《机器学习》考点&习题

第一章 机器学习概述

1、人工智能简介

(1)智能：人类智力和能力的总称

(2)人工智能：智能机器和智能系统的总称

2、三次浪潮

(1)自动推理(1956 年——20 世纪 70 年代中期)

线性感知器算法无法解决异或等非线性问题

(2)知识(20 世纪 80 年代)

中心：知识

最出名：专家系统(ES)

新进展：人工智能(ANN)

(3)学习(20 世纪 90 年代至今)

3、人工智能的研究领域

(1)机器学习

监督学习、非监督学习、强化学习

(2)专家系统

知识库、推理机、综合数据库、解释器、人机交互界面、知识获取

(3)自然语言处理 NLP

(4)智能决策系统 DSS

会话部件、数据库、模型库

(5)推荐系统

系统过滤系统，用于预测用户对物品的评分或偏好

(6)智能识别

本质：模式识别 PR

4、机器学习的主要工作

(1)从数据中学习——监督学习

(2)分析无经验的新问题——无监督学习

在不提供监督信息(预测量的真实值)的条件下进行学习

(3)边行动边学习——强化学习

5、人工智能的目的

让机器能够模拟、延伸和扩展人的智能，以实现某些脑子劳动的机械化

第二章 Python 数据处理基础

1、基本数据类型

(1)可变：列表、集合、字典——内存地址不发生变化

(2)不可变：数字、字符串、元组——重新开辟一块内存空间

2、字典(Dictionary)

能否修改、重复；特点；使用

(1)能否修改、重复&特点：

- 字典使用大括号{}定义，格式为：*d = {key1 : value1, key2 : value2}*
- 键一般是唯一的。如果出现了重复，则后面的键值对会替换前面的键值对
- 值的数据及类型不限，可以是字符串、数字或元组

(2)使用：

- 访问

```
dict = {'Name': 'Mary', 'Age': 7, 'Class': 'First'};
print(dict);
print("Name: ", dict['Name'])
print("Age: ", dict['Age'])
```

- 添加&修改

```
dict = {'Name': 'Zara', 'Class': 'First'};
#添加add
dict['Gender']="Female"
print(dict)
#修改update
dict.update({"No":"001"})
print(dict)
#也可以使用update方法添加/修改多个数据
dict.update({'Gender':"F","Id":1})
print(dict)
```

- 删除

```
del dict['Gender']  删除一个字典
print(dict)
dict.clear()        清空字典
print(dict)
```

3、什么是索引？——索引是为了加速对表中数据行的检索而创建的存储结构，表示有序列表中的位置，Dataframe 的列可以通过索引进行访问，是一个 Index 对象，因此 Series 的索引不只是数字，也包括字符等

```
lis= ['蚂蚱','螳螂','蝈蝈','蝗虫','蛐蛐']
#(1)直接遍历
for item in lis:
    print(item)
 #(2)按索引遍历
for i in enumerate(lis):
    print(i)
#(3)对于列表类型，还有一种通过下标遍历的方式，如使用range()函数
for i in range(len(lis)):
    print(lis[i])
```

4、数字(略)&字符串型

获取字符串的一部分的操作称为**切片**

```
                a     b     c     d     e     f     g
栅栏式位置：   |-----|-----|-----|-----|-----|-----|-----|
正序位置编号：0     1     2     3     4     5     6     7
正序字符编号：| 0 | 1 | 2 | 3 | 4 | 5 | 6 |
逆序位置编号：-7   -6    -5    -4    -3    -2    -1
逆序字符编号：| -7 | -6 | -5 | -4 | -3 | -2 | -1 |
```

```
#字符串的访问
str = 'Picture'
print (str[1:3])     # 第二、三个字符
print (str[-3:-1]) # 倒数第二、三个字符
print (str[3:-1])   # 正数第四个到倒数第二个字符
print (str[-6:7])   # 倒数第六个到正数第七个字符
print (str[2:])      # 第三个字符开始的所有字符
print (str * 2)      # 输出字符串两次
print (str + "TEST") # 连接字符串
```

```
ic
ur
tur
icture
cture
PicturePicture
PictureTEST
```

注：由于字符串是不可变类型，所以向字符串某位置赋值会导致错误

```
word = 'Python'
print(word[0], word[5])
print(word[-1], word[-6])
如果继续添加一行语句：
word[0] = Q'   #无法修改 word 字符串
```

