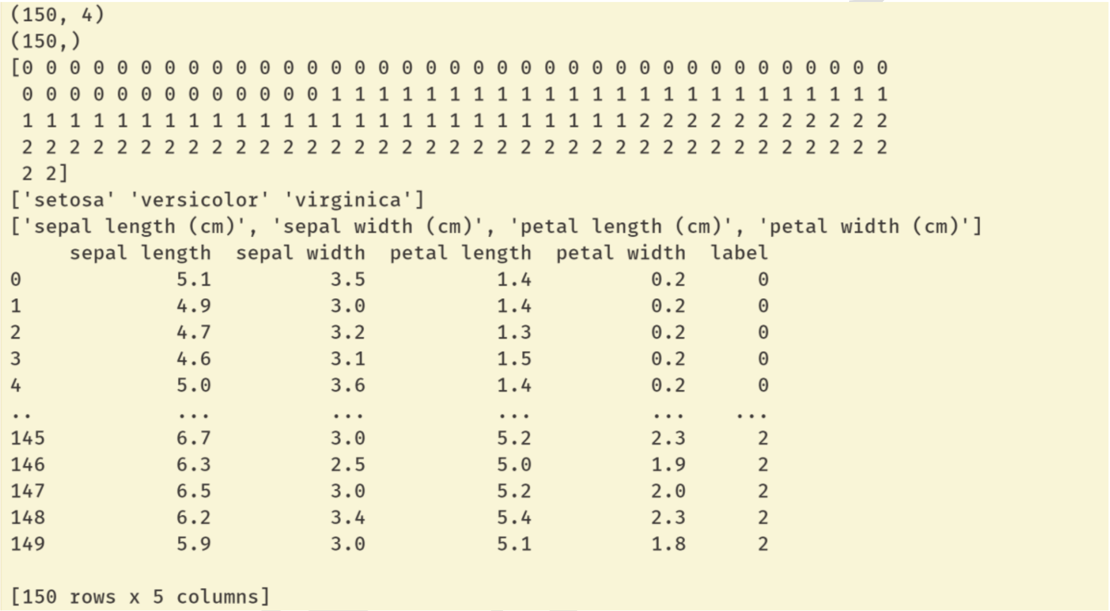
# 实验二

学号：220111028 姓名：许辰涛

## 一、 运行截图（配合截图结果，可以添加一些文字说明）

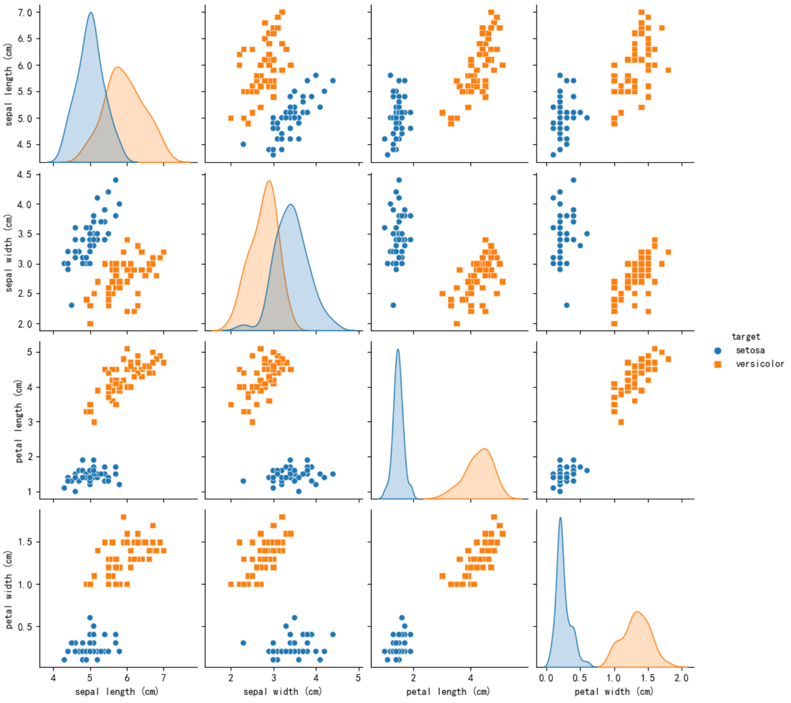
### ● 任务一



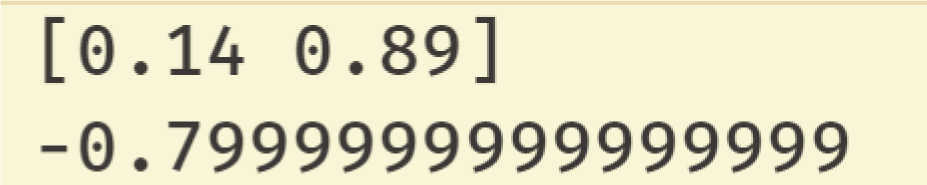
```
(150, 4)
(150,)
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2]
['setosa' 'versicolor' 'virginica']
['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']
     sepal length  sepal width  petal length  petal width  label
0             5.1          3.5           1.4          0.2      0
1             4.9          3.0           1.4          0.2      0
2             4.7          3.2           1.3          0.2      0
3             4.6          3.1           1.5          0.2      0
