

python第三次实验相关资料

相关介绍

1. Matplotlib

Matplotlib 是一个 Python 的 2D 绘图库，通过 Matplotlib，开发者可以仅需要几行代码，便可以生成折线图，直方图，条形图，饼状图，散点图等，**Matplotlib中最基础的模块是pyplot。**

2. Pandas

Pandas 是 Python 语言的一个扩展程序库，用于数据分析。它是一个开放源码、BSD 许可的库，提供高性能、易于使用的数据结构和数据分析工具。Pandas一个强大的分析结构化数据的工具集，基础是 Numpy（提供高性能的矩阵运算）。它可以从各种文件格式比如 CSV、JSON、SQL、Microsoft Excel 导入数据。Pandas还可以对各种数据进行运算操作，比如归并、再成形、选择，还有数据清洗和数据加工特征

1. 打开命令行界面，分别安装pandas和matplotlib工具库

```
pip install pandas
pip install matplotlib
如果因网速问题无法下载上述两个工具库，可以使用以下命令
pip install -i https://pypi.tuna.tsinghua.edu.cn/simple pandas
pip install -i https://pypi.tuna.tsinghua.edu.cn/simple matplotlib
```

2. 在程序中分别引入pandas和matplotlib库的画图方法

```
import pandas as pd
import matplotlib.pyplot as plt
```

3. Matplotlib.pyplot基础语法

方法名称	方法作用
plt.figure(figsize, facecolor)	创建一个空白画布，figsize参数可以指定画布大小，像素，单位为英寸。
plt.plot(x,y,ls,lw,lable,color)	根据x, y数据绘制直线、曲线、标记点，ls为线型linestyle，lw为线宽linewidth，lable为标签文本内容，color为颜色。
plt.scatter(x, y, c, marker, label, color)	绘制散点图：x、y为相同长度的序列，c为单个颜色字符或颜色序列，marker为标记的样式，默认的是'o'，label为标签文本内容，color为颜色

方法名称	方法作用
plt.bar(x, height, width, bottom)	绘制条形图
plt.title(string)	在当前图形中添加标题，可以指定标题的名称、位置、颜色、字体大小等参数。
plt.xlabel(string)	在当前图形中添加x轴名称，可以指定位置、颜色、字体大小等参数。
plt.ylabel(string)	在当前图形中添加y轴名称，可以指定位置、颜色、字体大小等参数。
plt.xlim(xmin,xmax)	指定当前图形x轴的范围，只能确定一个数值区间，而无法使用字符串标识。
plt.ylim(ymin,ymax)	指定当前图形y轴的范围，只能确定一个数值区间，而无法使用字符串标识。
plt.legend()	指定当前图形的图例，可以指定图例的大小、位置、标签。
plt.show()	在本机显示图形。

