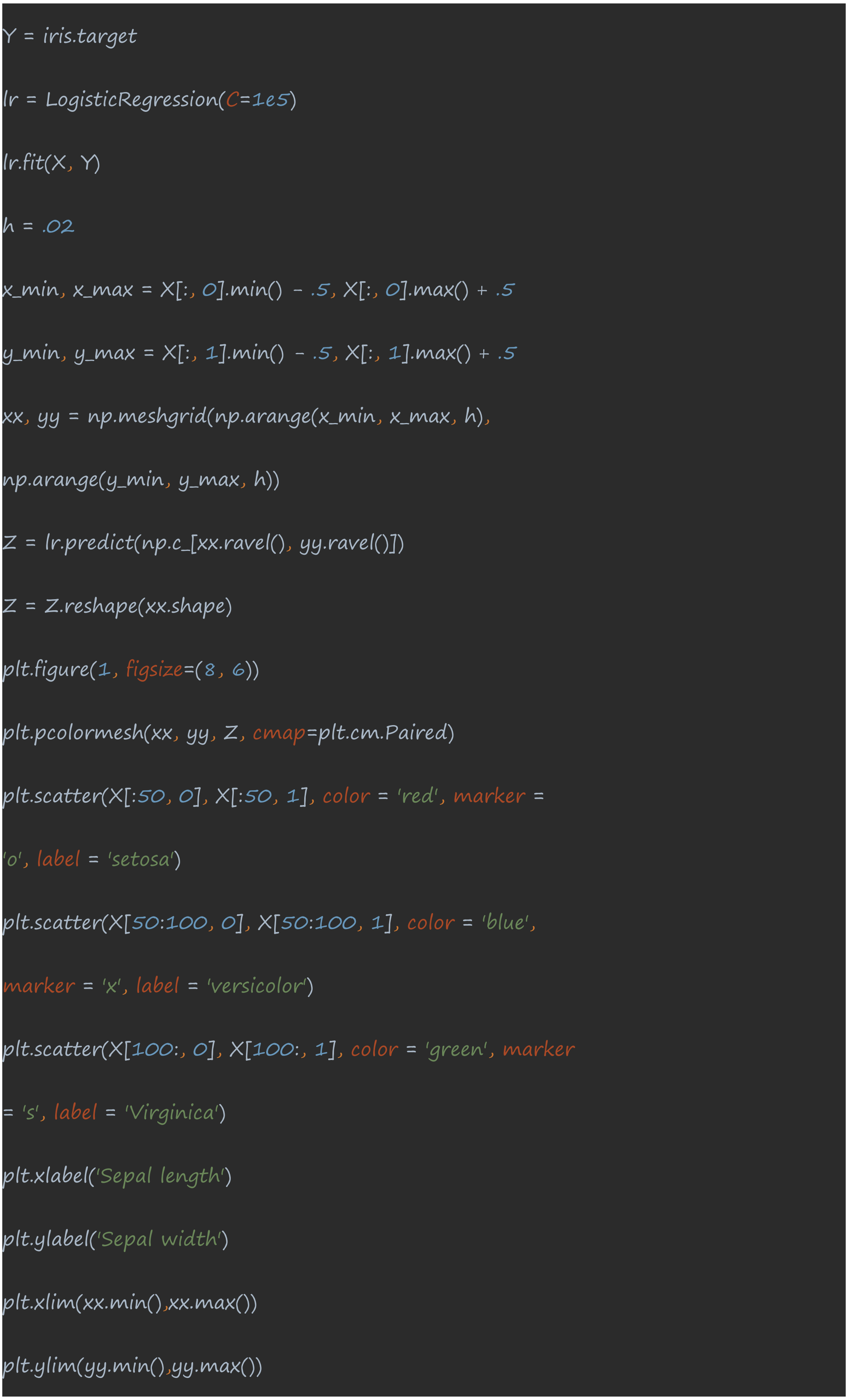


```
C:\Users\徐豆豆\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/徐豆豆/PycharmProjects/pythonProject/main.py
Coefficients:  [[-36.45485824  30.74790948]
 [ 17.27627299 -15.57630379]
 [ 19.17858526 -15.17160568]]

Process finished with exit code 0
```

