

天津大学

机器学习实验报告



题目：机器学习实验---安装

学 院 智能与计算学部
专 业 人工智能
年 级 2021
姓 名 装不起来 python 环境
学 号 你还学个球的人工智能
2023 年 2 月 23 日

机器学习实验---安装

一、实验目的

1. 熟悉运行环境。

二、实验内容

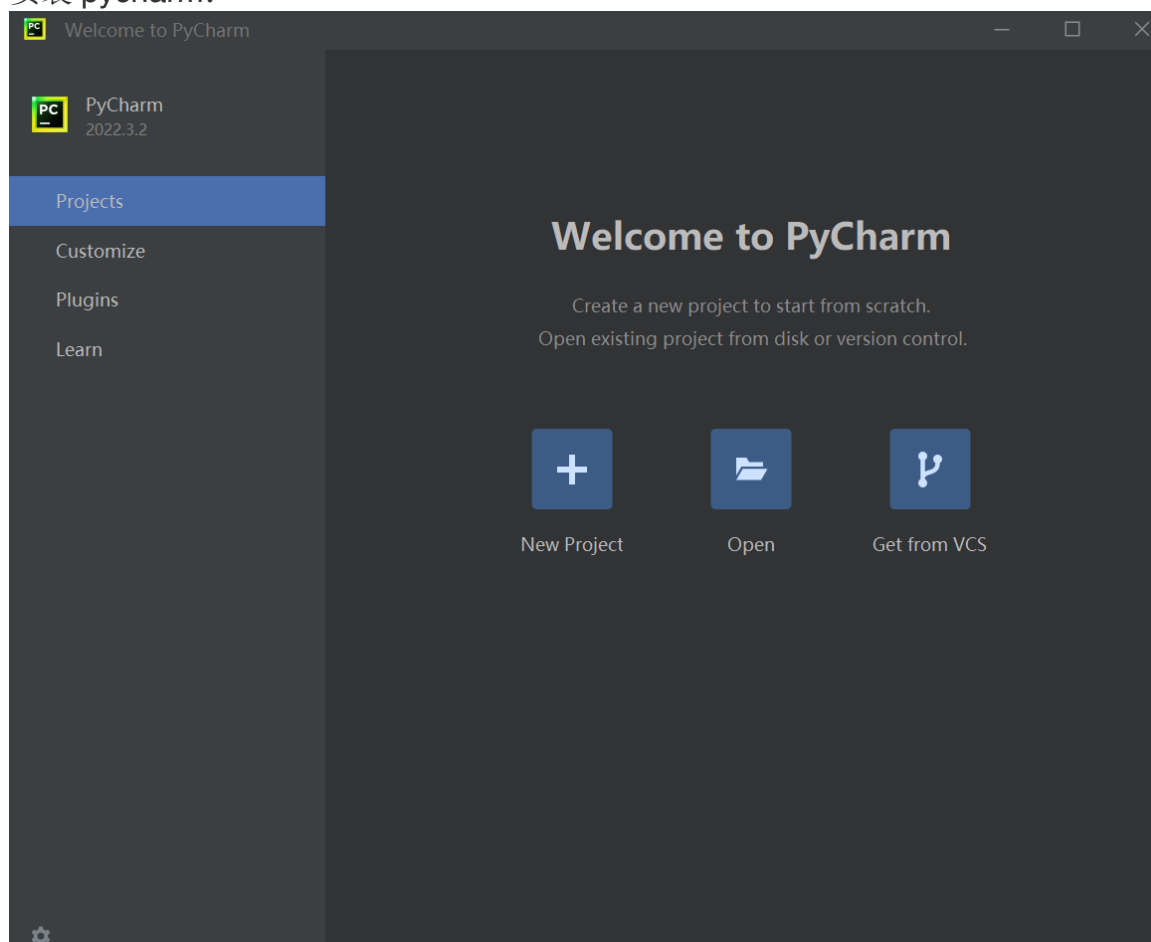
1. 安装 **Pycharm**，注册学生版。
2. 安装常见的机器学习库，如 **Scipy**、**Numpy**、**Pandas**、**Matplotlib**，**sklearn** 等。
3. 熟悉 **iris** 数据集。

