Python数据分析到人工智能基础复习资料

一、绪论

1. Python在数据分析领域的优势

- 开源免费：无需商业授权费用。
- 丰富的库支持：如NumPy、Pandas、Matplotlib、Scikit-learn等。
- 易学易用：语法简洁，适合快速开发。
- 跨平台兼容性：支持Windows、Linux、macOS等系统。
- 社区活跃：海量开源项目和解决方案。

2. Anaconda与Python的区别与联系

- **Python**：基础的编程语言。
- **Anaconda**：Python的发行版，集成了数据分析常用库（如NumPy、Pandas）和环境管理工具（如conda）。
- 联系：Anaconda基于Python，提供更便捷的包管理和开发环境配置。

二、**Python**基本语法

1. 数据类型

- 内置数据类型
    - 可变数据类型：列表，字典，集合
    - 不可变数据类型：int，float，complex，bool，tuple，str，frozenset
    - 序列类型：元组，列表，字符串
- 第三方拓展包中的数据类型（通常比python自带的数据类型更高效、方便使用）
    - pandas中的DaraFrame
- 如何查看对象的数据类型
    - type(x)：返回x的数据类型
- 判断数据类型

- isinstance(x,type)：判断x是否为type类型，返回True/False

2. 选择语句

- 多行写法

```
if score >= 90:
    print("A")
elif score >= 80:
    print("B")
else:
    print("C")
```

- 单行写法

```
Result = Y if x > 0 else "N"
```

3. 循环语句

```
# for循环
for i in range(5):
    print(i)

# while循环
count = 0
while count < 3:
    print(count)
    count += 1
```

4. 列表操作

- 索引与切片：

```
lst = [10, 20, 30, 40]
print(lst[0])    # 输出 10
print(lst[1:3:1])  # 输出 [20, 30]
```

- 增删元素：

```
lst.extend([20,10])    # 添加元素(以元素方式)
lst + [20,10]          # 添加元素(以元素方式)
lst.append(lst)  # 添加元素(以成员方式)

lst.insert(1,8)  #在下标1位置插入元素8

lst.pop(2)         # 删除索引为2的元素
del lst[2]       # 删除索引为2的元素
lst.remove(20)   # 删除第一个20元素
```

- 列表 **vs** 元组：列表可变，元组不可变。

5. 字典操作

```
# 创建字典
person = {"name": "Bob", "age": 30}

# 增加/修改元素
person["city"] = "Shanghai"  # 新增键值对
person["age"] = 31           # 修改值
```

三、**NumPy**基本操作

1. 创建数组

```
import numpy as np
arr = np.arange(1,20)                # 等价于np.array(range(1,20,1))
arr1 = np.array([1, 2, 3])           # 一维数组
arr2 = np.array([[1, 2],
                 [3, 4]])    # 二维数组

arr = np.zeros((5,5))        # 生成五行五列都为0的array
arr = np.ones((5,5))         # 生成五行五列都为1的array
arr = np.full((3,5),2)       # 生成三行五列都为2的array
```

2. 数组属性

```
print(arr2.ndim)    # 维数 → 2
print(arr2.shape)   # 形状 → (2, 2)
print(arr2.size)    # 大小 → 4
print(arr2.dtype)   # 数据类型 → int32
```

3. 索引与切片

```
print(arr2[0, 1])   # 输出 2
print(arr2[:, 1])   # （x1,x2）其中x1为所有行，x2为1列
```

4. 合并与变形

```
arr3 = np.concatenate([arr1, arr1])  # 合并数组
arr4 = arr2.reshape(4)               # 返回新的数组 → [1, 2, 3, 4]
arr2.resize(4)                # 修改原数组 → [1, 2, 3, 4]
```

四、**Pandas**基本操作

1. 创建DataFrame与读取CSV

```
import pandas as pd
df = pd.DataFrame({"A": [1, 2], "B": ["x", "y"]})  # 手动创建
df = pd.read_csv("data.csv")                        # 读取文件
```

2. 访问元素

```
df.index         #查看行名
df.index.size    #计算行数
df.columns       #查看列名
df.columns.size #计算列数
df.shape         #显示DataFrame的形状，同时显示行列数，shape[0]为行数，
shape[1]为列数

# 显示前/后几行
df.head(5)  #前5行
df.tail(7)  #后7行
# 读取“id”列(可读取非连续多列，不同列名间用”,“隔开)
```

```
df["id"]
df.id
#读取“id”列中的"2"行
df["id"][2]
df.id[2]
#切片读取“2，4”行
df["id"][[2,4]]

# 显式(自定义)索引和隐式（默认）索引
df.loc[1,"id"]  #“1”行,“id”列(显示)
df.iloc[1,0]    #”1“行,”0“列(隐式)

# 更改index
df.reindex(index=["3","1","2"],columns=["area_mean","id"])
```

3. 删除行/列

```
# inplace为True时就地修改，为False时返回新DataFrame对象
df.drop(columns="A", inplace=True)  # 删除列
df.drop(index=0, inplace=True)      # 删除行

# 切片操作+del语句删除
del df["area_mean"]
```

4. 处理缺失值

```
df.fillna(0)       # 填充缺失值为0
df.dropna()        # 删除包含缺失值的行
```

5. 条件过滤

```
df[df["Score"] > 80]  # 筛选分数大于80的行
```

6. 统计函数

```
df.mean()    # 平均值
df.median()  # 中位数
df.std()     # 标准差
```

7. 分组统计

```
# ()内为分组条件，[]内为计算对象
df.groupby("City")["Sales"].sum()  # 按城市分组统计总销售额
```

五、可视化

1. Matplotlib绘图

```
import matplotlib.pyplot as plt
#显示汉字
plt.rcParams['font.family'] = "SimHei"

# 绘图
plt.plot(1,2,2,3,3,1)  # 曲线图,对应三点(1,2)(2,3)(3,1)
plt.scatter(1,2,3,4)  # 散点图

# 修改线的颜色和形状
plt.plot(df[id],df[area_mean],"o")  #"o"对应点，”g--“对应绿色虚
线，”rD“红色钻石

# 设置图名
plt.title(图标题)
plt.xlabel("X轴")
plt.ylabel("Y轴")
plt.legend(loc = "upper left")  #图例设置在左上角
plt.show()  # 展示图像
```

2. Pandas柱状图

```
df["Sales"].plot(kind = "bar/barh")
```