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　　迪丽努尔·艾在子**

**学　 号 2035061302**

**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

5. 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**

1. **直方图&箱式图**

# 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

1. **散点图**

import matplotlib.pyplot as plt

import pandas as pd

import seaborn as sns

import numpy as np

from sklearn.datasets import load\_iris

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()

1. **平行坐标图**

import pandas as pd

import pyecharts

from pyecharts.charts import Parallel

import pyecharts.options as opts

import numpy as np

#引入需要的库

url = 'https://www.gairuo.com/file/data/dataset/iris.data'

df = pd.read\_csv(url)#读取鸢尾花数据文件

dataset = df

data\_ = np.array(dataset[["sepal\_length", "sepal\_width", "petal\_length", "petal\_width"]]).tolist()

parallel\_axis = [{"dim": 0, "name": "萼片长度"},

{"dim": 1, "name": "萼片宽度"},

{"dim": 2, "name": "花瓣长度"},

{"dim": 3, "name": "花瓣宽度"}]

parallel = Parallel(init\_opts=opts.InitOpts(width="600px", height="400px"))

parallel.add\_schema(schema=parallel\_axis)

parallel.add("鸢尾花平行坐标图", data=data\_, linestyle\_opts=opts.LineStyleOpts(width=4, opacity=0.5))

# parallel.render\_notebook()

parallel.render('iris.html')#存为html文件

1. **PR曲线&ROC曲线**

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()

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()

1. **模型评估&决策树**

data = pd.read\_csv('iris.data', header=None) #读取文件

iris\_types = data[4].unique() #获取类型

for i, iris\_type in enumerate(iris\_types):

data.loc[data[4] == iris\_type, 4] = i

x = data.iloc[:, :-1] #获取所有特征

n, features = x.shape #获取样本个数和特征

y = data.iloc[:, -1].astype(np.int) #读取一共有多少类

c\_number = np.unique(y).size #获取类别

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, train\_size=0.6, random\_state=0) #划分训练集和测试集

transfer = StandardScaler() #标准化

x\_train = transfer.fit\_transform(x\_train) #标准化

x\_test = transfer.transform(x\_test) #标准化

alpha = np.logspace(-2, 2, 20) #生成-2到2的对数

models = [

['DT',DecisionTreeClassifier(random\_state=8, criterion="entropy")],

] #选取的模型

colors = cycle('gmcrb') #选取颜色

plt.figure(figsize=(7, 6), facecolor='w') #绘制图像框

for (name, model), color in zip(models, colors):

model.fit(x\_train, y\_train)

if hasattr(model, 'predict\_proba'):

y\_score = model.predict\_proba(x\_test)

else:

y\_score = model.decision\_function(x\_test)

# print('y\_score',y\_score)

y\_predict = model.predict(x\_test)

# print("y\_predict: \n", y\_predict)

# print("直接对比真实值和预测值：\n", y\_predict == y\_test)

# 计算准确率

score = model.score(x\_test, y\_test)

print("准确率为：\n", score)

# 利用精确率和召回率、F1-score

target\_names = ['class 0', 'class 1', 'class 2']

report = classification\_report(y\_test, y\_predict, target\_names=target\_names)

print("精确率和召回率、F1-score: \n", report)

print('混淆矩阵')

c = confusion\_matrix(y\_test,y\_predict)

disp = ConfusionMatrixDisplay(confusion\_matrix=c, display\_labels=iris\_types)

disp.plot(

include\_values=True, # 混淆矩阵每个单元格上显示具体数值

cmap="viridis",

ax=None,

xticks\_rotation="horizontal",

values\_format="d" )

#决策树可视化

fig = plt.figure(figsize=(25,20))

\_ = tree.plot\_tree(

clf,

feature\_names=iris.feature\_names,

class\_names=iris.target\_names,

filled=True

)

# Save picture

fig.savefig("decistion\_tree\_entropy.png")

1. **实验过程记录**
2. 首先，本次实验所用集成环境为jupyter notebook、python3.9.12、Anaconda，所需要的软件包有sklearn、matplotlib、numpy、pandas、itertools、pyechart。
3. 其次，利用scatter、hist、boxplot、Parallel函数绘制相应的图像。