任务 9：逻辑回归回归散点图展示

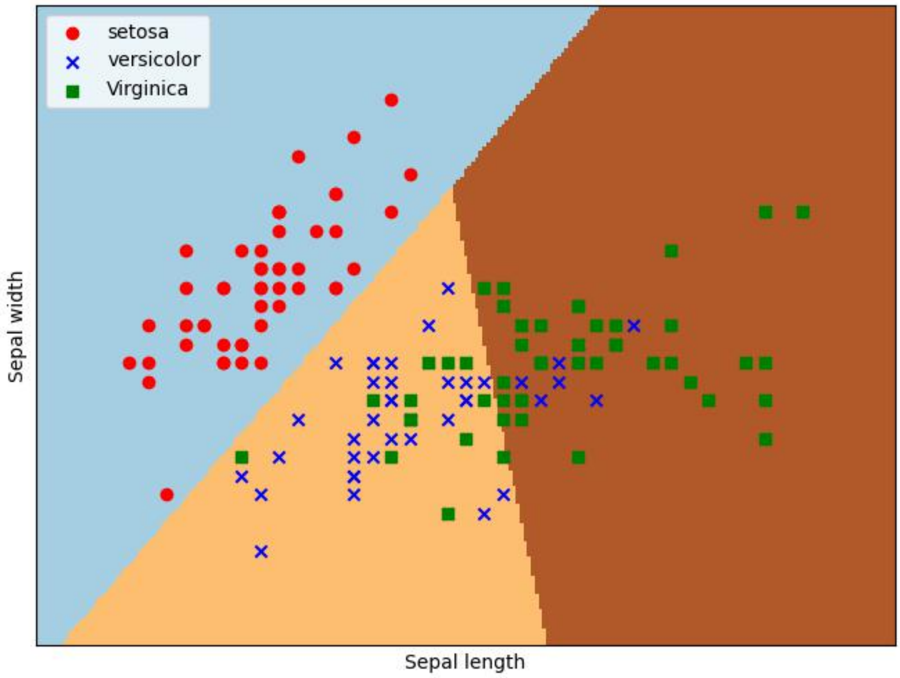
对应关键代码段：

```
import matplotlib.pyplot as plt

import numpy as np

from sklearn.datasets import load_iris

from sklearn.linear_model import LogisticRegression

iris = load_iris()

X = iris.data[:, :2]
```



```
Y = iris.target
lr = LogisticRegression(C=1e5)
lr.fit(X, Y)
h = .02
x_min, x_max = X[:, 0].min() - .5, X[:, 0].max() + .5
y_min, y_max = X[:, 1].min() - .5, X[:, 1].max() + .5
xx, yy = np.meshgrid(np.arange(x_min, x_max, h),
np.arange(y_min, y_max, h))
Z = lr.predict(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)
plt.figure(1, figsize=(8, 6))
plt.pcolormesh(xx, yy, Z, cmap=plt.cm.Paired)
plt.scatter(X[:50, 0], X[:50, 1], color = 'red', marker =
'o', label = 'setosa')
plt.scatter(X[50:100, 0], X[50:100, 1], color = 'blue',
marker = 'x', label = 'versicolor')
plt.scatter(X[100:, 0], X[100:, 1], color = 'green', marker
= 's', label = 'Virginica')
plt.xlabel('Sepal length')
plt.ylabel('Sepal width')
plt.xlim(xx.min(),xx.max())
plt.ylim(yy.min(),yy.max())
plt.xticks(())

plt.yticks(())

plt.legend(loc=2)

plt.show()
```

