上海理工大学光电信息与计算机工程学院

**《数据挖掘》实验报告**

****

**专　　业 智能科学与技术**

**姓 名　　 徐馨怡**

**学　 号 2035060404**

**年　　级 2020级**

**指导教师 孙占全**

**成 绩：**

**教师签字：**

# 实验一 数据预处理与可视化

一、实验目的和要求

1. 掌握数据预处理方法，如缺失值修复、数据标准化等。

2. 掌握数据挖掘中的分类聚类方法和分类性能的评价估计

3. 熟悉ROC和PR曲线评估结果的计算过程

4. 以鸢尾花数据集为例（'http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data），学会使用python做去除缺失数据记录或用均值补全缺失数据、做数据标准化、萼片和花瓣长宽属性10等分的直方图、萼片和花瓣长宽属性箱式图、萼片和花瓣长宽属性两两之间散点图

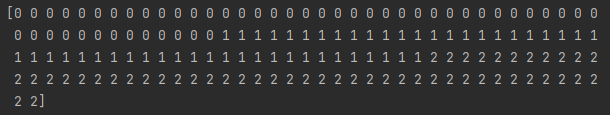
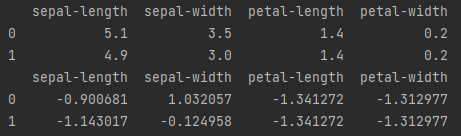
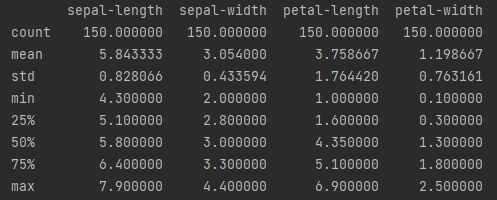
1. 基于鸢尾花数据集，利用决策数分类器进行分类，画出测试结果的ROC和PR曲线。

二、实验算法

import matplotlib.pyplot as plt  
import numpy as np  
from sklearn.datasets import load\_iris  
from sklearn.preprocessing import StandardScaler  
import pandas  
import pandas as pd  
import graphviz  
#from sklearn.tree import export\_graphviz  
from sklearn import tree  
  
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"  
names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'class']  
dataset = pandas.read\_csv(url, names=names)# 读取csv数据  
print(dataset.describe())  
# 数据标准化  
features = dataset.iloc[:, :-1]  
print(features.head(2))  
scaler = StandardScaler()  
features\_new = scaler.fit\_transform(features)  
features\_new = pd.DataFrame(features\_new, columns=names[:-1])  
print(features\_new.head(2))  
  
# 散点图  
iris = load\_iris()# 载入鸢尾花数据集  
datas = iris.data# (150,4)：150 行 5 列  
print(datas)  
print(iris.target)  
x1 = [x[0] for x in datas]# 第1列：花萼长度 sepal\_length  
x2 = [x[1] for x in datas]# 第2列：花萼宽度 sepal\_width  
x3 = [x[2] for x in datas]# 第3列：花瓣长度 petal\_length  
x4 = [x[3] for x in datas]# 第4列：花瓣宽度 petal\_width  
plt.scatter(x1[:50], x2[:50], color='red', marker='o', label='setosa') # 前50个样本  
plt.scatter(x1[50:100],x2[50:100],color='blue',marker='x',label='vericolor')# 中间50个样本  
plt.scatter(x1[100:150],x2[100:150],color='green',marker='+',label='Virginica')# 后50个样本  
plt.legend(loc=1)# loc=1，2，3，4分别表示label在右上角，左上角，左下角，右下角  
plt.show()  
plt.scatter(x3[:50], x4[:50], color='red', marker='o', label='setosa') # 前50个样本  
plt.scatter(x3[50:100],x4[50:100],color='blue',marker='x',label='vericolor')# 中间50个样本  
plt.scatter(x3[100:150],x4[100:150],color='green',marker='+',label='Virginica')# 后50个样本  
plt.legend(loc=2)# loc=1，2，3，4分别表示label在右上角，左上角，左下角，右下角  
plt.show()  
  
# 单个直方图  
# sepal\_length 直方图  
plt.hist(x1)  
plt.xlabel("sepal\_length")  
plt.ylabel("sepal\_length")  
plt.show()  
# sepal\_width 直方图  
plt.hist(x2)  
plt.xlabel("sepal\_width")  
plt.ylabel("sepal\_width")  
plt.show()  
# petal\_length 直方图  
plt.hist(x3)  
plt.xlabel("petal\_length")  
plt.ylabel("petal\_length")  
plt.show()  
# petal\_width 直方图  
plt.hist(x4)  
plt.xlabel("petal\_width")  
plt.ylabel("petal\_width")  
plt.show()  
data\_np = np.array(datas)  
labels = ['sepal length', 'sepal width', 'petal length', 'petal width']  
plt.boxplot(data\_np,labels)  
# data\_np  
  
import pandas as pd  
from pyecharts.charts import Parallel  
import pyecharts.options as opts  
import numpy as np  
  