4             5.0          3.6           1.4          0.2      0
..            ...          ...           ...          ...    ...
145           6.7          3.0           5.2          2.3      2
146           6.3          2.5           5.0          1.9      2
147           6.5          3.0           5.2          2.0      2
148           6.2          3.4           5.4          2.3      2
149           5.9          3.0           5.1          1.8      2

[150 rows x 5 columns]
```

准备数据，打印数据的基本信息

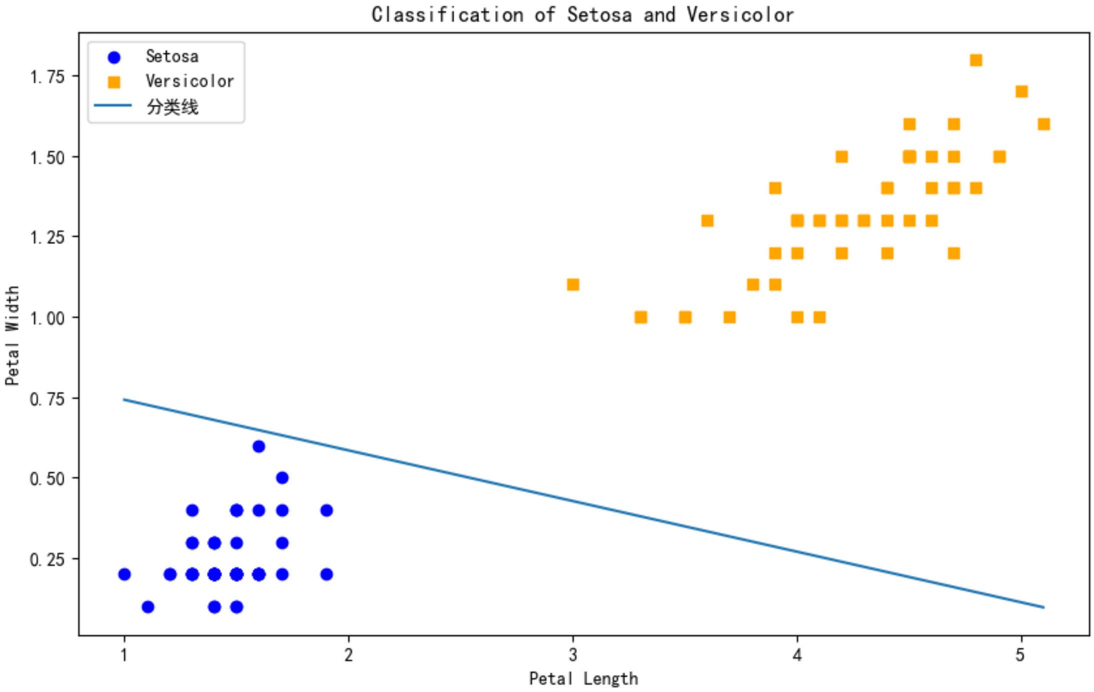


将原数据可视化，可以看出 petal length 和 petal width 两个特征在 setosa 和 versicolor 这两种花之间重合度最小，故选取这两个特征进行训练



