**实验一**

**学号： 200810216 姓名： 李墨**

1. **运行截图**（注意截图清晰）

**截图内容包含：**

1. **代码**

**任务一代码：**

**import pandas as pd**

**import numpy as np**

**import matplotlib.pyplot as plt**

**from sklearn.datasets import load\_iris**

**from sklearn.linear\_model import Perceptron**

**from sklearn.model\_selection import train\_test\_split**

**def create\_df():**

**iris = load\_iris()**

**#将鸢尾花4个特征，以4列存入pandas的数据框架**

**df = pd.DataFrame(iris.data, columns=iris.feature\_names)**

**#在最后一列追加加入（目标值〉列数据,之前是没有label标签这一列的，他是新加入的**

**df[ 'label'] = iris.target**

**df.columns = [ 'sepal length', 'sepal width','petal length','petal width', 'label']**

**return df**

**def show\_image(df):**

**plt.scatter(df[:50]['sepal length'],df[:50]['sepal width'],label='setosa')**

**plt.scatter(df[100:150]['sepal length'],df[100:150]['sepal width'],label='virginica')**

**plt.xlabel('sepal length')**

**plt.ylabel('sepal width')**

**plt.legend()**

**plt.show()**

**def data\_slice(df):**

**#.选取数据，前50行和100到150行,.前两个特征，最后一列的目标值**

**data1 = df.iloc[:50,[0,1,-1]]**

**data2 = df.iloc[100:150,[0,1,-1]]**

**data\_all = [data1,data2]**

**data = np.array(pd.concat(data\_all))**

**#X装的是两列特征，y装的是标签，答案**

**X,y = data[:,:-1],data[:,-1]**

**#生成感知机的标签值，+1， -1，第一种 -1，第二种+1**

**for i in range(len(data) ):**

**if data[i,-1] == 0:**

**data[i,-1] = -1**

**if data[i,-1] == 2:**

**data[i,-1] = 1**

**#这里换了花朵之后要记得改**

**return [data,X,y]**

**#数据线性可分，二分类数据**

**class Model:**

**def \_\_init\_\_(self,data):**

**self.w = np.ones(len(data[0])-1,dtype=np.float32)**

**self.b = 0**

**self.l\_rate = 0.1**

**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, v\_train):**

**signal=0**

**while(signal==0):**

**signal=1**

**for xi,yi in zip(x\_train,v\_train):**

**if yi\*(np.dot(self.w,xi)+self.b)<=0:**

**self.w = self.w + self.l\_rate\*yi\*xi**

**self.b = self.b+self.l\_rate\*yi**

**signal=0**

**def show(data,perceptron):**

**x\_points = np.linspace(4,7,10)#在4到7之间造10个点，包含4与7**

**y\_ = -(perceptron.w[0] \* x\_points + perceptron.b) / perceptron.w[1]**

**plt.plot(x\_points, y\_,label = 'line')**

**plt.plot(data[:50,0],data[:50,1],'bo',color='blue',label='setosa')**

**plt.plot(data[50:100,0],data[50:100,1],'bo', color='orange',label='virginica')**

**plt.xlabel('sepal length')**

**plt.ylabel('sepal width')**

**plt.legend()**

**plt.show()**

**if \_\_name\_\_=='\_\_main\_\_':**

**#加载训练集**

**df = create\_df()**

**#可视化数据集**

**show\_image(df)**

**#数据切片**

**[data, X, y] = data\_slice(df)**

**#构造感知机对象，对数据集进行训练，得出模型参数**

**perceptron = Model(data)**

**perceptron.fit(X, y)**

**#绘制图像**

**show(data,perceptron)**

**任务二代码：**

**import pandas as pd**

**import numpy as np**

**import matplotlib.pyplot as plt**

**from sklearn.datasets import load\_iris**

**from sklearn.linear\_model import Perceptron**

**from sklearn.model\_selection import train\_test\_split**

**def create\_df():**

**iris = load\_iris()**

**#将鸢尾花4个特征，以4列存入pandas的数据框架**

**df = pd.DataFrame(iris.data, columns=iris.feature\_names)**

**#在最后一列追加加入（目标值〉列数据,之前是没有label标签这一列的，他是新加入的**

**df[ 'label'] = iris.target**

**df.columns = [ 'sepal length', 'sepal width','petal length','petal width', 'label']**

**return df**

**def show\_image(df):**

**plt.scatter(df[:50]['sepal length'],df[:50]['sepal width'],label='setosa')**

**plt.scatter(df[100:150]['sepal length'],df[100:150]['sepal width'],label='virginica')**

**plt.xlabel('sepal length')**

**plt.ylabel('sepal width')**

**plt.legend()**

**plt.show()**

**def create\_data(df):**

**#.