5、列表

```
#列表的访问
list = [ 'a', 56 , 1.13, 'HelloWorld',[7,8,9] ]
print(list)            #完整列表
print(list[4])       # 第五个元素
print(list[-2:5]) # 从倒数第二个到正数第五个元素
print(list[2:])     # 第三个元素开始的所有元素
```

```
['a', 56, 1.13, 'HelloWorld', [7, 8, 9]]
[7, 8, 9]
['HelloWorld', [7, 8, 9]]
[1.13, 'HelloWorld', [7, 8, 9]]
```

```
#列表的修改
a = [1, 2, 3, 4, 5, 6]
a[0] = 9      # 将第一个元素设为 9
print(a)
a.append(7) # 在列表末尾追加 7
print(a)
a[2:5] = [] # 将第三到五个元素值设置为空值
print(a)
a.pop(2)      # 将第三个元素移除
print(a)
```

```
[9, 2, 3, 4, 5, 6]
[9, 2, 3, 4, 5, 6, 7]
[9, 2, 6, 7]
[9, 2, 7]
```

```
#列表的遍历
lis= ['蚂蚱','螳螂','蝈蝈','蝗虫','蛐蛐']
#(1)直接遍历
for item in lis:
    print(item)
 #(2)按索引遍历
for i in enumerate(lis):
    print(i)
#(3)对于列表类型，还有一种通过下标遍历的方式，如使用 range()函数
for i in range(len(lis)):
    print(lis[i])
```

6、元组——元素不能修改

```
#元组的访问
tuple = ( 'SpiderMan',2017 ,33.4, 'Homecoming', 14 )
tinytuple = (16, 'Marvel')
print (tuple)          # 输出完整元组
print (tuple[0])     # 输出元组的第一个元素
print (tuple[3:4]) # 输出从第二个元素开始到第三个元素
print (tuple + tinytuple) # 连接元组
```

```
('SpiderMan', 2017, 33.4, 'Homecoming', 14)
SpiderMan
('Homecoming',)
('SpiderMan', 2017, 33.4, 'Homecoming', 14, 16, 'Marvel')
```

#第三行列表对应的输出为['Homecoming'](**无逗号**)

```
#元组的元素不可改变，但如果元组包含了可变类型的数据项，则该数据项可以修改
tuple = ([16, 'Marvel'] , 'SpiderMan',2017 ,33.4, 'Homecoming', 14,)
print(tuple[0])
tuple[0][0]='Marvel'
tuple[0][1]='16'
print (tuple)
```

```
[16, 'Marvel']
(['Marvel', '16'], 'SpiderMan', 2017, 33.4, 'Homecoming', 14)
```

7、集合

```
#创建集合
#创建一个空集合
var = set()
print(var,type(var))      #显示集合内容和类型
#具有数据的集合
var = {'LiLei','HanMeiMei','ZhangHua', 'LiLei', 'LiLei'}
print(var,type(var))      #显示集合内容和类型
```

```
#集合成员检测
#判断元素在集合内
result = 'LiLei' in var
print(result)
#判断元素不在集合内
result = 'lilei' not in var     #大小写敏感
print(result)
```

```
#增加、删除集合元素。
var = {'LiLei','HanMeiMei','ZhangHua'}
var.add('LiBai')        #add 方法添加元素
print(var)
var.update('DuFu')     #update 方法首先拆分元素，然后各个添加
print(var)                  #数据项无序，且去除重复项
var.remove('D')
var.remove('F')
var.remove('u')
print(var)
```

```
#集合的遍历
anml ={'紫貂','松貂','青鼬','狼獾'}
for item in anml:print(item)
&&
for item in enumerate(anml):print(item)
```

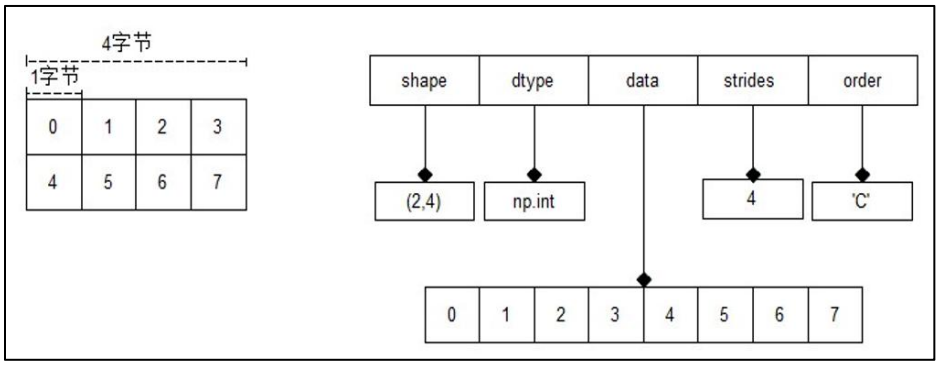
第三章 Python 常用机器学习库

1、有哪三个常用的机器学习库？

NumPy, Pandas , Matplotlib

2、NumPy

(1)ndarray 的数据结构



order：C 为行方向，F 为列方向，A 为任意方向(默认)