# pr curve  
from sklearn import datasets  
import matplotlib.pyplot as plt  
  
plt.rcParams['font.sans-serif'] = ['SimHei'] # 用来正常显示中文标签  
plt.rcParams['axes.unicode\_minus'] = False # 用来正常显示负号  
  
# 从4.5的标准开始  
iris = datasets.load\_iris() # iris是一个字典集  
  
from sklearn import datasets  
import matplotlib.pyplot as plt  
  
plt.rcParams['font.sans-serif'] = ['SimHei'] # 用来正常显示中文标签  
plt.rcParams['axes.unicode\_minus'] = False # 用来正常显示负号  
  
# 从4.5的标准开始  
iris = datasets.load\_iris() # iris是一个字典集  
  
P = [] # 查准率  
R = [] # 查全率  
  
num = 4.5 # 判别标准  
  
for it in range(1, 26):  
 num += 0.1  
 TP = 0  
 FN = 0  
 FP = 0  
 TN = 0  
 for i in range(1, 100):  
  
 # 小于num认为他是对的  
 if (iris.data[i][0] <= num):  
 if (iris.target[i] == 0):  
 TP += 1  
 else:  
 FN += 1  
  
 else:  
 if (iris.target[i] == 1):  
 TN += 1  
 else:  
 FP += 1  
  
 chazhun = TP / (TP + FP)  
 chaquan = TP / (TP + FN)  
 print(TP, " ", FN, " ", FP, " ", TN)  
 P.append(chazhun)  
 R.append(chaquan)  
  
print(P)  
print(R)  
  
plt.figure(figsize=(8, 6))  
plt.plot(R, P, color="red", linestyle='--')  
plt.title("P-R曲线")  
plt.xlabel("查全率")  
plt.ylabel("查准率")  
# plt.grid(True,linestyle='--',color="red",alpha=0.5)  
plt.show()  
  
# roc curve  
from sklearn import datasets  
import matplotlib.pyplot as plt  
  
plt.rcParams['font.sans-serif'] = ['SimHei'] # 用来正常显示中文标签  
plt.rcParams['axes.unicode\_minus'] = False # 用来正常显示负号  
  
# 从4.5的标准开始  
iris = datasets.load\_iris() # iris是一个字典集  
  
TPR = [] # 真正例率  
FPR = [] # 假正例率  
  
num = 4.5 # 判别标准  
  
for it in range(1, 26):  
 num += 0.1  
 TP = 0  
 FN = 0  
 FP = 0  
 TN = 0  
 for i in range(1, 100):  
  
 # 小于num认为他是对的  
 if (iris.data[i][0] <= num):  
 if (iris.target[i] == 0):  
 TP += 1  
 else:  
 FN += 1  
  
 else:  
 if (iris.target[i] == 1):  
 TN += 1  
 else:  
 FP += 1  
  
 print(TP, " ", FN, " ", FP, " ", TN)  
 TPR.append(TP / (TP + FN))  
 FPR.append(FP / (TN + FP))  
  
print(TPR)  
print(FPR)  
  
plt.figure(figsize=(8, 6))  
plt.plot(FPR, TPR, color="red", linestyle='--')  
plt.title("ROC曲线")  
plt.xlabel("假正例率")  
plt.ylabel("真正例率")  
# plt.grid(True,linestyle='--',color="red",alpha=0.5)  
plt.show()  
  
  
clf = tree.DecisionTreeClassifier(criterion="entropy")# 构造决策树  
# 可视化决策树  
feature\_names = iris.feature\_names  
target\_names = iris.target\_names  
  
clf\_dot = tree.export\_graphviz(clf,out\_file = None,  
 feature\_names = feature\_names,  
 class\_names = target\_names,  
 filled = True,rounded = True)  
  
#clf\_tree = clf.fit(X\_train, y\_train)  
# 6.可视化  
#from sklearn.tree import export\_graphviz  
#export\_graphviz(clf\_tree, out\_file="clf.dot", )  
  
  
  
graph = graphviz.Source(clf\_dot, filename= "iris\_decisionTree.gv",format= "png")  
graph.view()

三、实验过程记录

1.导入鸢尾花数据集（'http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data），对数据进行标准化处理

