

第九周：Python 计算生态概览

9.1 从数据处理到人工智能

9.1.1 Python 库之数据分析

Numpy：表达 N 维数组的最基础库，C 语言实现，Python 借口使用。它是 Python 数据分析及科学计算的基础库，支持 Pandas 等。

Pandas：提供了简单易用的数据结构和数据分析工具，它可以帮助我们理解数据类型与索引关系，操作索引即操作数据。

SciPy：提供了一批数学算法及工程数据运算功能，类似 Matlab，可以用于如傅里叶变换、信号处理等应用。

9.1.2 Python 库之数据可视化

Matplotlib：提供了超过 100 种数据可视化展示效果。通过 Matplotlib.pyplot 子库调用各可视化效果。

Seaborn：统计类数据可视化功能库。

Mayavi：三位科学数据可视化功能库。

9.1.3 Python 库之文本处理

PyPDF2：用来处理 pdf 文件的工具集。完全 python 语言实现，稳定。

NLTK：自然语言文本处理第三方库。

Python-docx：创建或更新 MS Word 文件的第三方库。

9.1.4 Python 库之机器学习

Scikit-learn：机器学习方法工具集。

Tensorflow

MXNet：基于神经网络的深度学习计算框架。

9.2 实例 15：霍兰德人格分析雷达图

9.2.1 问题分析

通用雷达图绘制：matplotlib 库；专业的多维数据表示：numpy 库。

9.2.2 代码编写

代码:

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib
matplotlib.rcParams['font.family']='SimHei'
radar_labels = np.array(['研究型(I)', '艺术型(A)', '社会型(S)', \
                          '企业型(E)', '常规型(C)', '现实型(R)']) #雷达标签

nAttr = 6
data = np.array([[0.40, 0.32, 0.35, 0.30, 0.30, 0.88],
                  [0.85, 0.35, 0.30, 0.40, 0.40, 0.30],
                  [0.43, 0.89, 0.30, 0.28, 0.22, 0.30],
                  [0.30, 0.25, 0.48, 0.85, 0.45, 0.40],
                  [0.20, 0.38, 0.87, 0.45, 0.32, 0.28],
                  [0.34, 0.31, 0.38, 0.40, 0.92, 0.28]]) #数据值
data_labels = ('艺术家', '实验员', '工程师', '推销员', '社会工作者', '记事员')
angles = np.linspace(0, 2*np.pi, nAttr, endpoint=False)
data = np.concatenate((data, [data[0]]))
angles = np.concatenate((angles, [angles[0]]))
fig = plt.figure(facecolor="white")
plt.subplot(111, polar=True)
plt.plot(angles, data, 'o-', linewidth=1, alpha=0.2)
plt.fill(angles, data, alpha=0.25)
plt.thetagrids(angles*180/np.pi, radar_labels, frac = 1.2)
plt.figtext(0.52, 0.95, '霍兰德人格分析', ha='center', size=20)
legend = plt.legend(data_labels, loc=(0.94, 0.80), labelspace=0.1)
plt.setp(legend.get_texts(), fontsize='large')
plt.grid(True)
plt.savefig('holland_radar.jpg')
plt.show()
输出结果:
```

图 1

9.3 从 Web 解析到网络空间

9.3.1 Python 库之 Web 开发

注: 网络爬虫、Web 信息提取在其它课程已经具体介绍, 这里不再整理。

Django: 最流行的 Web 应用框架:

- (1) 提供了构建 Web 系统的基本应用框架;

- (2) 采用 MTV 模式：模型（model）、模板（Template）、视图（Views）；
- (4) Python 最重要的 Web 应用框架，比较复杂。

Pyramid：规模适中的 Web 应用框架

- (1) 提供了简单方便构建 Web 系统的应用框架；
- (2) 规模适中，适合快速构建并适度扩展类应用；
- (3) Python 产品级 Web 应用框架，起步简单可扩展性好。

Flask：Web 应用开发微框架

- (1) 提供了最简单构建 Web 系统的应用框架；
- (2) 特点是：简单、规模小、快速。

9.3.2 Python 库之网络应用开发

WeRoBot：微信公众号开发框架。

aip：百度 AI 开放平台借口。

MyQR：二维码生成的第三方库。

9.4 从人机交互到艺术设计

9.4.1 Python 库之图形用户界面

PyQt5：Qt 开放框架的 Python 接口

- (1) 提供了创建 Qt5 程序的 Python API 接口；
- (2) Qt 是非常成熟的跨平台桌面应用开发系统，完备 GUI；
- (3) **非常推荐！**

wxPython：跨平台 GUI 开发框架。

PyGObject：使用 GTK+ 开发 GUI 的功能库

- (1) 提供了整合 GTK+、WebKitGTK+ 等库的功能；
- (2) GTK+：跨平台的一种用户图形界面 GUI 框架；
- (3) Anaconda 采用该库构建 GUI。

9.4.2 Python 库之游戏开发

PyGame：简单的游戏开发功能库。

Panda3D：开源、跨平台的 3D 渲染和游戏开发库。

cocos2d：构建 2D 游戏和图形界面交互式应用的框架，提供了基于 OpenGL 的游戏开发图形渲染功能，支持 GPU 加速，采用树形结构分层管理游戏对象类型，适用于专业级 2D 游戏开发。

9.4.3 Python 库之虚拟现实

VR Zero：在树莓派上开发 VR 应用的 Python 库，提供了大量与 VR 开发相关的功能，

针对树莓派的 VR 开发库，支持设备小型化、配置简单化，非常适合初学者实践 VR 开发及应用。

Vizard：基于 Python 的通用 VR 开发引擎。专业的企业级虚拟现实开发引擎，提供详细的官方文档，可以用于多种 VR 设备。

9.4.4 Python 库之图形艺术

Quads：迭代的艺术。对图片进行四分迭代，形成像素风，可以生产动图或者静图。

Ascii_art：ASCII 艺术库。将普通图片转为 ASCII 艺术风格，输出可以是纯文本或彩色文本，可采用图片格式输出。

turtle：海龟绘图体系。

9.5 实例 16：玫瑰花绘制

9.5.1 代码编写

代码如下：