ndarray()和 array()都可以用来创建数组函数

(2)例题：使用 ndmin 参数设置数组的最小维度。

```
import numpy as np
a = np.array([1,2,3,4,5], ndmin=2)
print(a)
```

输出：[[1,2,3,4,5]]

(3)数据属性

• 常见术语

轴 axis(为 0 对列操作，为 1 对行操作)，秩 rank

注：几个数组就是几维数组、秩就是几，数组的长度即为轴的长度

例如：[0,1,2]是一维数组，只有一个轴，其秩为 1，轴长度为 3；[[0,1,2],[3,4,5]]是二维数组，有两个轴，其秩为 2，轴的长度分别为 2、3。

• 基本属性：ndarray.shape 显示数组的维数，调整数组大小(reshape 也可以调整数组大小)

例如：

```
a=np.array([0,1,2,3,4,5,6,7])
b=a.reshape(2,2,2)
print(b)
print(b.ndim)
```

输出：

```
[[[0 1]
  [2 3]]
 [[4 5]
  [6 7]]]
```

(4)产生数列函数： range(a,b,c) 和 arange(a,b,c) a:开始 ，b:结束，c:步长

- 区别：range()的步长要求为整数，且默认为 1，可更改；
  arange()的步长随便，默认为 1
- 例如 arange(4) 产生 0~3 的步长为 1 的数列：[0 1 2]

(5)切片

arr[a : b] :不管正序还是逆序，都是 a 到 b-1 为止；逆序没有 0 下标

例如：

```
# 截取第 2 行到最后一行，第 1 列到第 4 列构成的 ndarray
arr1=arr[1:, :3]
```

(6)sort()函数

- 格式及各字段含义

```
sort(a,axis,kind=None,order)
a：数组
axis：0 在列上排序，1 在行上排序
kind：算法类型
order：字符串列表
```

- 例题

设 a=[[20,3,100],[1,200,30],[300,10,2]]

axis 为 0 时，sort(a,axis)，结果为：
[[1,3,2],[20,10,30],[300,200,100]]
axis 为 1 时，sort(a,axis)，结果为：
[[3,20,100],[1,30,200],[2,10,300]]

3、Pandas——三种数组：Serie,DataFrame,Penel

(1)series 数据结构

- 创建 series 对象——使用函数：pd.series(data, index)
  - data 表示数据值，index 是索引，默认情况下是 0 到 N-1(N 为数据的长度)的整型索引
  - 访问 Series 对象成员可以用索引编号，也可以按索引名

```
import pandas as pd
#使用列表创建，索引值为默认值。
print('--------     列表创建 series     ----------')
s1=pd.Series([1,1,1,1,1])
print(s1)
print('--------     字典创建 series     ----------')
#使用字典创建，索引值为字典的 key 值
s2=pd.Series({'Longitude':39,'Latitude':116,'Temperature':23})
print('First value in s2:',s2['Longitude'])
print('--------  用序列作 series 索引  ----------')
#使用 range 函数生成的迭代序列设置索引值
s3=pd.Series([3.4,0.8,2.1,0.3,1.5],range(5,10))
print('First value in s3:',s3[5])
```

• 修改 series 数据对象

• 增加对象成员两个 series 对象可以通过 append()函数进行拼接

```
stiny=pd.Series({'humidity':84})
s4=s2.append(stiny)
```

• 利用 drop()函数删除对象成员、

(2)DataFrame 对象

• DataFrame 既有行索引也有列索引，所以 DataFrame 也可以看成是 Series 的容器

• DataFrame 是一个表格型的数据结构。列索引(columns)对应字段名，行索引(index)对应行号，值(values)是一个二维数组

• 创建 DataFrame 对象——基本方法是使用 DataFrame()函数构造，格式如下：

DataFrame([data, index, columns, dtype, copy])

```
import pandas as pd
#从字典构建
dict1={‘col1’:[1,2,3],‘col2’:[j,k,l]’}   #[1,2,3],[j,k,l]可用列表代替
df=pd.DataFrame(dict1)
df
#从数组构建
a=pd.DataFrame([[1,2,3],
                [4,5,6],
                [7,8,9]],colums=[“t1”,“t2”,“t3”])
a
```