三、实验报告要求

1. 按实验内容撰写实验过程；
2. 报告中涉及到的代码，每一行需要有详细的注释；
3. 按自己的理解重新组织，禁止粘贴复制实验内容。

四、实验记录

安装 **pycharm**:

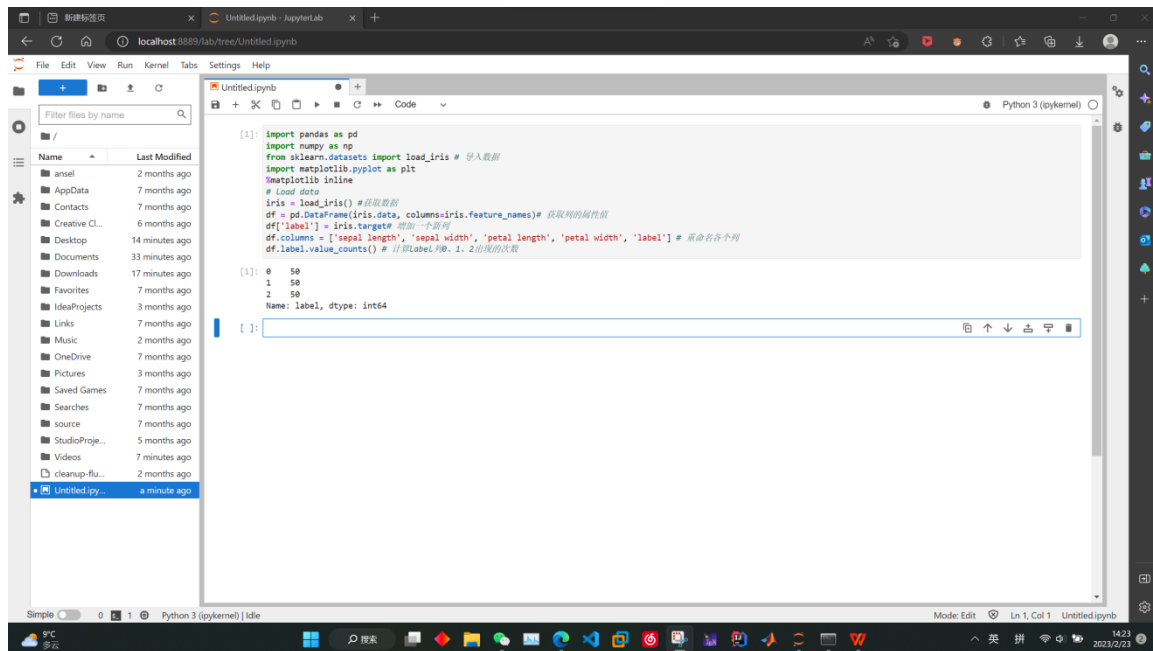


```
(base) C:\Users\zq>jupyter lab
Note: NumExpr detected 20 cores but "NUMEXPR_MAX_THREADS" not set, so enforcing safe limit of 8.
NumExpr defaulting to 8 threads.
[I 2023-02-23 14:58:57.544 ServerApp] jupyterlab | extension was successfully linked.
[I 2023-02-23 14:58:57.701 ServerApp] nbclassic | extension was successfully linked.
[I 2023-02-23 14:58:57.701 ServerApp] panel.io.jupyter_server_extension | extension was successfully linked.
[I 2023-02-23 14:58:57.716 ServerApp] nbclassic | extension was successfully loaded.
[I 2023-02-23 14:58:57.716 LabApp] JupyterLab extension loaded from D:\anaconda\lib\site-packages\jupyterlab
[I 2023-02-23 14:58:57.716 LabApp] JupyterLab application directory is D:\anaconda\share\jupyter\lab
[I 2023-02-23 14:58:57.732 ServerApp] jupyterlab | extension was successfully loaded.
[I 2023-02-23 14:58:57.732 ServerApp] panel.io.jupyter_server_extension | extension was successfully loaded.
[I 2023-02-23 14:58:57.732 ServerApp] Serving notebooks from local directory: C:\Users\zq
[I 2023-02-23 14:58:57.732 ServerApp] Jupyter Server 1.18.1 is running at:
[I 2023-02-23 14:58:57.732 ServerApp] http://localhost:8888/lab?token=ed2ef807af86e2daf53ba5fab0bb116e8088e81d4b631b08
[I 2023-02-23 14:58:57.732 ServerApp] or http://127.0.0.1:8888/lab?token=ed2ef807af86e2daf53ba5fab0bb116e8088e81d4b631b08
[I 2023-02-23 14:58:57.732 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confi
tion).
[C 2023-02-23 14:58:57.763 ServerApp]

To access the server, open this file in a browser:
file:///C:/Users/zq/AppData/Roaming/jupyter/runtime/jpserver-1276-open.html
Or copy and paste one of these URLs:
http://localhost:8888/lab?token=ed2ef807af86e2daf53ba5fab0bb116e8088e81d4b631b08
or http://127.0.0.1:8888/lab?token=ed2ef807af86e2daf53ba5fab0bb116e8088e81d4b631b08
[I 2023-02-23 14:59:00.795 LabApp] Build is up to date
```

```
import pandas as pd
import numpy as np
from sklearn.datasets import load_iris # 导入数据
import matplotlib.pyplot as plt
%matplotlib inline
# Load data
iris = load_iris() # 获取数据
df = pd.DataFrame(iris.data, columns=iris.feature_names) # 获取列的属性值
df['label'] = iris.target # 增加一个新列
df.columns = ['sepal length', 'sepal width', 'petal length', 'petal width', 'label'] # 重命名各个列
df.label.value_counts() # 计算label列0、1、2出现的次数
```

