School of Economics and Management, Beihang University

Python数据分析

2. 数据结构与词频分析

本周内容

- Python面向对象编程基础
  - 关键字
  - 代码格式
  - 标识符与变量
  - 数据类型
  - 数据专题：文本简单处理

Python关键词

- 关键词
  - import keyword
  - print(keyword.kwlist)
  - with open("test.txt", "w") as f:
    - 上下文管理器，常在资源管理中用到，能够处理异常
  - None和任何其他数据类型比较永远返回False
  - nonlocal在函数或其他作用域中使用外层（非全局）变量
  - yield在生成器和协程部分会详细讨论

```
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'cla
ss', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from
', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pas
s', 'raise', 'return', 'try', 'while', 'with', 'yield']
```

代码格式

- 缩进
  - 缩进空格数可变，但是同一个代码块的语句必须包含相同的缩进空格数
  - 建议使用4个空格
- 多语句
  - 可以用；在同一行显示多条语句
  - 语句很长时可使用\来实现多行语句
    - 在[]，{}，或()中的多行语句不需要使用\
- 注释
  - 单行注释用#
  - 多行注释用'''或"""

Python标识符

- 标识符
  - 标识符由字母、数字、下划线(_)组成
  - 所有标识符可以包括英文、数字以及下划线，但不能以数字开头
    - 非ASCII码亦可
    - 类：骆驼命名
    - 属性和方法：Unix式命名
  - 标识符区分大小写
  - 以下划线开头的标识符有特殊意义
    - 以单下划线开头代表不能直接访问的类属性（“保护”）
    - 以双下划线开头代表类的私有成员，不宜被继承（“私有”）
    - 以双下划线开头和结尾代表 Python 里特殊方法专用的标识（比较关键）
    - 在类编程部分会详细介绍和使用

Python变量

- 变量
  - 变量不需要声明
  - 每个变量在使用前都必须赋值
    - 赋值以后该变量才会被创建
  - 变量没有类型
    - Python是动态类型语言
      - `a, b, c = 1, 2, "string"`
  - del 语句删除对象引用
    - `del var_a, var_b`
    - 删除后不能再引用，除非再次赋值

Python数据类型

- 标准数据类型
  - 变量所指的内存中对象的类型
  - Number(`int, bool, float, complex`)：数字
  - String(`str`)：字符串
  - List(`list`)：列表
  - Tuple(`tuple`)：元组
  - Set(`set`)：集合
  - Dictionary(`dict`)：字典

Python数据类型

- 类型的划分
  - 不可变数据： Number、String、Tuple
  - 可变数据：List、Dictionary、Set
- 类型的查询
  - `type()` 函数可以用来查询变量所指的对象类型
  - `type()`不会认为子类是一种父类类型
- 类型的判断
  - `isinstance(a, int)`可以用来判断是否是某类型
  - `isinstance()`会认为子类是一种父类类型

Python数据类型

- 数字(Number)类型
  - 整数、布尔型、浮点数和复数。
  - int (整数), 长整型
  - bool (布尔), 如 True和False。
    - type(True)和isinstance(True,int)的返回结果？
  - float (浮点数)，如3.14
  - complex (复数)，如1+3j
  - 数据类型不允许改变,这就意味着如果改变数字数据类型的值，将重新分配内存空间
    - a=20
    - b=20
    - id(a)==id(b)是否成立？如果让b=30，其地址会变化吗？
    - 对于可变类型（如后要讲的list），情部是否会变化？

Python数据类型

- 数字类型运算
  - 常规运算比较简单
  - 需要注意的一些地方
    - True 和 False 关键字的值是 1 和 0，它们可以和数字相加
      - if -1 if 0 if 2 如何判断?
    - /：除法，得到浮点数
    - //：除法，得到整数 (不一定，分子分母为浮点时得到浮点)
      - 2.//3 = 0.0 或 4.1 // 3 = 1.0
    - **：幂
    - complex：a + bj,或者complex(a,b)表示， 复数的实部a和虚部b都是浮点型
    - 比较操作可以传递（a<b==c)
      - a,b=2,3
      - print(a<b==3)

Python数据类型

- 内置常量
  - False
    - The false value of the bool type. Assignments to False are illegal and raise a SyntaxError.
  - True
    - The true value of the bool type. Assignments to True are illegal and raise a SyntaxError.
  - None
    - An object frequently used to represent the absence of a value, as when default arguments are not passed to a function. Assignments to None are illegal and raise a SyntaxError. **None is the sole instance of the NoneType type.**
  - NotImplemented
    - A special value which should be returned by the binary special methods (e.g. __eq__(), __lt__(), __add__(), __rsub__(), etc.) to indicate that **the operation is not implemented with respect to the other type**; may be returned by the in-place binary special methods (e.g. __imul__(), __iand__(), etc.) for the same purpose. It should not be evaluated in a boolean context. **NotImplemented is the sole instance of the types.NotImplementedType type**.
  - Ellipsis
    - The same as the ellipsis literal "..." . Special value used mostly in conjunction with extended slicing syntax for user-defined container data types. **Ellipsis is the sole instance of the types.EllipsisType type.**
  - __debug__
    - This constant is true if Python was not started with an -O option.