• 访问 DataFrame 数据

```
import numpy as np
import pandas as pd
ser=pd.Series(np.arange(4),index=['A','B','C','D'])
data=pd.DataFrame(np.arange(16).reshape(4,4)
                  index=['BJ','SH','GZ','SZ']
                  columns=['q','r','s','t'])
print("ser['C']:",ser['C'])
print("ser[2]:",ser[2])
#访问某几列
print("data['q']:",data['q'])
print("data[['q','t']]:",data[['q','t']])
#数据切片与筛选
data=[:2]
data[data[‘s’]<=10]
#.loc[]，.iloc[]是分别按照索引名、下标切片
```

• 修改 DataFrame 数据

```
#修改指定位置
data[‘q’][‘BJ’]=8
#修改指定列
data[‘t’]=8
data
#增加列(只要索引名不存在就添加新列，否则修改列值)
data[‘u’]=8
data
#删除数据
#删除 index 值为‘SZ’的行
dt1=data.drop(‘SZ’,axis=0)
#删除‘r’‘u’列
dt2=data.drop([‘r’,‘u’],axis=1)
#从原数据中删除一行
data.drop(‘SZ’,inplace=True)
```

4、Matplotlib

(1)参数

figure 创建一个空白画布

add_subplot 创建子图

subplots 创建一系列子图

xlabel() x 轴名称

ylabel() y 轴名称

xlim() x 轴刻度范围

ylim() y 轴刻度范围

legend 指定图例

savefig 保存图形

show 显示图形

(2)**图表类型**

scatter（散点图）、bar（条形图）、pie（饼图）、hist（直方图）、plot（坐标图）

(3)例题

```
fig=plt.figure()
ax=fig.add_subplot(1,1,1)
rect=plt.Rectangle((0.2,0.75),0.4,0.15,color='r',alpha=0.3)#长方形
circ=plt.Circle((0.7,0.2),0.15,color='b',alpha=0.3)#椭圆形
pgon=plt.Polygon([[0.15,0.15],[0.35,0.4],[0.2,0.6]],color='g',alpha=0.9)#三角形
#一定要有，不然只是生成，没有增加到子图中
ax.add_patch(rect)
ax.add_patch(circ)
ax.add_patch(pgon)
plt.show()
```

第四章 机器学习基础

1、SVM 属于浅层模型，复杂的模型训练费时，并且容易产生过拟合

2、机器学习过程： 样本标注（收集数据），训练（学习），得到模型

机器学习的方法包括：损失函数、优化算法、模型、模型评估指标等要素。

评估指标有：准确率、错误率、精确率、召回率、F-measure 指数、均方误差

3、**训练集、验证集和测试集合的区别**

(1)训练集：训练用的数据集

(2)测试集：测试用的数据集

(3)验证集：模型**训练**过程中单独留出的样本集，用于调整模型的超参数和初步评估模型的能力

4、**过拟合和欠拟合的区别**

(1)过拟合(过学习)，指模型过度学习了训练数据的固有关系。直观表现是算法在**训练集上表现好，但在测试集上表现不好，泛化性能差**(原因：训练集的数量级和模型复杂度不匹配)

避免过拟合：交叉验证法(循环估计)

(2)欠拟合(欠学习)，指模型没有学到训练数据的内在关系，**对样本的一般性质学习不足**。例如耳朵长度超过 56 毫米的是狗(原因：模型学习不足、模型过于简单)

5、**监督学习、非监督学习、强化学习的本质区别**

(1)监督学习：使用已有的数据进行学习的机器学习方法。如：KNN 分类算法

(2)非监督学习：直接对没有标记的训练数据进行建模学习，算法可以在缺乏经验数据下使用，如：K-Means 聚类算法

（**两者区别是建模的数据有没有标签**）

(3)强化学习(增强学习)：根据系统状态和优化目标进行自主学习，不需要预备知识，也不事先知道要采取什么动作，通过尝试，根据环境的反馈去确定哪一个动作获得最大受益。