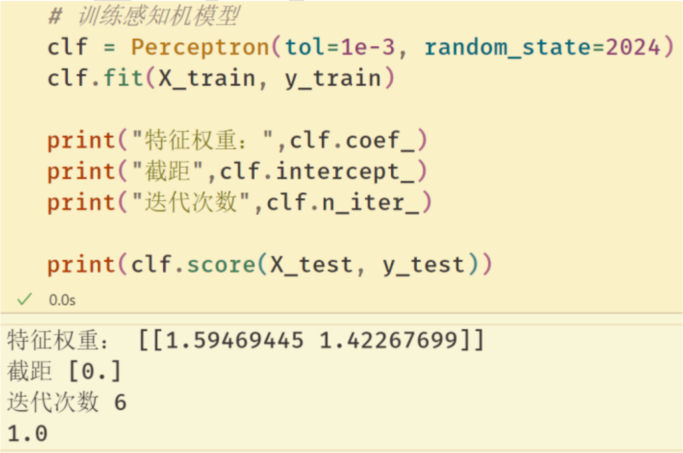
```
[0.14 0.89]
-0.7999999999999999
```

训练结果



使用得到的参数绘制分类线图像

● 任务二



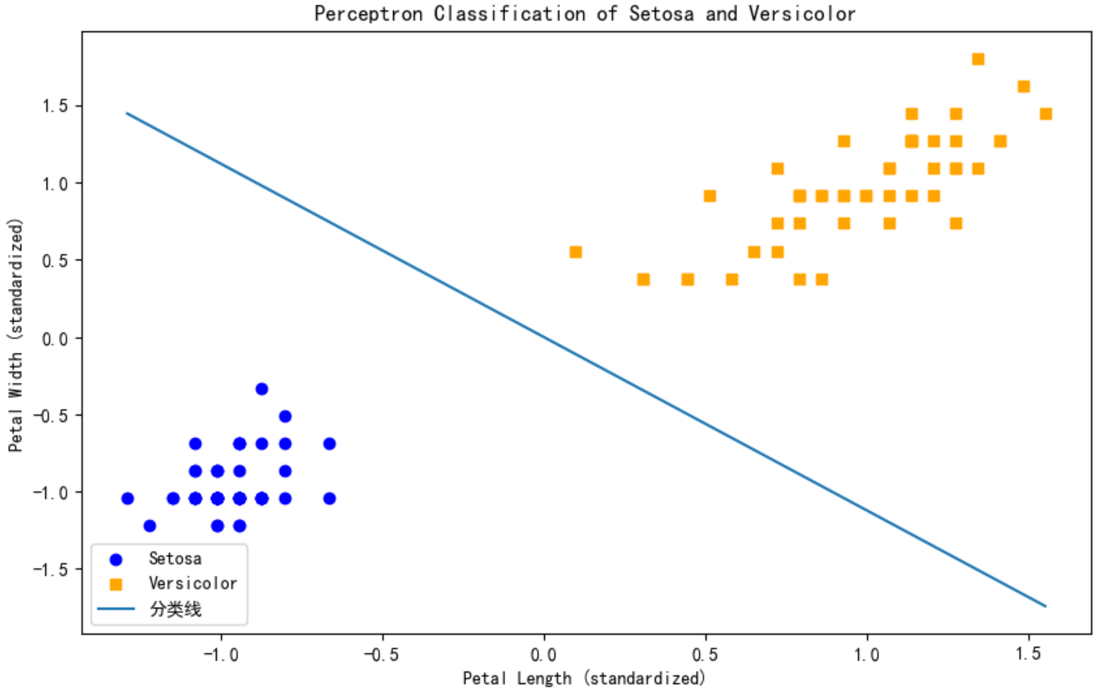
```
# 训练感知机模型
clf = Perceptron(tol=1e-3, random_state=2024)
clf.fit(X_train, y_train)

print("特征权重：",clf.coef_)
print("截距",clf.intercept_)
print("迭代次数",clf.n_iter_)

print(clf.score(X_test, y_test))
```

```
特征权重： [[1.59469445 1.42267699]]
截距 [0.]
迭代次数 6
1.0
```