Python数据类型

- 字符串(String)
  - 单引号和双引号使用完全相同
  - 使用三引号('''或""")可以指定一个多行字符串。
  - 转义符 '\'
    - 反斜杠可以用来转义，使用r可以让反斜杠不发生转义
    - `print(r'\n')`
  - 按字面意义级联字符串，如"this " "is " "string"会被自动转换为this is string
  - 字符串可以用 + 运算符连接在一起，用 * 运算符重复。

Python数据类型

- 字符串(String)
  - 字符串有两种索引方式，从左往右以 0 开始，从右往左以 -1 开始。
  - 字符串不能改变
    - `s='abc'`
    - `s[0]= 'a'` #可否修改？
    - `s+='b'`
      - 这时还是不是同样的内存地址？
    - `s2='abc'` #s2的地址为何？
  - 没有单独的字符类型，一个字符就是长度为 1 的字符串

Python数据类型

- 字符串(String)
  - 字符串常用的函数
    - find()
    - strip()
    - split()
    - zfill(width)
      - a=''
      - b=a.zfill(10) #补0
      - id(a) == id(b)??

Python数据类型

- f-strings
  - 语法更间洁，速度也更快

```
year = 2025
event = 'Ceremony'
print(f'Results of the {year} {event}')

yes_votes = 42_572_654
no_votes = 43_132_495
percentage = yes_votes / (yes_votes + no_votes)
print(f'{yes_votes:-9} YES votes  {percentage:2.2%}')

animals='eels'
print(f'My hovercraft is full of {animals!r}.')
```

Python数据类型

- 读入字符串
  - 从文件读入

```
with open(file_path, 'r') as f:
    for line in f:
        pass
```

  - 用input从标准输入读入
    - 获取用户输入，返回字符串

```
age = int(input("请输入你的年龄："))
expression = input("请输入一个数学表达式（例如 2 + 3 * 4）：")
result = eval(expression)
```

Python数据类型

- 列表(List)
  - 使用最频繁的数据类型
    - `list1=['a','b','c']`
    - `list2=['d','e','f']`
    - `print(list1*2)`
    - `print(list1+list2)`
  - 有步长的元素截取
    - `list1[1:4:2]`
    - `nl=[0,1,2,3,4,5,6,7,8,9,10]`
    - `print(nl[0::2])`将输出？
    - `print(nl[::2]`将输出？
    - `print(nl[1::2]`将输出？

Python数据类型

- 列表(List)
  - 逆序
    - `ll=['A','B','C','D','E','F','G']`
    - `print(ll[-1::-1])`
    - `print(ll.reverse())`
  - 元素的删除
    - `del ll[0]`
    - `ll.remove('A')`
    - `a.clear()` 相当于 `del a[:]`
    - `del a`
  - 元素的增加
    - `a.append(x)` 相当于 `a[len(a):]=[x]`
    - `a.extend(l)` 相当于 `a[len(a):]=l`

Python数据类型

- 列表(List)
  - 排序
    - a.sort()
  - 复制
    - a.copy()
    - slice copy，在可能对原列表有修改的操作中较常见
    - b=a[:]#深浅拷贝在后面会继续讨论
  - 推导式
    - a=[x**2 for x in range(6)]
    - a=list(map(lambda x:x**2,range(6)))
    - pis=[str(round(math.pi, i)) for i in range(1, 6)]
    - a=[(x, y) for x in [1,2,3] for y in [3,1,4] if x != y] (跟zip的区别)

Python数据类型

- 列表(List)
  - 一些复杂但可能有用的操作
    - `ll=[[1,3,4],[5,6,7,8],[9,10,11]]`
    - `ll_f=[e for ele in ll for e in ele] #flatten list`
  - 矩阵交换行列
    - `matrix=[[1,2,3],[4,5,6],[7,8,9]]`
    - `matrix2=[[row[i] for row in matrix] for i in range(3)]`
      - 按列遍历矩阵
  - 比较大小
    - 序列对象可以与其它序列对象比较
    - 注意运算规则，不建议直接使用