注：核心是评价策略的优劣

- 根据是否依赖模型分为：基于模型的强化学习 和 无模型的强化学习
- 根据回报函数分为：正向强化学习（回报函数人为指定的） 和逆向强化学习（无法指定）

6、评估指标计算

- 精确率=正确识别的个体总数/预测出的个体总数

  P(cat)=2/4=0.5，P(dog)=1/1=1，P(rabbit)=0/1=1
- 召回率=正确识别的个体总数/测试集中存在的个体总数

  R(cat)=2/2=1，R(dog)=1/2=0.5，R(rabbit)=0/2=0
- F-Measure 指数=2/(1/精确率+1/召回率)
- 错误率=分类错误的样本数/样本总数

  E(总)=(1+2)/6=0.5，E(cat)=0/2=0，E(dog)=1/2=0.5，E(rabbit)=2/2=1

真实结果	预测结果（只）		
	猫	狗	兔
猫	2	0	0
狗	0	1	1
兔	2	0	0

第五章 KNN 分类算法

1、重点：

(1)KNN 和 K-Means 算法区别

• K-means 本质上是无监督学习，而 KNN 是监督学习

• K-means 是聚类算法，KNN 是分类(或回归)算法

• K-means 算法把一个数据集分割成簇,使得形成的簇是同构的，每个簇里的点相互靠近。KNN 算法尝试基于其 k 个周围邻居来对未标记的观察进行分类。

(2)KNN 什么情况下分类最好？

一般 KNN 算法在**样本较少但典型性好**的情况下效果较好

2、KNN(K 邻近算法)核心：

如果一个样本在特征空间中的 k 个最相似（即特征空间中最邻近）的样本中的大多数属于某一类，则该样本也属于这个类别。

• k 值过小，近似误差变小了但是模型变得复杂，容易过拟合

• k 值过大，会导致学习误差增大，导致分类迷糊，即欠拟合

3、习题 1——鸢尾花分类

```
from sklearn import datasets
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
#导入鸢尾花数据并查看数据特征
iris = datasets.load_iris()
print('数据集结构：',iris.data.shape)
# 获取属性
iris_X = iris.data
# 获取类别
iris_y = iris.target
# 划分成测试集和训练集
iris_train_X,iris_test_X,iris_train_y,iris_test_y=train_test_split(iris_X,iris_y,test_size=0.2,
random_state=0)
#分类器初始化
knn = KNeighborsClassifier()
#对训练集进行训练
knn.fit(iris_train_X, iris_train_y)
#对测试集数据的鸢尾花类型进行预测
predict_result = knn.predict(iris_test_X)
print('测试集大小：',iris_test_X.shape)
print('预测结果：',predict_result)
print('预测精确率：',knn.score(iris_test_X, iris_test_y))
```

```
数据集结构： (150, 4)
测试集大小： (30, 4)
真实结果： [2 1 0 2 0 2 0 1 1 1 2 1 1 1 1 0 1 1 0 0 2 1 0 0 2 0 0 1 1 0]
预测结果： [2 1 0 2 0 2 0 1 1 1 2 1 1 1 2 0 1 1 0 0 2 1 0 0 2 0 0 1 1 0]
预测精确率： 0.9666666666666667
```

4、习题 2——使用 SKlearn 的 neighbors 模块，对水果数据进行 KNN 分类，然后预测下面两只 A、B 两只水果的类别

样本	mass	width	height	color_score
A	192	8.4	7.3	0.55
B	200	7.3	10.5	0.72

```
import numpy as np
from sklearn import neighbors

tmp = np.loadtxt("fruit_data.txt")
clf = neighbors.KNeighborsClassifier(3, weights='distance')
data=tmp[:,1:]
labels=tmp[:,0]
clf.fit(data,labels)
r=clf.predict([[192,8.4,7.3,0.55]])   #对 A 进行预测
print('KNN result is:',int(r))
r=clf.predict([[200,7.3,10.5,0.72]])   #对 B 进行预测
print('KNN result is:',int(r))
```

上面的程序运行后，可以得到对水果 A 和 B 的预测结果。水果 A 的预测结果为 1（苹果），水果 B 的预测结果为 4（柠檬）。

第六章 K-Means 聚类算法

1、聚类和分类的区别

(1)分类：监督、有标签

(2)聚类：无监督、训练数据无标签，**最终目标是获得紧凑、独立的簇集合**(簇中心也叫聚类中心)