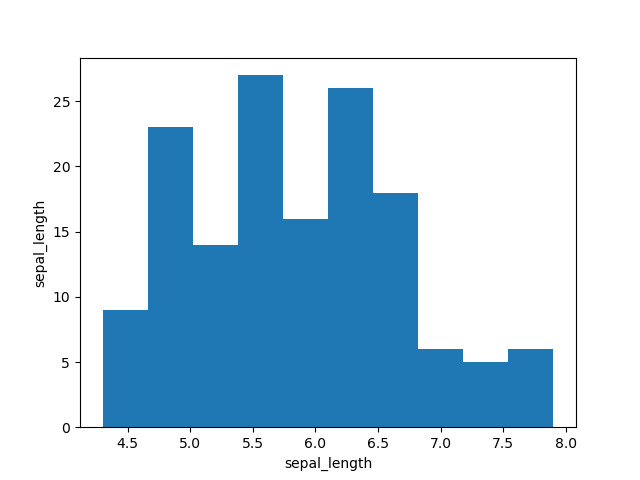


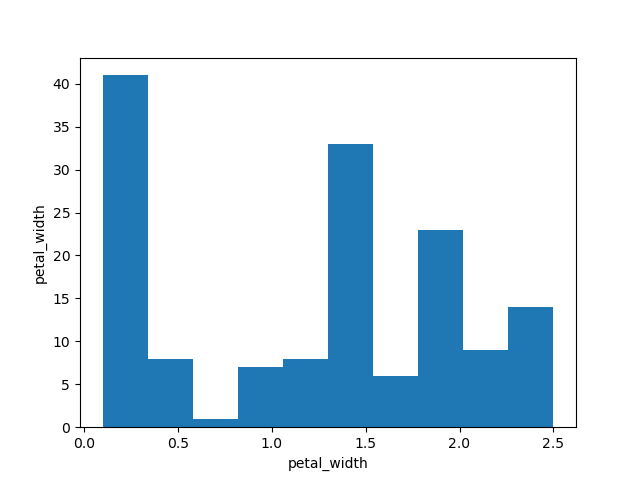
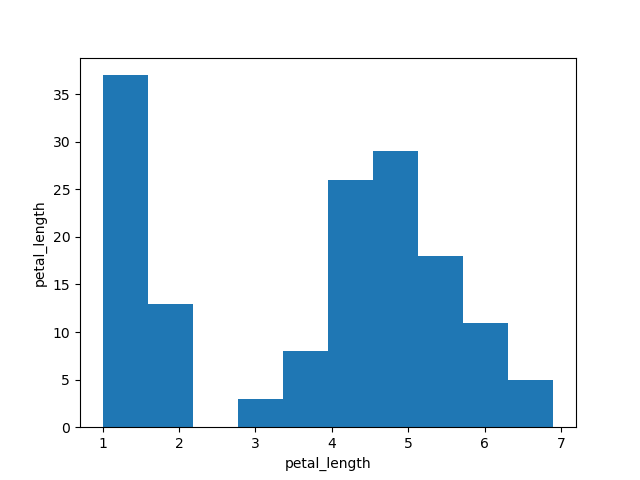
2.分别进行直方图、散点图、箱式图的绘制