Python数据类型

- 列表排序
  - dl=[random.randint(0,1000) for i in range(100)]
  - dl.sort()
  - dls=sorted(dl,reverse=True)
  - dls=[]
  - import heapq
  - for n in dl:
    - heapq.heappush(dls,n)
  - heapq.heapify(dl)
- 高效合并多个有序列表
  - d1=[random.randint(0,1000) for i in range(100)]
  - d2=[random.randint(0,1000) for i in range(100)]
  - d1.sort()
  - d2.sort()
  - d=list(heapq.merge(d1,d2))

Python的数据类型

- 列表排序
  - 如何在插入新元素时自动更新顺序
  - import bisect
  - nums = [random.random() for i in range(10)]
  - l=[]
  - for n in nums:
    - position = bisect.bisect(l, n)
    - print(position)
    - bisect.insort(l,n)
    - #bisect.insort_left(l,n)
  - print(l)

Python数据类型

- 元组（tuple）
  - 与列表类似，不同之处在于元素不可修改
    - 可以把字符串看作一种特殊的元组
  - 虽然tuple的元素不可改变，但它可以包含可变的对象
    - `tup1=([1,2,3],0)`
    - `tup1[0]=[1,2,3,4]??`
    - `tup1[0].append(4)??` （联想到深浅拷贝的区别）
  - 构造包含 0 个或 1 个元素的元组
    - `tup1 = ()  # 空元组`
    - `tup2 = (20,)` # 一个元素，需要在元素后添加逗号，否则含义不明确
  - 元组中的元素值不允许删除，但`del`能删除整个元组

Python数据类型

- 命名元组
  - namedtuple
  - 元组中的元素可通过名称进行访问
  - import collections
  - Student = collections.namedtuple('Student','name credit hometown')
  - s1=Student(name='zjc',credit='95',hometown='gs')#元素列表要完整
  - print(s1)
  - print(s1.credit)
  - s1.hometown='bj'
  - # AttributeError: can't set attribute

Python数据类型

- 集合(Set)
  - 一般用于进行成员关系测试和删除重复元素
  - 可以使用大括号 { } 或者 set() 函数创建集合
    - s1={e1,e2,e3}
    - s2=set(value)
  - 创建空集合必须用 set() 而非 { }
    - { } 用来创建一个空字典
  - 集合的运算
    - - | &
    - update()可实现批量添加元素
  - 集合的实现
    - 散列实现,不能包含可变类型
    - 包含list的元组可否加入set??

Python数据类型

- 字典(Dictionary)
  - 一种映射类型，其元素是键值对
  - 无序的键(key)：值(value) 集合
    - 键(key)必须使用不可变类型
    - 同一个字典中键(key)必须是唯一的
  - 函数 dict() 可以直接从键值对序列中构建字
    - 构造函数
    - `dict([('sape', 4139), ('guido', 4127), ('jack', 4098)])`
    - `dict(sape=4139, guido=4127, jack=4098)`
    - `dict(list(enumerate(['one','two','three'],start=1)))`
  - 散列实现

推导式

- 字典推导（dictionary comprehension）
  - `even_dict={x : x**2 for x in range(11) if x % 2 ==0}`
- 集合推导（set comprehension）
  - `odd_set={x**2 for x in range(11) if x % 2 ==1}`

字典更新

- 一般方法
  - `freq={}#统计词频`
  - `if term in freq:`
    - `freq[term]+=1`
  - `else:`
    - `freq[term]=1`
  - 更简单的方案
    - `count=freq.get(term,0)`
    - `freq[term]=count+1`

Python数据类型

- Counter
  - 常用于计数（如文本处理中）
  - 可自动进行计数的更新
  - freq=collections.Counter(['a', 'b', 'c', 'a', 'a', 'b'])
  - freq.update('abca')
  - for l in 'abc‘:
    - print("{}:{}".format(l,freq[l]))

Python数据类型

- defaultdict
  - 可以设定键的默认值
  - 避免无键时出现错误
  - def default_value():#该可调用对象用来设定默认值
    - return -1
  - dc = collections.defaultdict(default_value, zjc='39')#default_value可以是None
  - print(dc['zjc'])
  - print(dc['lsy'])
  - dc['lsy']+=1
  - print(dc['lsy'])

Python数据类型

- 字典(Dictionary)
  - 如何有序地输出？
  - `for key in sorted(dic.keys()):`
    - `print(key,dic[key])`
  - 如何构建有序字典
    - 即对字典排序
    - `import collections`
    - `dic = collections.OrderedDict()`
    - `dic=collections.OrderedDict(`
    - `sorted(unsorted_d.items(),`
    - `key=lambda dc:dc[1],`
    - `reverse = True))`