2、聚类

(1)聚类是指将不同的对象划分成由多个对象组成的多个类的过程。由聚类产生的数据分组，同一组内的对象具有相似性，不同组内的对象具有相异性

(2)核心：将数据对象集合按相似度划分成多个类。得到的每个类称为聚类的**簇**

(3)优点：可以发现新知识、新规律

(4)相似度：作为聚类的依据，两个对象距离越近，相似度越大

(5)K-Means 聚类算法的类别划分依赖于样本之间的距离

- 选取 k 个数据点作为簇中心（种子）
- 逐个计算各数据点到簇中心的距离，把它分配到离它最近的簇
- 重复上述过程，直到分配不在发生变化或簇中心不在变。

算法核心：基于簇中心的距离进行分类

注：距离有欧式距离、曼哈顿距离

(6)聚类算法的理想目标是：类内距离最小，类间距离最大

(7)聚类算法性能评估：

- 外部有效评价——反应聚类结果的整体直观效果
- 内部有效评价——利用数据集的内部特征来评价
- 相关性测试评价——选定某个评价指标

注：
- 聚类算法的生成的类别不是原始数据集中的标签，是系统自动生成的
- 聚类还可以用于文本分析领域

3、K-means

(1)不足：

- k 值选定较难
- 初始聚类中心的选择对聚类结果影响较大
- 算法时间开销较大
- 算法功能具有局限性

(2)科学确定 k 值

- 经验值
- 观测值
- 肘部方法——**拐点位置为模型参数的关键**

注：如果数据不是很聚类的话，肘部方法的效果会变差，不聚类的数据会生成一条平滑的曲线，k 的最佳值将不清楚。

- 性能指标法

(3)使用后处理提高聚类效果——误差平方和 SSE

- SSE 越小，表明该簇离散程度越低，聚类效果越好
- SSE 计算：统计每个点到**所属簇中心**的距离的平方和

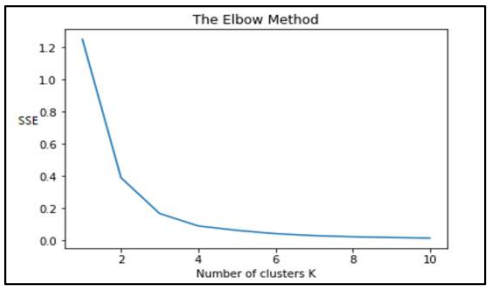
8、习题 1：银行客户分组画像

步骤 1：获得数据

```
# -*- coding: utf-8 -*-
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
# 客户存款、年龄数据集
dataset=pd.read_csv('Customer_Info.csv')
X=dataset.iloc[: , [4,3]].values
```

步骤 2：使用肘部方法找到最优的簇数

```
from sklearn.cluster import KMeans
sumDS = []
for i in range(1, 11):
    kmeans=KMeans(n_clusters=i)
    kmeans.fit(X)
    sumDS.append(kmeans.inertia_) #样本到簇中心的距离平方和
    #print(kmeans.inertia_)  #数值为 10 的 11 次方-le11
plt.plot(range(1, 11),sumDS)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters K')
plt.ylabel('SSE')
plt.show()
```



从上面的肘部折线图中可看出，k 取 3 或 4 合适

步骤 3：在数据集上使用 k=3 进行聚类

```
kmeans=KMeans(n_clusters=3, init='k-means++', max_iter= 300, n_init= 10, random_state= 0)
y_kmeans=kmeans.fit_predict(X)

# 集群可视化
plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s = 100, marker='^', c = 'red', label='Not very rich')
plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], s = 100, marker='o', c = 'green', label='Middle')
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s = 100, marker='*', c = 'blue', label='Rich')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 250, c = 'yellow', label='Centroids')
plt.title('Clusters of customer Info')
plt.xlabel('Deposit    ')
plt.ylabel('Age')
plt.legend()
plt.show()
```

第七章 推荐算法

1、推荐系统的评价指标

- 用户信任度
- 预测准确度
- 覆盖率
- 多样性
- 实时性

2、协同过滤推荐算法

(1)基于用户的协同过滤

如果两个用户对一些项目的评分相似，则他们对其他项目的评分也具有相似性

(2)基于物品的协同过滤

3、基于内容的推荐算法

分析系统的历史数据，提取对象的内容特征和用户的兴趣偏好。对被推荐对象，先和用户的兴趣偏好相匹配，再根据内容之间的关联程度，将关联度高的内容推荐给用户

(1)优势：

- 不需要大量数据
- 方法简单、有效，推荐结果直观，容易理解，不需要领域知识
- 不存在稀疏问题

(2)缺点：

- 对物品内容进行解析时，受到对象特征提取能力的限制
- 推荐结果相对固化，难发现新内容
- 用户兴趣模型与推荐对象模型之间的兼容问题，比如模式、语言等是否一致对信息匹配非常关键