的对比分析。

1. 最后，实验所选用网络模型有逻辑回归、决策树、K-邻近算法，分别对这三种模型进行了数据指标。
2. **结果与分析**
3. **直方图&箱式图**



表 1萼片长度

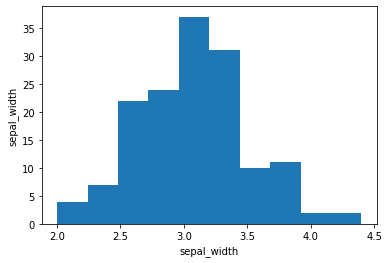


表 2萼片宽度

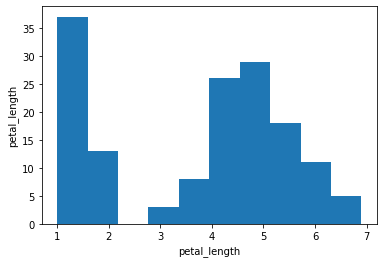


表 3花瓣长度

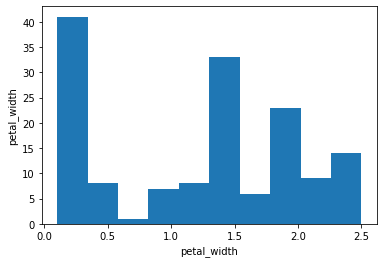


表 4花瓣宽度

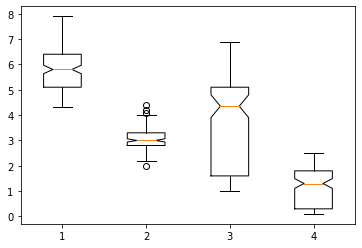


表 5箱式图

1. **散点图**

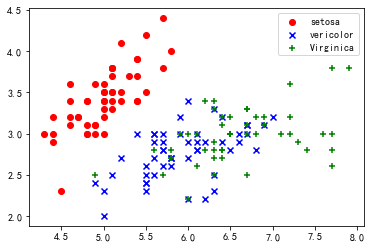


表 6萼片

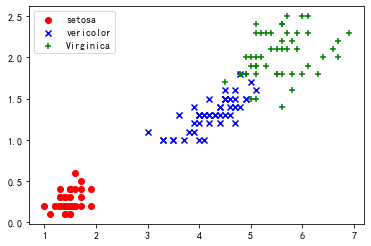


表 7花瓣

1. **平行坐标图**

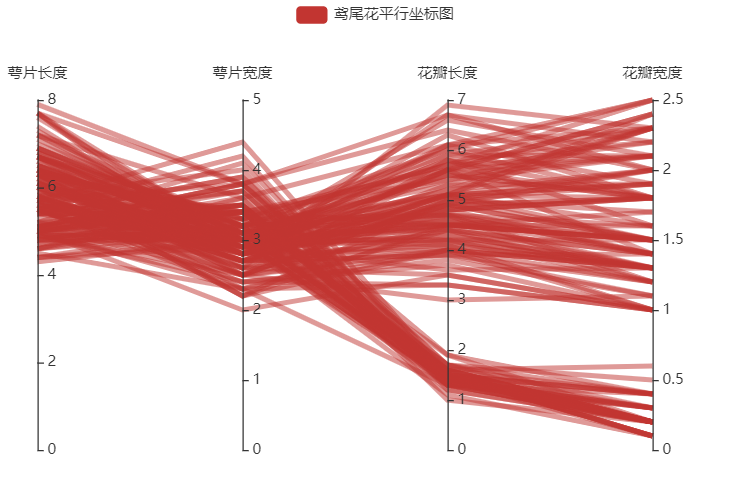


表 8平行坐标图

1. **PR曲线&ROC曲线**

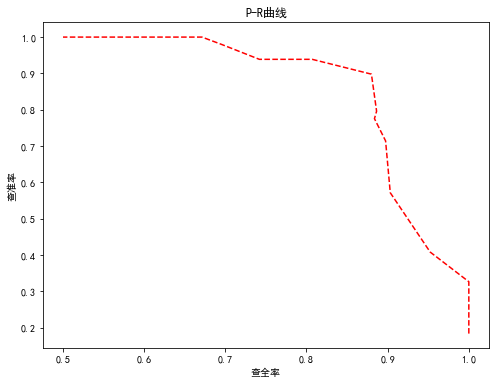


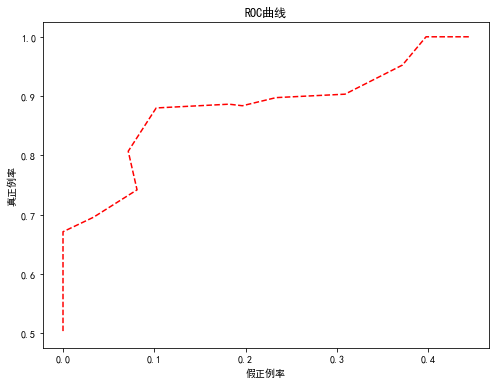
表 9 PR曲线