Python数据类型

- 判断相等
  - is运算符
    - a is b 相当于 id(a)==id(b)
  - is 用于判断两个变量引用对象是否为同一个
  - == 用于判断引用变量的值是否相等

Python数据类型

- 强制类型转换
  - int(x [,base]) 将字符串x转换为一个整数
  - float(x) 将x转换到一个浮点数
  - complex(real [,imag]) 创建一个复数
  - str(x) 将对象 x 转换为字符串
  - repr(x) 返回对象 x 的字符串表达
  - eval(str) 用来计算在字符串中的有效Python表达式,并返回一个对象
    - eval('print(2+3)')
    - eval(`exit()')
  - tuple(s) 将序列 s 转换为一个元组
  - list(s) 将序列 s 转换为一个列表

Python数据类型

- 强制类型转换
  - set(s) 转换为可变集合
  - dict(d) 创建一个字典，d 必须是一个 (key, value)元组序列。
  - frozenset(s) 转换为不可变集合
  - chr(x) 将一个整数转换为一个字符
  - ord(x) 将一个字符转换为它的整数值
  - hex(x) 将一个整数转换为一个十六进制字符串

数据专题：词频统计

- 文本处理的第一步往往是词频统计
  - 分词 (jieba)
    - import jieba
    - print(list(jieba.cut("我来到北京航空航天大学")))
  - 去除停用词
  - 找到“特征词”
    - 词频
    - 词性
  - unigram和bigram
  - 文本向量
  - Demo: jdemo.py

数据专题：大模型处理

- 通过编写prompt让大模型进行分词和词频统计等
  - 角色：数据分析工程师
  - 请进行中文分词/词频/词性/特征词
  - 输出要求：一行一个

本周作业

- 见机考平台
  - 访问地址：http://10.212.253.252/indexcs/simple.jsp?loginErr=0

SI-Python变量

- 变量
  - 获取变量所指对象的内存地址
  - `id(var)`
  - `print(hex(id(var)))`
  - `a=2`
  - `del a`
  - `print(id(2))`
  - `c=2`
  - `print(id(c))`

SI-Python数据类型

- 内存分配
  - 为提高内存利用效率,对于简单对象如int对象或字符串对象等，会采取重用对象内存的办法
    - 解释器依赖，不同的解释器可能有不同的实现
    - 对值大小的依赖（如-5至256）
  - x=2
  - y=2
  - print(id(x))
  - print(id(y))
  - x=3
  - print(id(3))
  - print(id(x))

SI-Python数据类型

- 字符串模板
  - 格式化输出（后续内容中还会提及）
    - values = {'price':'5.99','dis':'50'}
    - t1 = string.Template("""PRICE: $price$$\nDISCOUNT: $dis%""")
    - print(t1.substitute(values))
- 字符串比较
    - import difflib
    - d = difflib.Differ()
    - diff = d.compare(s1,s2)#’a’, ‘abc’
    - for letter in diff:
      - print(letter)

SI-Python数据类型

- 字符串对齐
  - import textwrap
  - textwrap.dedent(sample_text)
  - textwrap.indent(wrap,'>')
  - textwrap.fill(wrap,width=50)
  - textwrap.fill(wrap,initial_indent='',subsequent_indent='*4,width=50)
  - textwrap.shorten(sample_text,30,placeholder='...')

SI-Python数据类型

- 内存分配
  - L=[1,2,3]
  - L2=[1,2,3]
  - id(L)==id(L2)??
  - L1=L[:]
  - id(L1)==id(L)??
  - L.append(4)
  - id(L)==?

SI-Python数据类型

- ChainMap
  - 组合多个字典使其在逻辑上表现为一个整体
  - 创建了一个单一的可更新视图
  - 键查找时返回第一次出现的目标键
  - 注意对其修改会影响原始的字典
  - 常用于上下文参数或配置的管理
  - `d1={'path':'/c/d/e','cmd':'clear','pwd':'$'}`
  - `d2={'root':'/','cmd':'cls','pwd':'#','prompt':'True'}`
  - `d=collections.ChainMap(d1,d2)`
  - `print(d['path'])`
  - `print(d['cmd'])`
  - `d['cmd']='update'`
  - `print(d1['cmd'])`
  - `print(d2['cmd'])`

SI-如何理解repr

- repr()函数
  - 返回对象的可打印表示形式
  - printable representation
  - 代码测试时常被使用（打印对象的原始“面貌”）
  - print(5)与print('5')有何区别？
  - repr(5)与repr('5')有何区别？
  - a='10'
  - b=eval(a) #b=10
  - b=eval(repr(a))#b='10'
  - b == a ??