4、习题 1：使用 KNN 算法推荐图书

```
# -*- coding: utf-8 -*-
import numpy as np
from sklearn import neighbors
knn = neighbors.KNeighborsClassifier(1)     #取得 knn 分类器
data = np.array([[1.1, 1.5, 1.4, 0.2],
                 [1.9, 1.0, 1.4 ,0.2],
                 [1.7, 1.2, 1.3, 0.2],
                 [2.6, 2.1 ,1.5, 0.2],
                 [2.0, 2.6 ,1.4, 0.2]])
labels = np.array(['A','B','C','D','E'])
knn.fit(data,labels)
print("预测结果：",knn.predict(np.array([[1.6, 1.5, 1.2, 0.1]]).reshape(1,-1)))
```

执行结果是 C，即表明用户 C 与用户 F 更相似，那么用户 C 喜欢过的而用户 F 未喜欢过的书就可以列入推荐列表

注：reshape(1,-1)表示一维数据在列上的变化

5、习题 2：计算用户 U2 对物品 I2 的评分

物品 / 用户	I1	I2	I3	I4
U1	5	-	4	4
U2	3	0	3	3
U3	2	5	2	1
U4	4	3	5	4

先计算$U1$与各个用户的相似度，使用欧式距离：

$$d(U1,U2)=\sqrt{(5-3)^2+(4-3)^2+(4-3)^2}=\sqrt{6}$$
$$d(U1,U3)=\sqrt{(5-2)^2+(4-2)^2+(4-1)^2}=\sqrt{22}$$
$$d(U1,U4)=\sqrt{(5-4)^2+(4-5)^2+(4-4)^2}=\sqrt{2}$$
$$sim(U1,U2)=1/(1+\sqrt{6})\approx 0.29$$
$$sim(U1,U3)=1/(1+\sqrt{22})\approx 0.18$$
$$sim(U1,U4)=1/(1+\sqrt{2})\approx 0.41$$

各个用户对物品评价均值为：

$$\overline{r_{U1}}=(5+4+4)/3=4.33$$
$$\overline{r_{U2}}=(3+0+3+3)/4=2.75$$
$$\overline{r_{U3}}=(2+5+2+1)/4=2.5$$
$$\overline{r_{U4}}=(4+3+5+4)/4=4$$

预测用户$U1$对物品$I2$的评价为：

$$r(U1,I2)=\overline{r_{U1}}+\frac{\sum[sim(U1,U)*(r(U,I2)-\overline{r_U})]}{\sum sim(U1,U)}$$
$$=4.33+\frac{0.29*(0-2.75)+0.18*(5-2.5)+0.41*(3-4)}{0.29+0.18+0.41}$$

第八章 回归算法

1、回归分析属于监督学习算法

(1)按照变量的个数：一元回归分析和多元回归分析

(2)按照自变量和因变量的关系：线性回归分析和非线性回归分析

2、什么是逻辑回归？

逻辑回归是根据现有数据对分类边界线建立回归公式，以此进行分类。逻辑回归在线性回归模型的基础上，通过引入 Sigmoid 函数，将线性回归的输出值映射到(0,1)范围。使用阈值将结果转换成 0 或 1，能够完成两类问题的预测(线性回归输出的是一个值，而逻辑回归将值映射到(0,1)集合)

3、什么是 Sigmoid 函数？表达式和图像是什么样的？

(1)定义：

Logistic 函数的简单形式为：其中 x 的取值范围是$(-\infty,+\infty)$，而值域为$(0,1)$。由于 Logistic 函数的图形外形看起来像 S 形，因此 Logistic 函数经常被称为 Sigmoid 函数(S 形函数)