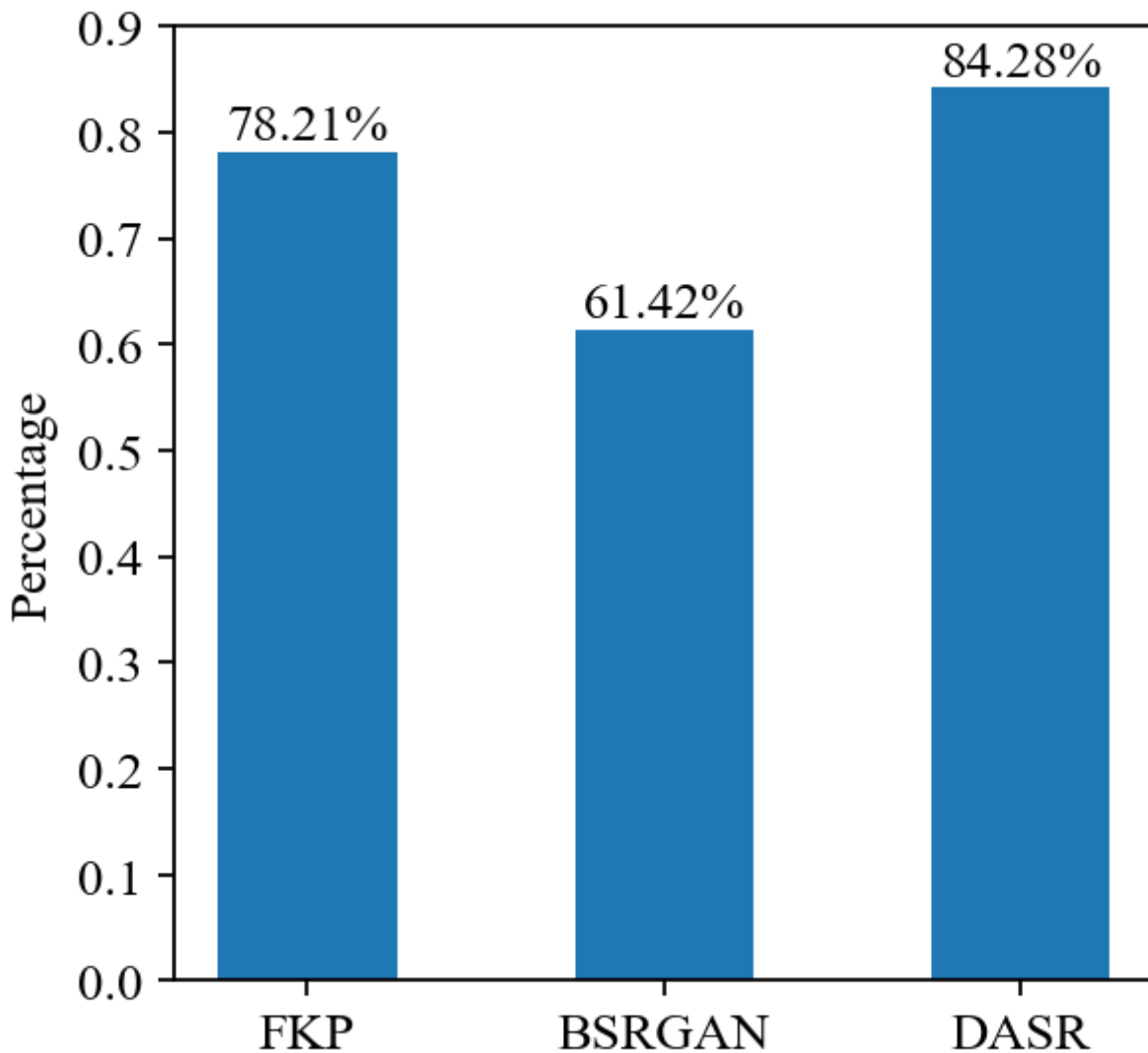
- matplotlib使用案例一

```
# 以下程序在jupyter notebook中编写，在其它编辑器中编写代码是不需要添加 %matplotlib inline
import matplotlib.pyplot as plt
%matplotlib inline

# algorithm = ["BSRGAN", "DASR", "SwinIR"]
algorithm = ["FKP", "BSRGAN", "DASR"]
results = [0.7821, 0.6142, 0.8428]

# 创建画布，指定画布的宽高，以及像素点的数量
plt.figure('label',figsize=(4,4), dpi = 160)
# width用于控制条形图之间的距离
plt.bar(algorithm, results, alpha=1, width=0.5)
# 将数据显示在柱状图上
for x,y in zip(algorithm, results):
    plt.text(x, y, '%.2f%' % (y*100), ha='center', va='bottom', fontdict={'family':'Times New Roman', 'size':12})
# 设置横坐标轴信息
plt.xticks(FontProperties="Times New Roman", size=12)
# 设置纵坐标轴label
plt.ylabel('Percentage', fontdict={'family':'Times New Roman', 'size':12})
# 设置纵坐标轴取值
plt.ylim([0, 0.9])
# 设置纵坐标轴信息
plt.yticks(FontProperties="Times New Roman", size=12)
```

output



- matplotlib使用案例二

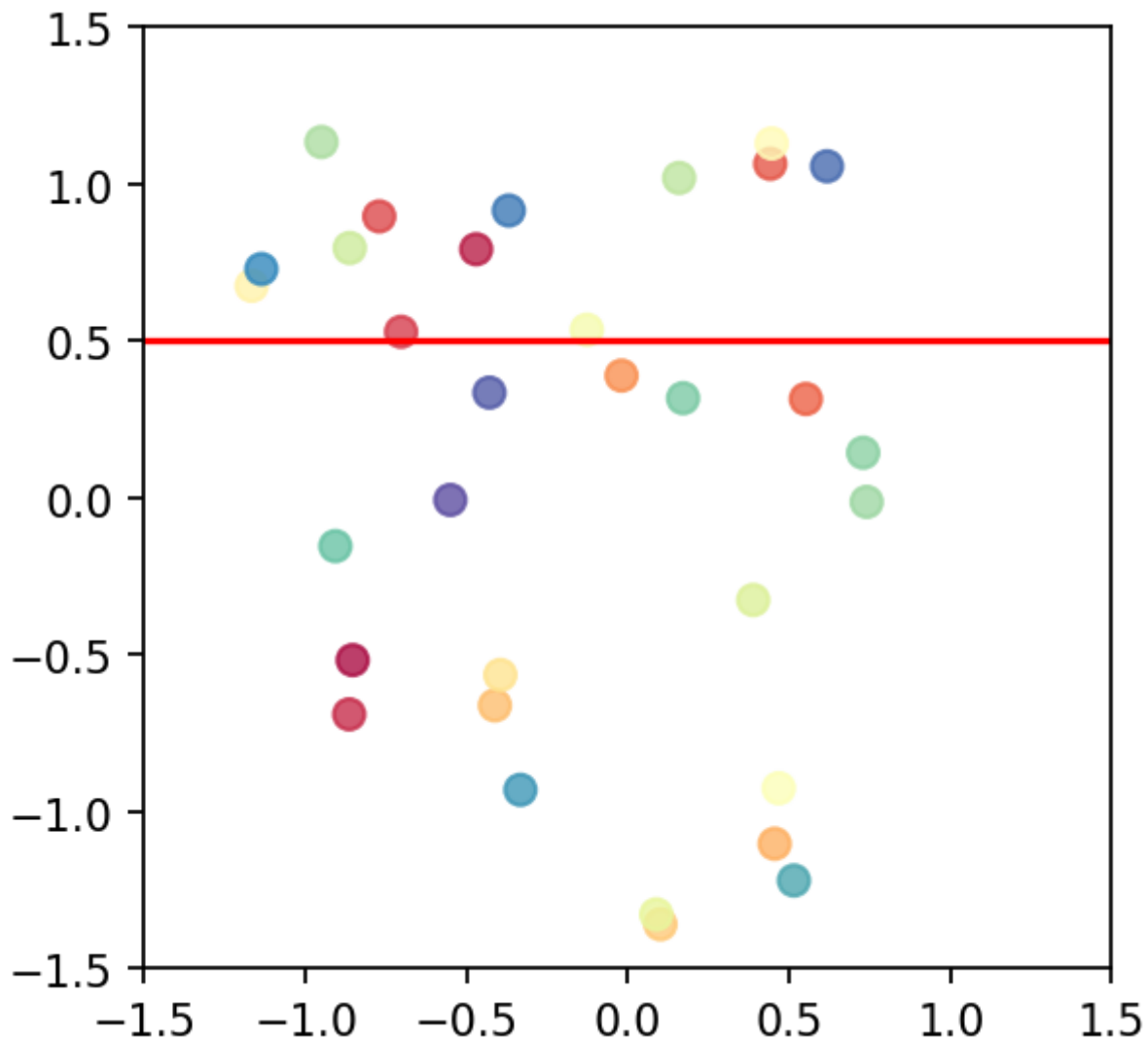
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
%matplotlib inline