输出如下:



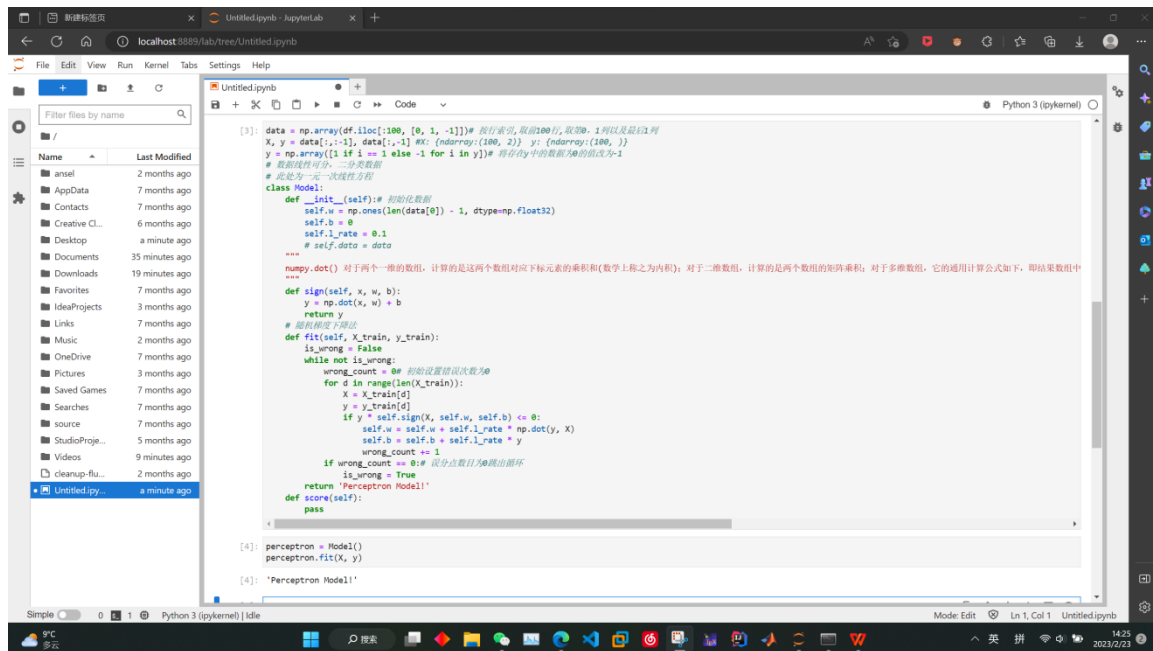
第二段内容：

```

"""
绘制散点图
"""
plt.scatter(df[:50]['sepal length'], df[:50]['sepal width'], label='0')
plt.scatter(df[50:100]['sepal length'], df[50:100]['sepal width'],
label='1')
plt.xlabel('sepal length')
plt.ylabel('sepal width')
plt.legend()
%matplotlib inline

```

输出如下:



第三段内容:

```
data = np.array(df.iloc[:100, [0, 1, -1]])  
# 按行索引,取前100行,取第0, 1列以及最后1列  
X, y = data[:, :-1], data[:, -1] #X: {ndarray:(100, 2)} y: {ndarray:(100, )}  
y = np.array([1 if i == 1 else -1 for i in y])# 将存在y中的数据为0的值改为-1  
# 数据线性可分, 二分类数据  
# 此处为一元一次线性方程
```

```
class Model:  
    def __init__(self):# 初始化数据  
        self.w = np.ones(len(data[0]) - 1, dtype=np.float32)  
        self.b = 0  
        self.l_rate = 0.1  
        # self.data = data  
    def sign(self, x, w, b):  
        y = np.dot(x, w) + b  
        return y  
    # 随机梯度下降法  
    def fit(self, X_train, y_train):  
        is_wrong = False  
        while not is_wrong:  
            wrong_count = 0# 初始设置错误次数为0  
            for d in range(len(X_train)):  
                X = X_train[d]  
                y = y_train[d]  
                if y * self.sign(X, self.w, self.b) <= 0:  
                    self.w = self.w + self.l_rate * np.dot(y, X)  
                    self.b = self.b + self.l_rate * y  
                    wrong_count += 1  
            if wrong_count == 0:# 误分点数目为0跳出循环  
                is_wrong = True
```

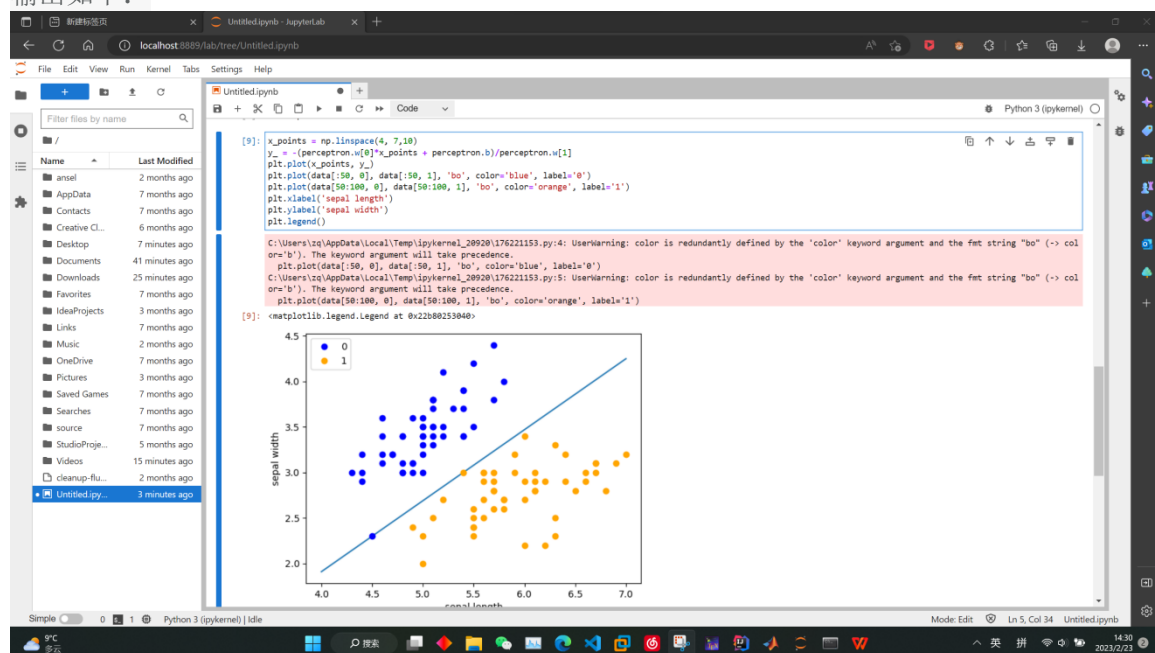
```

        return 'Perceptron Model!'
    def score(self):
        pass
perceptron = Model()
perceptron.fit(X, y)

x_points = np.linspace(4, 7, 10)
y_ = -(perceptron.w[0]*x_points + perceptron.b)/perceptron.w[1]
plt.plot(x_points, y_)
plt.plot(data[:50, 0], data[:50, 1], 'bo', color='blue', label='0')
plt.plot(data[50:100, 0], data[50:100, 1], 'bo', color='orange',
label='1')
plt.xlabel('sepal length')
plt.ylabel('sepal width')
plt.legend()