选取数据，前50行和100到150行,.前两个特征，最后一列的目标值**

**data1 = df.iloc[:50,[2,1,-1]]**

**data2 = df.iloc[100:150,[2,1,-1]]**

**data\_all = [data1,data2]**

**data = np.array(pd.concat(data\_all))**

**#生成感知机的标签值，+1， -1，第一种 -1，第二种+1**

**for i in range(len(data) ):**

**if data[i,-1] == 0:**

**data[i,-1] = -1**

**if data[i,-1] == 2:**

**data[i,-1] = 1**

**#这里换了花朵之后要记得改**

**return data**

**def show(clf,data):**

**x\_points = np.arange(0,8)**

**y\_ = -(clf.coef\_[0][0] \* x\_points + clf.intercept\_) / clf.coef\_[0][1]**

**plt.rcParams['font.sans-serif'] = ['SimHei']**

**plt.plot(x\_points, y\_, 'r', label='sklearn Perceptron Line')**

**plt.plot(data[:50,0],data[:50,1],'bo', color='blue', label='setosa')**

**plt.plot(data[50:100,0], data[50:100,1],'bo',color='orange',label='virginica')**

**plt.xlabel( 'petal length(cm) ' )**

**plt.ylabel( 'sepal width(cm) ' )**

**plt.title( 'Iris Perceptron classifier',fontsize=15)**

**plt.legend()**

**plt.show()**

**if \_\_name\_\_ == '\_\_main\_\_' :**

**#加载数据集**

**df = create\_df()#原数据可视化**

**show\_image(df)#数据切片**

**data=create\_data(df)#数据分割**

**print(data)**

**#X装的是两列特征，y装的是标签，答案**

**X,y = data[:,:-1],data[:,-1]**

**X\_train,X\_test,y\_train,y\_test = train\_test\_split(X,#被划分的样本特征集**

**y,#被划分的样本目标集**

**test\_size=0.3,#测试样本占比**

**random\_state=2)#随机数种子**

**#定义感知机**

**clf = Perceptron(fit\_intercept=False,max\_iter=100000000,shuffle=True)**

**#使用训练数据进行训练**

**clf.fit(X\_train, y\_train)**

**#计算模型的权重、截距迭代次数**

**print("特征权重: ",clf.coef\_)**

**#特征权重w**

**print("截距（偏置):",clf.intercept\_)**

**#截距b**

**print("迭代次数:",clf.n\_iter\_)**

**#评价模型**

**print(clf.score(X\_test, y\_test))**

**#绘制图形:观察分类结果**

**show(clf, data)**

1. **运行结果（权重+截距+准确率+带分类线的图）**

**例如：**

文本

描述已自动生成

图表, 散点图

描述已自动生成

1. **解释下为什么选这两个特征做分类（用图片或是文字）**

图表, 散点图

描述已自动生成

**在['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']四个特征中选择两个特征时，选择这两个特征'petal length (cm)', 'sepal width (cm)'作为自变量和因变量能够看出，setosa与virginica在这种条件是明显可通过直线分开的,倘若选择sepal length和sepal width，则用sklearn中的模型是分不开这些的。**

**如下图所示：**

图表, 散点图

描述已自动生成

1. **其他（附加题或者选做内容）**
2. 在做数据处理时，为什么要转化为dataFrame格式来处理，不能直接用 numpy吗

**答：.**NumPy中的ndarray用于处理多维数值型数组，重点在于进行数值运算，**无索引**。Pandas中的DataFrame中的每一列都可以看作是一个Series，**有索引，方便进行数据的查询，筛选**， Pandas重点在于进行数据分析。

在数学与统计方法上，NumPy中的ndarray只能进行**数值型统计**，而Pandas中的DataFrame既可以进行数值型，也可以进行非数值型统计。基于可以容纳不同的数据类型而定

1. 怎样挑选合适的特征来做分类，理由是什么？

应该选取能够通过超平面来将不同类区分开的特征来做分类，理由是如果选取了不合适的特征，可能根本不存在这样的超平面来将两类分开，感知机无法做到二分类。

1. 为什么要使用随机种子做数据分割？

防止每一次的实验都不一样，所以固定一个随机种子，那样每次实验是可重复的，可以暂时排除掉随机因素带来的影响。

1. 使用sklearn库来编码，学习率对迭代过程和最终结果有无影响？若有/无 影响的话，条件是什么

文本

描述已自动生成我试着把sklearn中的eta0参数改的特别特别小，最后得出的特征权重也变得很小。

eta0=0.00000000001

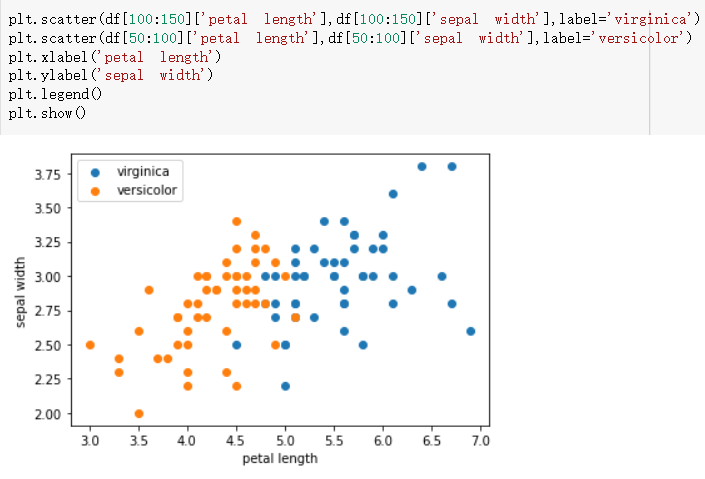
文本, 白板

描述已自动生成当我将eta0调到1时候，迭代次数多了一次。

感觉影响不大，特征权重只是按照倍数的伸缩了而已。

关于条件，暂时还没研究出来。

1. 本次实验能对versicolor 和virginica两种鸢尾花做分类吗？能的话，实现 出来；不能的话，说明理由。



做不出来。我尝试了不同种变量

图表, 散点图

描述已自动生成

下面这种特征选择方法是看起来最能够二分类的

图表, 散点图

描述已自动生成

然而当我尝试对其完全分类，程序陷入了死循环。无法实现对所有点的完全分类。

表格

描述已自动生成