

《机器学习编程实践》课程—— DAY 1

引言

1 问：如何学好编程？

- “上课影响学习” —— 与授课的原则相反，不需要有抽象定义
- 对于新手而言，只需看懂代码 —— 巧用Ctrl C + Ctrl V
- 理解大概框架，实践过程中遇到不懂的再查

2 课程主要内容

- python基础知识介绍
- 用k-means的实践了解如何使用sklearn库
- 简单线性回归的实践（加入画图+数据处理等流程）

一、 python基础知识介绍

1 列表

1.1 特点

- 可存放任何元素
- 不要求每行维度相同
- 用“[]”
- 例：

```
c = ["ml", a, b, ["list"]]
print(c[0], c[3][0])
```

显示: ml list

1.2 常见运算

```
c + ["aaa"]
```

列表中加入元素：“aaa”

```
c + ["aaa"]
```

列表中加入元素: ["aaa"]

```
c * 2
```

列表中所有元素拷贝一份加入元列表元素之后

```
c.append("new_element")
```

列表中加入元素: "new_element"

```
c.remove(["list"])
```

列表中移除["list"]

2 元组

2.1 特点

- 一旦产生不可更改
- 用"()"

3 字典

3.1 特点

- 可以理解为带名字的列表
- 用"{}"
- 例:

```
Dict = {"name1": "ml", "name2": 1, "name3": [1, 2]}
```

3.2 常见运算

```
print(Dict["name2"])
```

输出结果为: 1

```
Dict["name4"] = "new_ele"
```

字典中加入第四个元素, 键名为" name4", 值为"new_ele"

```
del Dict["name1"]
```

删除第一个元素

4 for循环

4.1

```
for i in c:  
    print(i)
```

输出结果为:

```
4  
abc  
new_element
```

4.2

```
for i in "abc":  
    print(i)
```

输出结果为:

```
a  
b  
c
```

4.3

```
for i in ["abc"]:  
    print(i)
```

输出结果为:

```
abc
```

4.4

```
for i in Dict.items():  
    print(i)
```

输出的是字典中每一个键值对组成的元组

- 使用for循环对字典进行遍历，默认获取的是字典的每个键

5 类

5.1 类的实例化

```
class MyClass:  
    i = 12345  
  
    def f(self, b):  
        print(self.i)  
        return 'hello world'  
  
x = MyClass()
```

5.2 访问、修改和增加类的属性

```
goal = x.i  
x.i = 123  
x.j = 456
```

5.3 初始化函数

```
class Circle:  
    pi = 3.1415  
    def __init__(self, r):  
        self.r = r  
  
circle1 = Circle(r=1)
```

- 使用初始化函数的好处：实例化时对某个变量的值进行更新时，不需要在实例化后额外加一步

- 类中的任何方法都需要self作为参数
- r可以作为默认参数
- pi与r的区别：pi是Circle的属性，r是只有实例才具有的属性

5.4 类的继承

```
class Ellipse(Circle):
    def __init__(self, r, r1):
        super(Ellipse, self).__init__(r)
        self.r1 = r1

ell = Ellipse(2, 3)
```

- Ellipse继承自Circle类
- 就算父类中r为默认参数，子类中也必须要有super(Ellipse, self).__init__(r)，不过可以写成super(Ellipse, self).__init__()

二、 sklearn库的使用—— K-means实践

1 如何查询库内各方法的使用方法

查询官方文档：scikit-learn —— User Guide —— clustering（以聚类方法为例）

2 简单库的介绍

- matplotlib 库—— 主要绘图库
- matplotlib.pyplot 模块—— 绘制图表（如散点图、折线图等）
- numpy 库—— 提供多维数组计算支持（如生成数据、矩阵运算）
- scikit-learn 的 datasets 模块—— 生成或加载数据集
- from sklearn.cluster import KMeans: 从 sklearn.cluster 导入 KMeans 类—— 实现 K-Means 聚类算法，用于无监督学习。

三、 简单线性回归的实践