```

输出如下:



第四段内容:

```

from sklearn.linear_model import Perceptron# 使用 scikit-learn 自带的感知机模型
clf = Perceptron(fit_intercept=False, max_iter=1000, shuffle=False)# 配置导入的感知机模型
clf.fit(X, y)# 使用上面的训练数据代入模型中进行训练

# Weights assigned to the features.
print(clf.coef_)

# 截距 Constants in decision function.
print(clf.intercept_)

x_ponits = np.arange(4, 8)

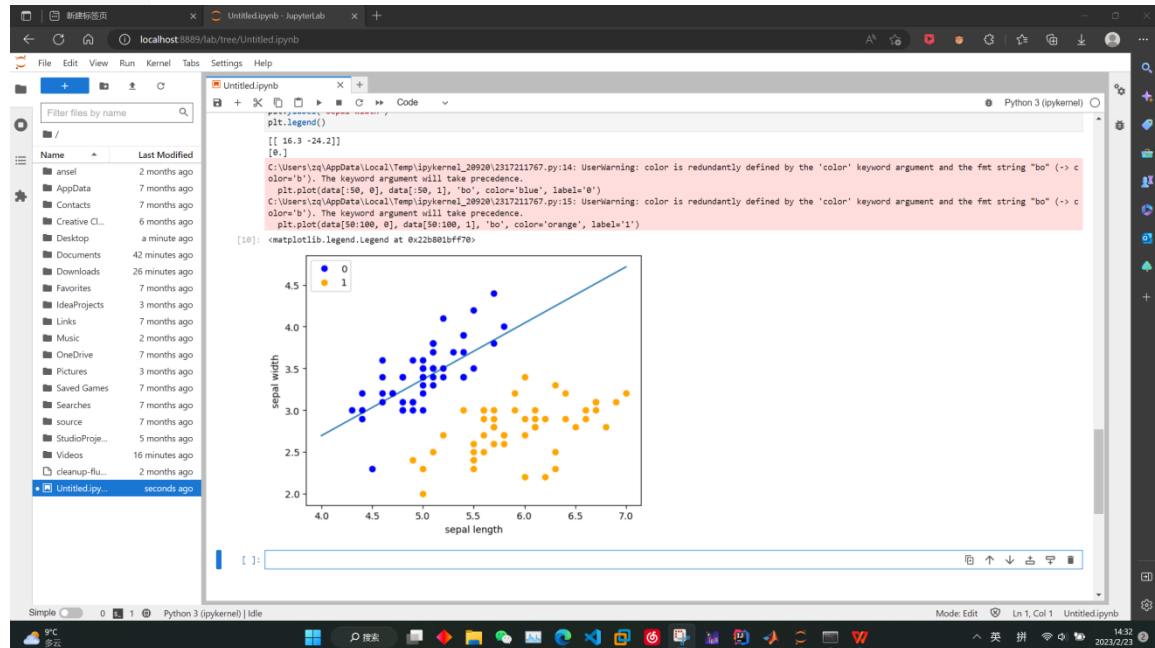
```

```

y_ = -(clf.coef_[0][0]*x_ponits + clf.intercept_)/clf.coef_[0][1]
plt.plot(x_ponits, y_)
plt.plot(data[:50, 0], data[:50, 1], 'bo', color='blue', label='0')
plt.plot(data[50:100, 0], data[50:100, 1], 'bo', color='orange',
label='1')
plt.xlabel('sepal length')
plt.ylabel('sepal width')
plt.legend()

```

输出如下:



五、运行结果

```
File Edit View Run Kernel Tabs Settings Help
localhost:8888/lab/tree/Untitled.ipynb

Filter files by name

Name Last Modified
ansel 2 months ago
AppData 7 months ago
Contacts 7 months ago
Creative CL 6 months ago
Desktop a minute ago
Documents 35 minutes ago
Downloads 19 minutes ago
Favorites 7 months ago
IdeaProjects 3 months ago
Links 7 months ago
Music 2 months ago
OneDrive 7 months ago
Pictures 3 months ago
Saved Games 7 months ago
Searches 7 months ago
source 7 months ago
StudioProje... 5 months ago
Videos 9 minutes ago
cleanup-flu... 2 months ago
Untitled.ipynb a minute ago

[3]: data = np.array(df.iloc[:100, [0, 1, -1]])# 按行索引,取前100行,取第0, 1列以及最后一列
X, y = data[:, :-1], data[:, -1] #X: (ndarray:(100, 2)) y: (ndarray:(100,))
y = np.array([1 if i == 1 else -1 for i in y])# 将存y中的数据为0的值改为-1
# 数据特征可分, 一维数据
# 此处为"一元"非线性方程

class Model:
    def __init__(self):# 初始化数据
        self.w = np.ones(len(data[0]) - 1, dtype=np.float32)
        self.b = 0
        self.l_rate = 0.1
        # self.data = data