3.从4.5的标准开始，通过for语句和if语句的循环嵌套，绘制pr和roc曲线

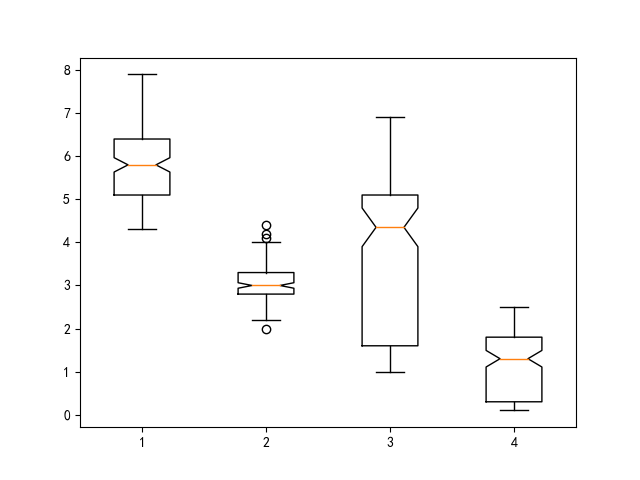
四、结果与分析

1.直方图

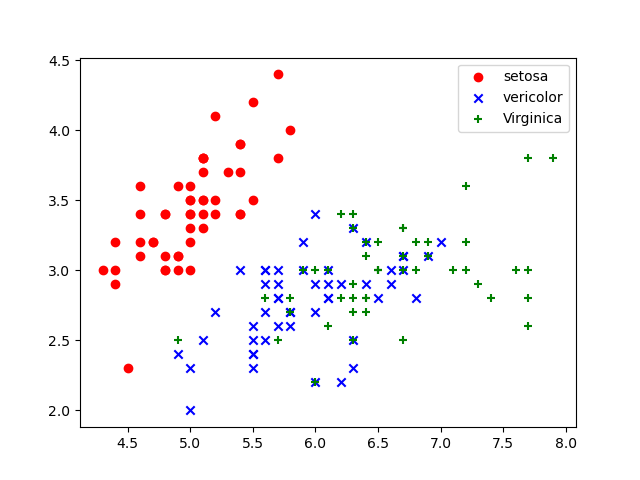


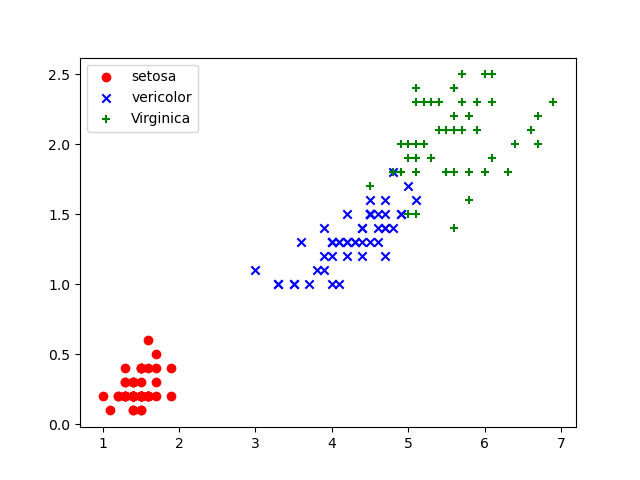


1. 箱式图

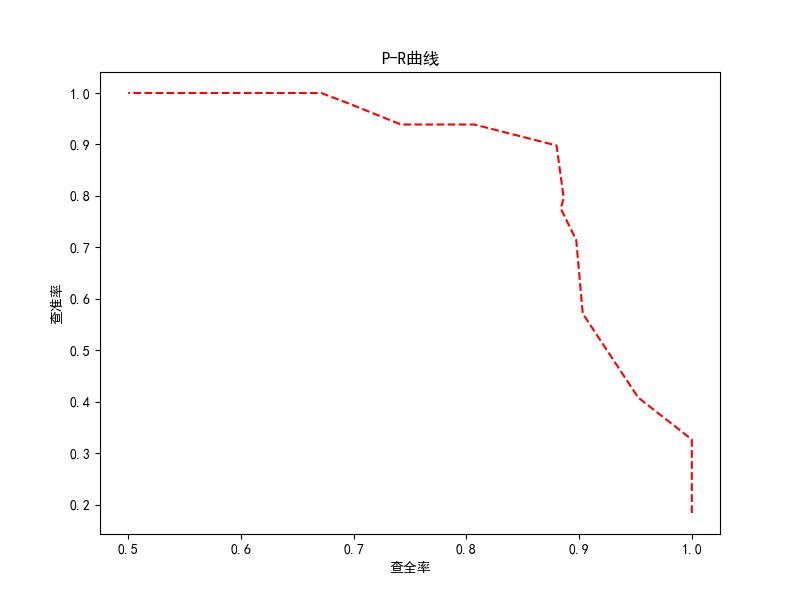


3.散点图

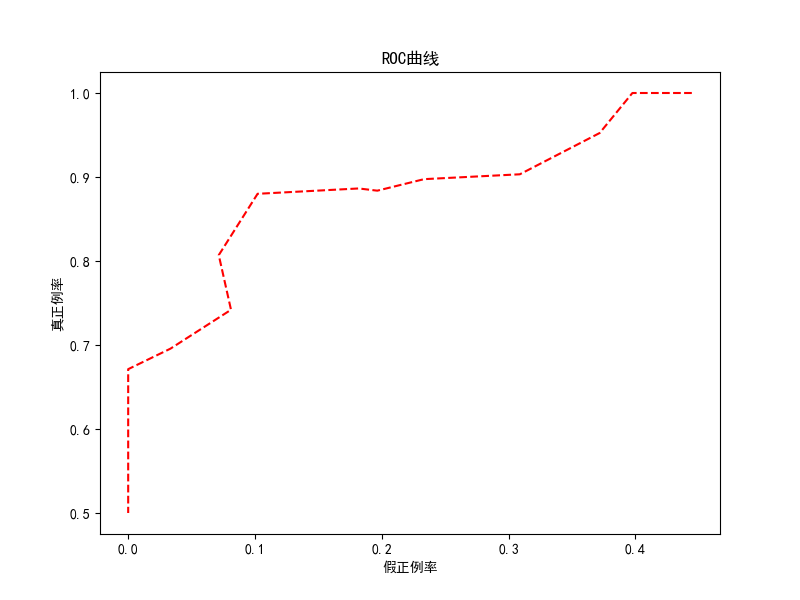




3.PR



6.ROC



五、自评

对语言的掌握不够熟练，实验完成太稚嫩还有很大可以完善改进的地方。