运行结果截图：



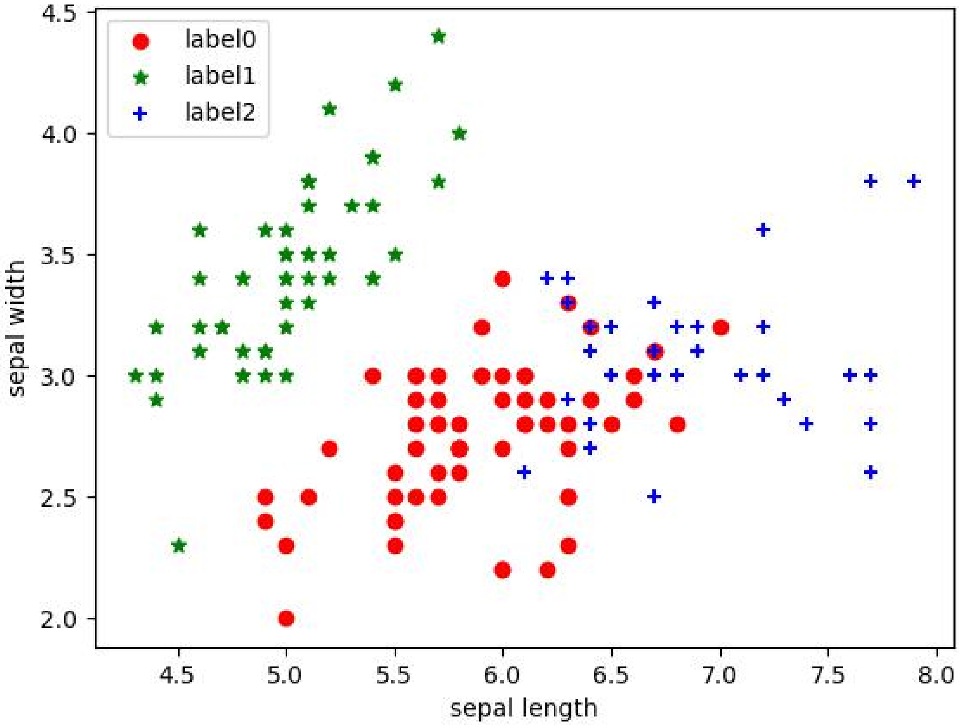
任务 10：对鸢尾花数据进行 K-means 聚类，绘制聚类中心为 3 的聚类结果图