表 10 ROC曲线

1. **模型评估&决策树**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 样本标签 | 网络模型名称：DecisionTreeClassifier | | | |
|  | Precision | Recall | F1-score |
| 0 | 1 | 1 | 1 |
| 1 | 0.92 | 0.92 | 0.94 |
| 2 | 0.95 | 0.95 | 0.93 |
| Accuracy |  |  | 0.95 |
| Macro avg | 0.96 | 0.95 | 0.95 |
| Weighted avg | 0.95 | 0.95 | 0.95 |

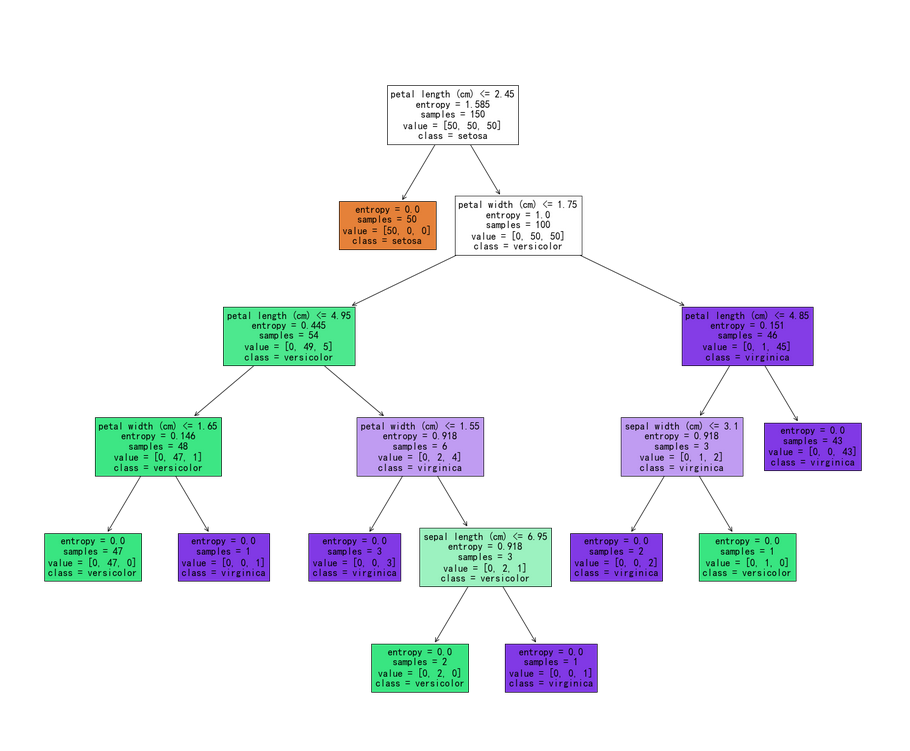


表 11 决策树

**五、自评**

掌握了数据预处理方法，缺失值修复、数据标准化，学会了数据挖掘中的分类聚类方法和分类性能的评价估计，熟悉了ROC和PR曲线评估结果的计算过程，并且通过以鸢尾花数据集为例学会了用各种函数算法绘制各种图形，也学会了绘制ROC曲线和PR曲线，并且由ROC曲线和PR曲线可以知道这些模型对鸢尾花数据集分类效果也很好。自己也熟悉了python的集成环境的编写与运行，对python有了一定程度的了解。