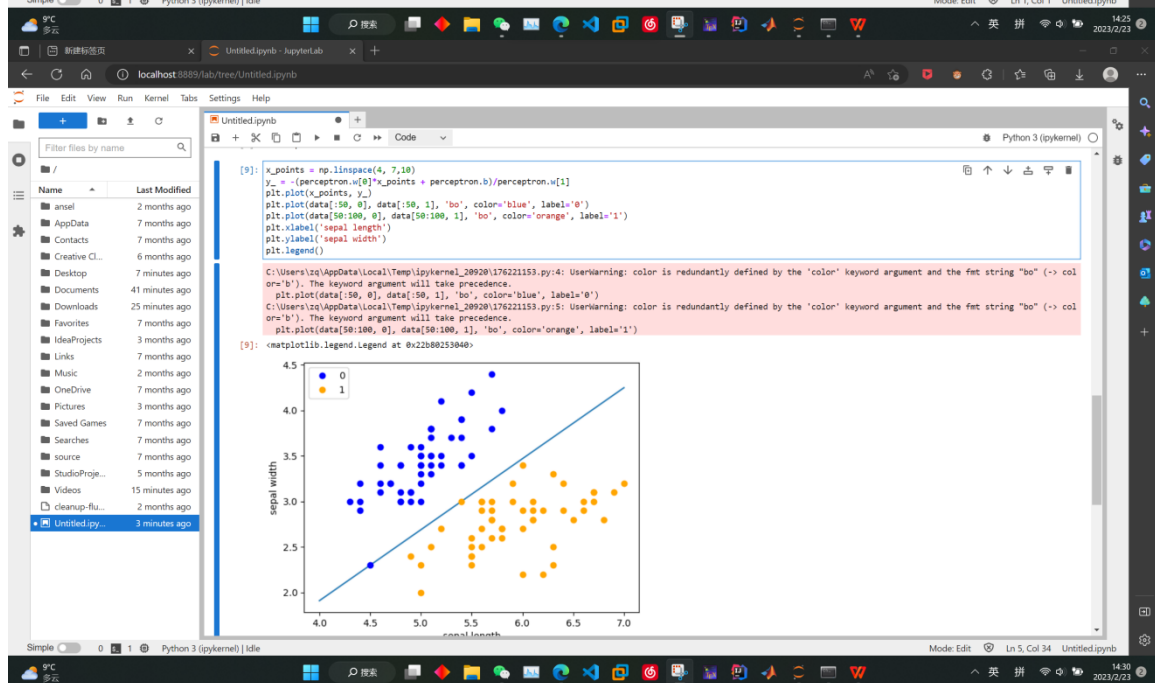
    def sign(self, x, w, b):
        y = np.dot(x, w) + b
        return y

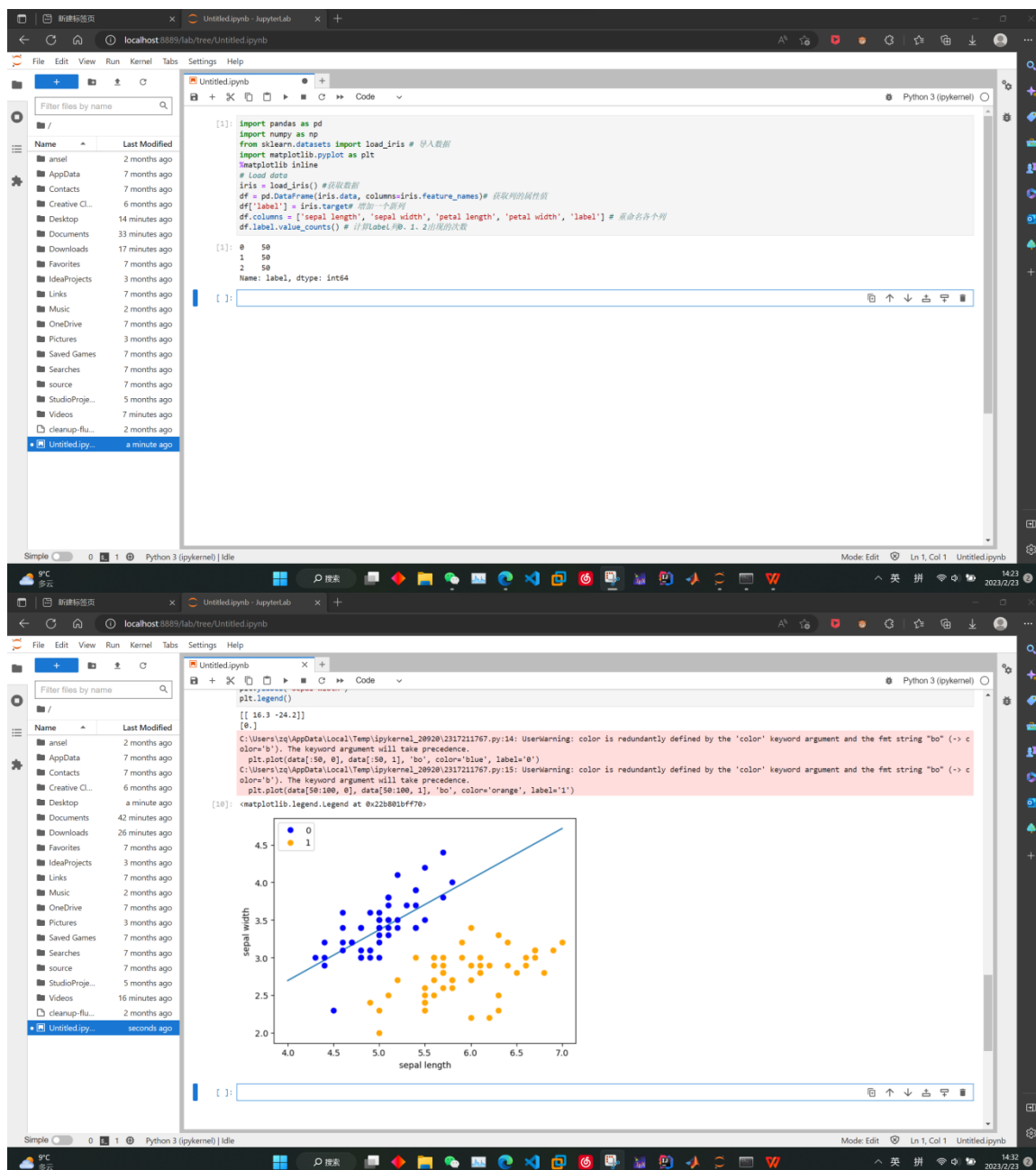
    def fit(self, X_train, y_train):
        is_wrong = False
        while not is_wrong:
            wrong_count = 0# 初始设置错误次数为0
            for d in range(len(X_train)):
                X = X_train[d]
                y = y_train[d]
                if y * self.sign(X, self.w, self.b) <= 0:
                    self.w = self.w + self.l_rate * np.dot(y, X)
                    self.b = self.b + self.l_rate * y
                    wrong_count += 1
            if wrong_count == 0:# 没分点数目为0跳出循环
                is_wrong = True
            return 'Perceptron Model!'

    def score(self):
        pass

[4]: perceptron = Model()
perceptron.fit(X, y)

[4]: 'Perceptron Model!'
```





六、实验小结

本次实验是理解感知机算法的原理并实现感知机算法，感知机称为单层感知机模型，其输入是实例的特征向量，输出为实例的分类类别。它是一种使用阶梯函数激活的人工神经元，以产生二分类输出，用于将数据分为两部分，因此也称为线性二分类器。实验中使用 jupyterbook 进行实验，并使用到了 pandas、numpy、Matplotlib、sklearn 等机器学习库，可对机器学习有初步理解。