```
import turtle as t
# 定义一个曲线绘制函数
def DegreeCurve(n, r, d=1):
    for i in range(n):
        t.left(d)
        t.circle(r, abs(d))
# 初始位置设定
s = 0.2 # size
t.setup(450*5*s, 750*5*s)
t.pencolor("black")
t.fillcolor("red")
t.speed(100)
t.penup()
t.goto(0, 900*s)
t.pendown()
# 绘制花朵形状
t.begin_fill()
t.circle(200*s, 30)
DegreeCurve(60, 50*s)
t.circle(200*s, 30)
DegreeCurve(4, 100*s)
t.circle(200*s, 50)
DegreeCurve(50, 50*s)
t.circle(350*s, 65)
DegreeCurve(40, 70*s)
```

```

t.circle(150*s,50)
DegreeCurve(20, 50*s, -1)
t.circle(400*s,60)
DegreeCurve(18, 50*s)
t.fd(250*s)
t.right(150)
t.circle(-500*s,12)
t.left(140)
t.circle(550*s,110)
t.left(27)
t.circle(650*s,100)
t.left(130)
t.circle(-300*s,20)
t.right(123)
t.circle(220*s,57)
t.end_fill()
# 绘制花枝形状
t.left(120)
t.fd(280*s)
t.left(115)
t.circle(300*s,33)
t.left(180)
t.circle(-300*s,33)
DegreeCurve(70, 225*s, -1)
t.circle(350*s,104)
t.left(90)
t.circle(200*s,105)
t.circle(-500*s,63)
t.penup()
t.goto(170*s,-30*s)
t.pendown()
t.left(160)
DegreeCurve(20, 2500*s)
DegreeCurve(220, 250*s, -1)
# 绘制一个绿色叶子
t.fillcolor('green')
t.penup()
t.goto(670*s,-180*s)
t.pendown()
t.right(140)
t.begin_fill()
t.circle(300*s,120)
t.left(60)
t.circle(300*s,120)

```

```
t.end_fill()
t.penup()
t.goto(180*s,-550*s)
t.pendown()
t.right(85)
t.circle(600*s,40)
# 绘制另一个绿色叶子
t.penup()
t.goto(-150*s,-1000*s)
t.pendown()
t.begin_fill()
t.rt(120)
t.circle(300*s,115)
t.left(75)
t.circle(300*s,100)
t.end_fill()
t.penup()
t.goto(430*s,-1070*s)
t.pendown()
t.right(30)
t.circle(-600*s,35)
t.done()
```

结果如下：

图 2