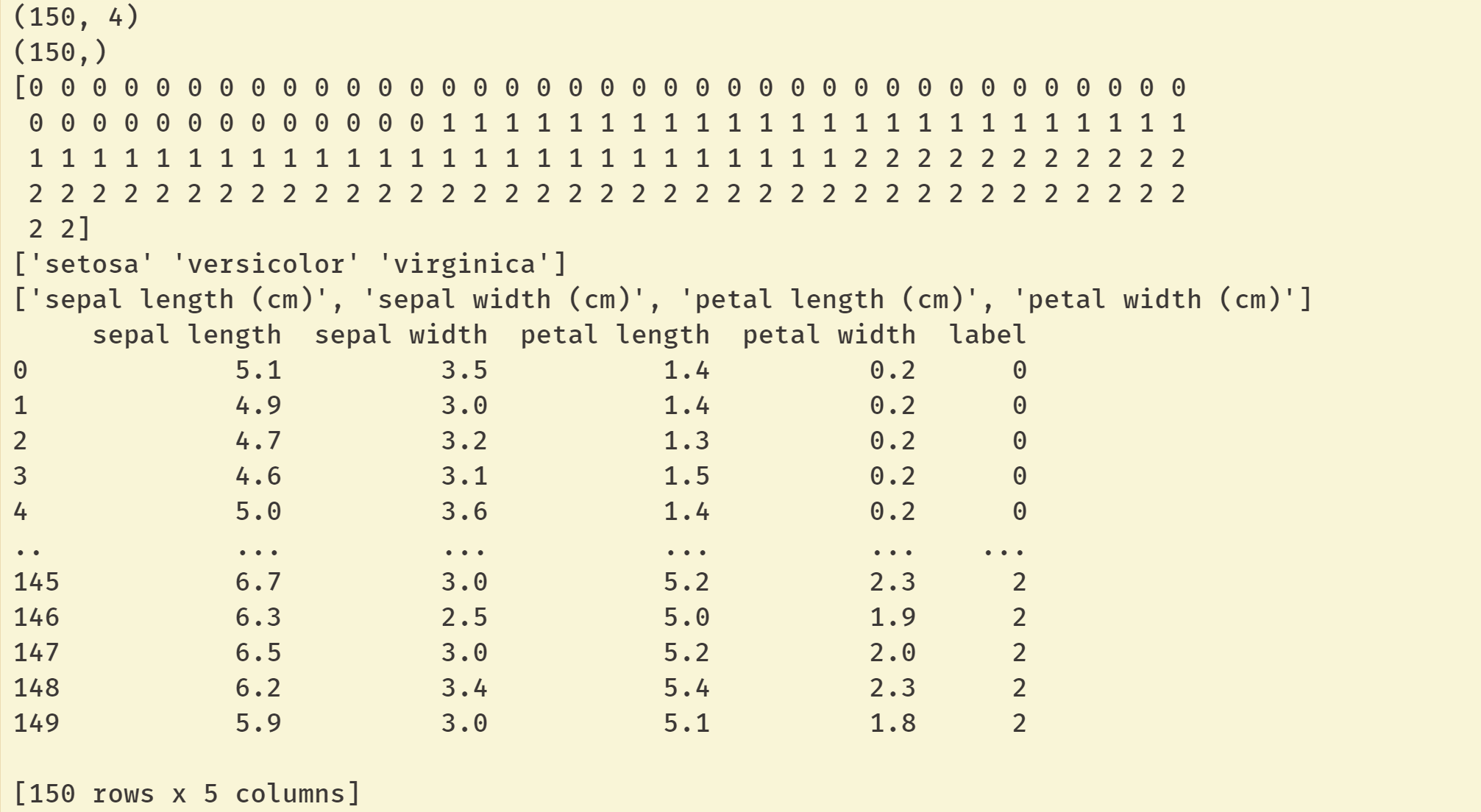
**实验二**

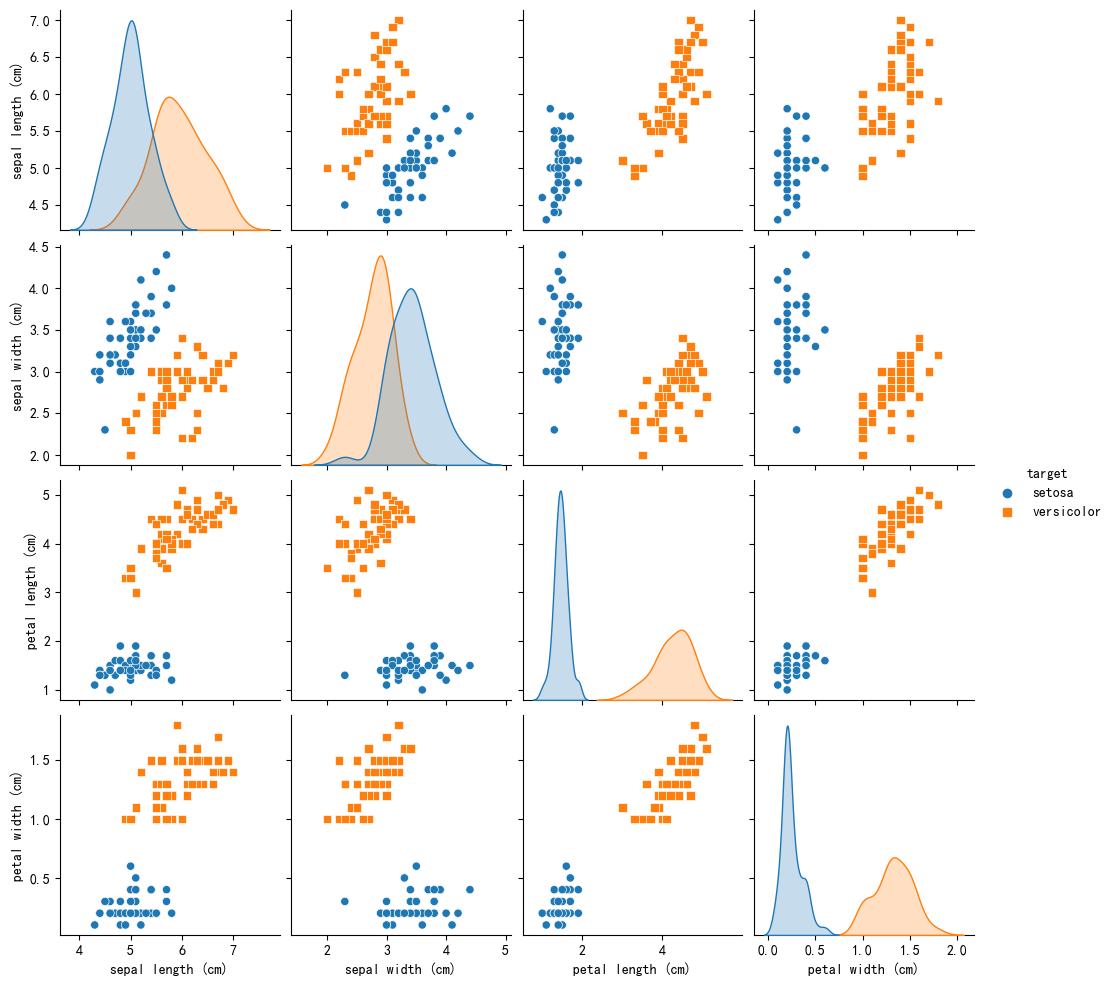
**学号： 220111028 姓名： 许辰涛**

1. **运行截图**（配合截图结果，可以添加一些文字说明）

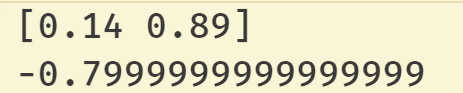