对应关键代码段：

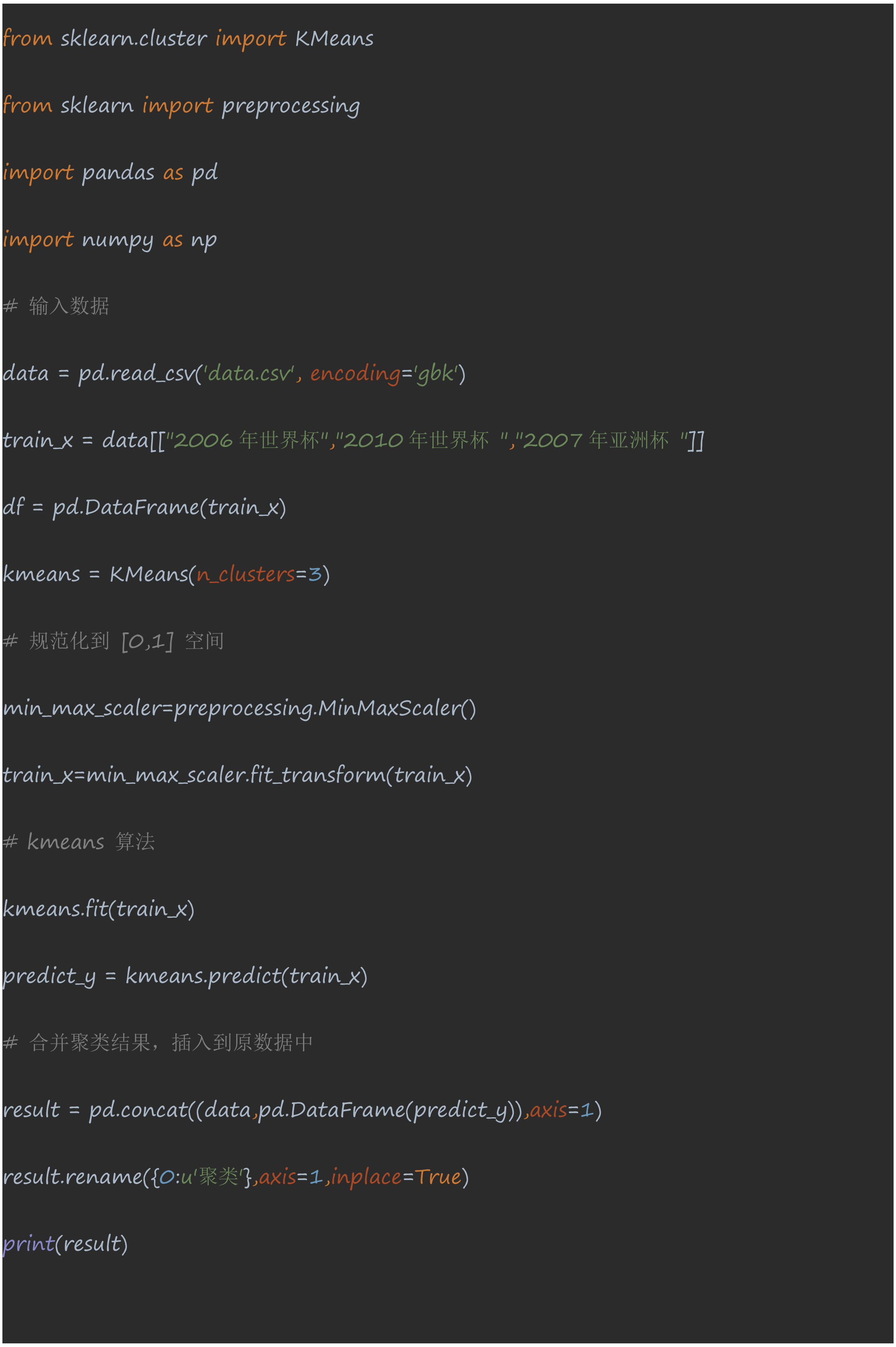
```
import matplotlib.pyplot as plt

import numpy as np

from sklearn.cluster import KMeans

from sklearn import datasets

iris = datasets.load_iris()

X = iris.data[:, :4]
```

```
print(X.shape)
plt.scatter(X[:, 0], X[:, 1], c="red", marker='o',
label='see')
plt.xlabel('sepal length')
plt.ylabel('sepal width')
plt.legend(loc=2)
plt.show()
estimator = KMeans(n_clusters=3)
estimator.fit(X)
label_pred = estimator.labels_
x0 = X[label_pred == 0]
x1 = X[label_pred == 1]
x2 = X[label_pred == 2]
plt.scatter(x0[:, 0], x0[:, 1], c="red", marker='o',
label='label0')
plt.scatter(x1[:, 0], x1[:, 1], c="green", marker='*',
label='label1')
plt.scatter(x2[:, 0], x2[:, 1], c="blue", marker='+',
label='label2')
plt.xlabel('sepal length')
plt.ylabel('sepal width')
plt.legend(loc=2)

plt.show()
```

运行结果截图：



附加题：
中国男足在亚洲水平的聚类实验
关键代码段：



```
from sklearn.cluster import KMeans
from sklearn import preprocessing
import pandas as pd
import numpy as np
# 输入数据
data = pd.read_csv('data.csv', encoding='gbk')
train_x = data[["2006年世界杯","2010年世界杯 ","2007年亚洲杯 "]]
df = pd.DataFrame(train_x)
kmeans = KMeans(n_clusters=3)
# 规范化到 [0,1] 空间
min_max_scaler=preprocessing.MinMaxScaler()
train_x=min_max_scaler.fit_transform(train_x)
# kmeans 算法
kmeans.fit(train_x)
predict_y = kmeans.predict(train_x)
# 合并聚类结果，插入到原数据中
result = pd.concat((data,pd.DataFrame(predict_y)),axis=1)
result.rename({0:u'聚类'},axis=1,inplace=True)
print(result)
```

组别	聚类	中心点
1	日本，韩国，伊朗，沙特	(0.21, 0.41, 0.16)
2	乌兹别克斯坦，巴林，朝鲜	(0.7, 0.7333, 0.4167)
3	中国，伊拉克，卡塔尔，阿联酋，泰国，越南，阿曼，印尼	(1, 0.94, 0.40625)

按照中心点位置（数值越小，排名越前），可以将聚类结果划分为三档：

亚洲一流：日本，韩国，伊朗，沙特
亚洲二流：乌兹别克斯坦，巴林，朝鲜
亚洲三流：中国，伊拉克，卡塔尔，阿联酋，泰国，越南，阿曼，印尼