准确率到达 100%



分类线图结果（特征正规化）

二、　代码（文本式粘贴，重要代码处，需添加注释）

- 任务一

```
import pandas as pd
import numpy as np
from sklearn.datasets import load_iris
import matplotlib.pyplot as plt
import seaborn as sns

plt.rcParams['font.sans-serif']=['SimHei']  #解决中文显示乱码问题
plt.rcParams['axes.unicode_minus']=False

#准备数据
iris = load_iris()
print(iris.data.shape)
print(iris.target.shape)
print(iris.target)
print(iris.target_names)
print(iris.feature_names)

df = pd.DataFrame(iris.data, columns=iris.feature_names)
df['label'] = iris.target

# 映射目标名称
df['target'] = df['label'].map({0: 'setosa', 1: 'versicolor', 2:
```

```
’virginica’})


# 使用seaborn绘制所有特征对的散点图矩阵
df_filtered = df[df[’target’].isin([’setosa’, ’versicolor’])]
sns.pairplot(df_filtered.iloc[:,[0,1,2,3,-1]], hue=’target’,
markers=[″o″, ″s″])
plt.show()


#数据准备(选取前100行 setosa和versicolor两类鸢尾花数据)
data = np.array(df_filtered.iloc[:,[2,3,4]])
X, y = data[:, :-1], data[:, -1]
for i in range(len(data)):
    if data[i,-1]==0:
        data[i,-1]=-1


# 感知机模型核心算法（随机梯度下降法）
def fit(data, X_train, y_train):
    w = np.ones(len(data[0]) - 1, dtype=np.float32)
    b = 0
    l_rate = 0.1

    # Training loop
    while True:
        error_count = 0
        for x_i, y_i in zip(X_train, y_train):
            if y_i * (np.dot(w, x_i) + b) <= 0:
                w = w + l_rate * y_i * x_i
                b = b + l_rate * y_i
                error_count += 1
        if error_count == 0:
            break

    return w, b


#调用感知机模型做，得到w和b
[w,b]=fit(data, X, y)
print(w)
print(b)
```

```
# 绘制散点图
plt.figure(figsize=(10, 6))
plt.scatter(X[y == -1][:, 0], X[y == -1][:, 1], color='blue', marker='o',
label='Setosa')
plt.scatter(X[y == 1][:, 0], X[y == 1][:, 1], color='orange', marker='s',
label='Versicolor')

# 绘制分类线
# 获取权重和偏置
x_values = np.linspace(X[:, 0].min(), X[:, 0].max(), 100)
y_values = -(w[0] * x_values + b) / w[1]
plt.plot(x_values, y_values, label="分类线")

# 设置图例和标签
plt.xlabel('Petal Length')
plt.ylabel('Petal Width')
plt.legend()
plt.title('Classification of Setosa and Versicolor')
plt.show()
```

● 任务二

```
# 导入必要的包
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.linear_model import Perceptron
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

plt.rcParams['font.sans-serif']=['SimHei']  #解决中文显示乱码问题
plt.rcParams['axes.unicode_minus']=False


# 加载数据集
iris = load_iris()
print(iris.data.shape)
print(iris.target.shape)
print(iris.target)
print(iris.target_names)
print(iris.feature_names)


# 加载 Iris 数据集
iris = load_iris()
df = pd.DataFrame(data=iris.data, columns=iris.feature_names)
df['target'] = iris.target

# 映射目标名称
df['target'] = df['target'].map({0: 'setosa', 1: 'versicolor', 2:
'virginica'})


# 使用 seaborn 绘制所有特征对的散点图矩阵
df_filtered = df[df['target'].isin(['setosa', 'versicolor'])]

sns.pairplot(df_filtered, hue='target', markers=["o", "s"])
plt.show()


# 选择最优的两个特征: Petal Length 和 Petal Width
X = df_filtered[['petal length (cm)', 'petal width (cm)']].values
y = df_filtered['target'].map({'setosa': 0, 'versicolor': 1}).values

# 标准化特征
scaler = StandardScaler()
X = scaler.fit_transform(X)

# 划分训练集和测试集
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=2024)


# 训练感知机模型
clf = Perceptron(tol=1e-3, random_state=2024)
clf.fit(X_train, y_train)

print("特征权重：",clf.coef_)
print("截距",clf.intercept_)
print("迭代次数",clf.n_iter_)

print(clf.score(X_test, y_test))
```

```
# 绘制散点图
plt.figure(figsize=(10, 6))
plt.scatter(X[y == 0][:, 0], X[y == 0][:, 1], color=’blue’, marker=’o’,
label=’Setosa’)
plt.scatter(X[y == 1][:, 0], X[y == 1][:, 1], color=’orange’, marker=’s’,
label=’Versicolor’)

# 绘制分类线
# 获取权重和偏置
w = clf.coef_[0]
b = clf.intercept_[0]
x_values = np.linspace(X[:, 0].min(), X[:, 0].max(), 100)
y_values = -(w[0] * x_values + b) / w[1]
plt.plot(x_values, y_values, label=″分类线″)

# 设置图例和标签
plt.xlabel(’Petal Length (standardized)’)
plt.ylabel(’Petal Width (standardized)’)
plt.legend()
plt.title(’Perceptron Classification of Setosa and Versicolor’)
plt.show()
```