* **任务一**



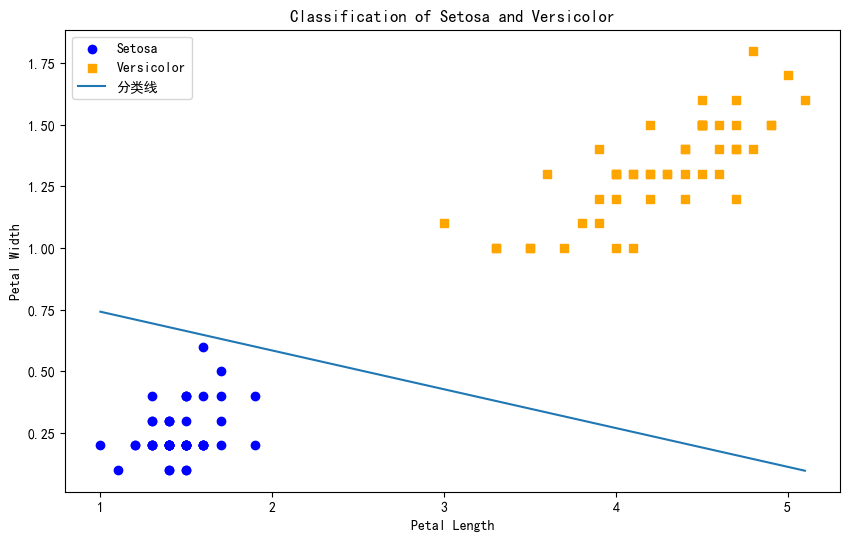
**准备数据，打印数据的基本信息**

****

**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**

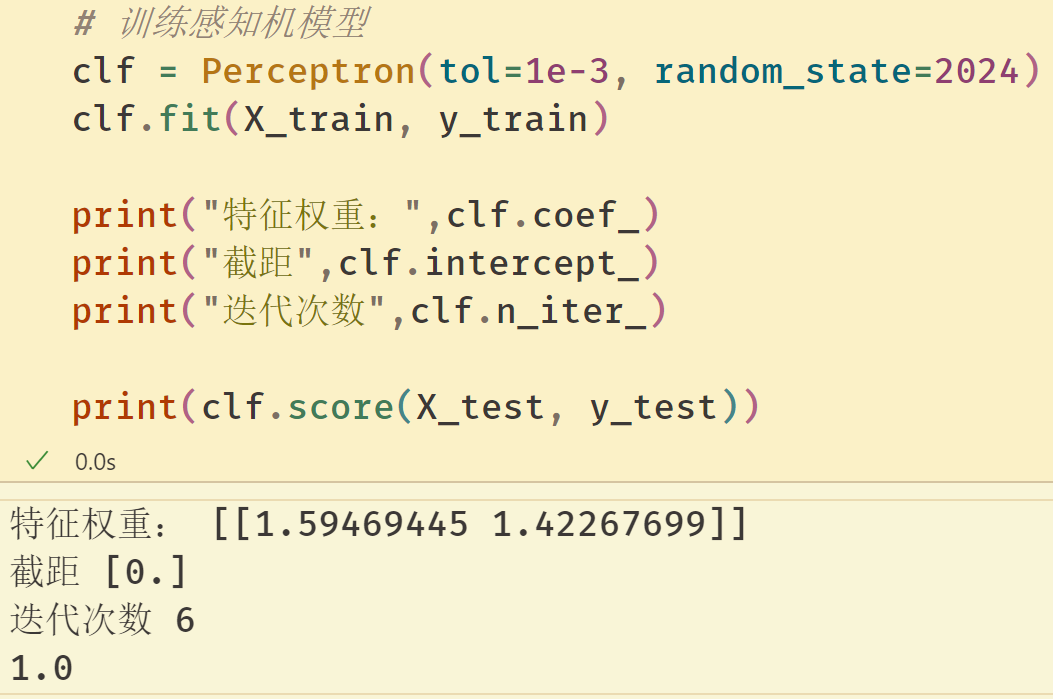