n = 40
# 绘制直线数据
x1 = np.arange(-1.5, 1.51, 0.3) # 从-1.5到1.51之间每隔0.3区一个数据, 共11个数据
y1 = np.full(shape=11, fill_value=0.5) # 创建一个具有11个0.5的一维数组
# 绘制散点图数据
x = np.random.normal(0, 1, n) # 按照正态分布, 均值为0, 标准差为1, 取n个点
y = np.random.normal(0, 1, n) # 按照正态分布, 均值为0, 标准差为1, 取n个点
t = np.array(range(n)) # 随机取40个点作为颜色与数值的映射

plt.figure("scatter", (4,4), dpi=160)
# 绘制直线, color表示设置颜色
plt.plot(x1, y1, color='r')
# 参数c与cmap:连在一起使用, cmap代表的是一个颜色的集合, c代表的是一个数字集合, cmap会将颜色映射到c对应的数值上, 从而改变点的颜色
# 若不设置参数c与cmap, 默认是蓝色
plt.scatter(x, y, s=50, c=t, cmap=plt.cm.Spectral, alpha=0.8)
# 设置x轴坐标的范围, 这里为-1.5到1.5
```

```
plt.xlim((-1.5, 1.5))
# 同上
plt.ylim((-1.5, 1.5))
# np.arange(start, end, step)表示从start开始到end结束，每隔step取一个数据，注意该方法取值时区间范围左闭右开
# 设置坐标值
plt.xticks(np.arange(-1.5, 1.51, 0.5))
plt.yticks(np.arange(-1.5, 1.51, 0.5))
# 显示图像
plt.show()
```

output



4. Pandas基础语法

Pandas有两种类型的数据结构: Series(可看作一维数组)和dataframe(可看作二维数组)

- Series是一维标记数组。它可以容纳任何类型的数据。

```
mySeries = pd.Series([3,-5,7,4], index=['a','b','c','d'])
type(mySeries)
```

- dataframe是一个二维数据结构，它包含列。

```
data = {'Country' : ['Belgium', 'India', 'Brazil' ],
        'Capital': ['Brussels', 'New Delhi', 'Brassilia'],
        'Population': [1234,1234,1234]}
datas = pd.DataFrame(data, columns=['Country','Capital','Population'])
print(type(data))
print(type(datas))
```

- 读取数据

```
df = pd.read_csv('data.csv')
df = pd.read_excel('filename.xlsx')
```

- 统计数据常用函数

方法名称	方法作用
<code>df.info()</code>	函数提供有关数据信息：
<code>df.shape</code>	显示数据的行数和列数
<code>df.index</code>	显示找到的索引总数
<code>df.columns</code>	给出数据框的所有列
<code>df.count</code>	给出每一列中有多少数据
<code>len(df)</code>	计算数据框的长度
<code>df.min()</code>	给出每一列中的最小值
<code>df.max()</code>	给出每一列中的最大值
<code>df.sum()</code>	给出每一列中的求和

```
# 使用pandas遍历csv文件，csv文件中数据如下：
# name,age,score
# jay,21,90
# zhangsan,21,65

import pandas as pd
# 读取csv文件中数据
df = pd.read_csv("./data.scv")
# 遍历读取的csv文件数据
for index,row in df.iterrows():
    print("name=",row["name"], "age=",row["age"],"score=",row["score"])

# 遍历结果输出如下
# name= jay age= 21 score= 90
# name= zhangsan age= 21 score= 65
```