****

**训练结果**

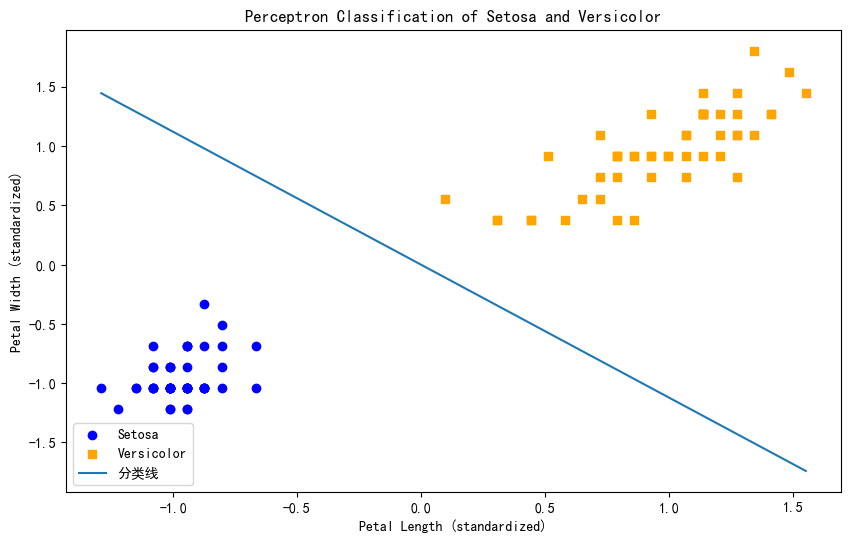
****

**使用得到的参数绘制分类线图像**

* **任务二**

****

**准确率到达100%**

****

**分类线图结果（特征正规化）**

1. **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()**