(2)表达式：

首先，我们已知线性回归的表达式为：

$$y = w_1x_1 + \cdots + w_nx_n + b$$

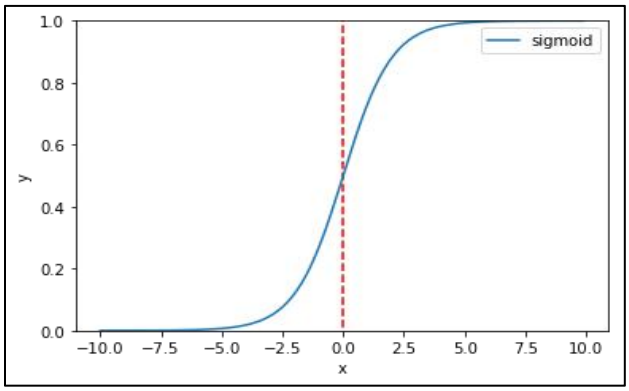
将y代入 Sigmoid 公式：

$$h(x) = \frac{1}{1+e^{-x}}$$

可以得到逻辑回归方程：

$$h(y) = \frac{1}{1+e^{-y}} = \frac{1}{1+e^{-(w_1x_1+\cdots+w_nx_n+b)}}$$

(3)图像：



第九章 支持向量机

1、支持向量机(属于监督学习)的核函数包括：线性核、多项式核、径向积核、S 型函数核
注：核函数作用是将低维特征映射到高维特征空间了，解决非线性问题

2、SVM 与最佳超平面有什么关系？SVM 的主要目标是寻找最佳超平面(最大间隔，最优解)
注：最优解对应的两侧虚线所穿过的样本点就是 SVM 中的支持向量

3、当样本被检测出异常值，怎么处理？
- 删除含有异常值的记录
- 视为缺失值
- 平均值修正：用前后两个观测值的平均值修正该异常值
- 不处理：直接在具有异常值的数据集上进行数据挖掘

4、给出两个点，找出最优分类面方程——中垂线/面

5、SVM 的优缺点：
(1)优点
- 小样本：并不是需要很少的样本，而是与问题的复杂程度比起来，需要的样本数量相对少
- 在高维空间中有效：样本的维度很高的情况下也可以处理
- 非线性：SVM 擅长处理非线性问题，主要通过核函数和惩罚变量完成
- 理论基础简单，分类效果较好
- 通用性好，可以自定义核函数

(2)缺点
- 计算复杂度高，对大规模数据的训练困难
- 不支持多分类，多分类问题需要间接实现
- 对不同的核函数敏感

6、收敛速度取决于权向量的初始值和学习率λ(每一次更新参数的程度大小)，学习率太大导致震荡、太小导致收敛过程很慢。

7、惩罚参数 C 越大，对误分类的惩罚增大，训练集上准确率高，泛化能力弱；越小，对误分类的惩罚减小，容错能力强，泛化能力强

8、问题 1(P229 例题 9.3)

(1)问题描述：

假设有三个样本，特征坐标分别为(2,0),(1,1),(2,3)，标签依次为 0、0、1。使用 SVC 模型建立分类器，并预测数据点(2,0)的类别。

(2)代码：

```
#-*- coding: utf-8 -*-
from sklearn import svm

# 样本特征
x = [[2, 0], [1, 1], [2, 3]]
# 样本的标签
y = [0, 0, 1]
# 建立 SVC 分类器
clf = svm.SVC(kernel='linear')
# 训练模型
clf.fit(x, y)
print(clf)
# 获得支持向量
print(clf.support_vectors_)
# 获得支持向量点在原数据中的下标
print(clf.support_)
# 获得每个类支持向量的个数
print(clf.n_support_)
# 预测(2,0)的类别
print( clf.predict( [[2, 0]] ) )
```

(3)运行结果

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto', kernel='linear',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
[[1. 1.]
 [2. 3.]]
[1 2]
[1 1]
[0]
```

9、问题 2(使用 SVM 解决非线性分类问题——moons 数据集分类)

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_moons
from sklearn.preprocessing import PolynomialFeatures
from sklearn.preprocessing import StandardScaler
from sklearn.svm import LinearSVC
from sklearn.pipeline import Pipeline
```

```
# 生成半环形数据
X, y = make_moons(n_samples=100, noise=0.1, random_state=1)
moonAxe=[-1.5, 2.5, -1, 1.5]          #moons 数据集的区间

# 显示数据样本
def dispData(x, y, moonAxe):
      pos_x0=[x[i,0]for i in range(len(y)) if y[i]==1]
      pos_x1=[x[i,1]for i in range(len(y)) if y[i]==1]
      neg_x0=[x[i,0]for i in range(len(y)) if y[i]==0]
      neg_x1=[x[i,1]for i in range(len(y)) if y[i]==0]

      plt.plot(pos_x0, pos_x1, \"bo\")
      plt.plot(neg_x0, neg_x1, \"r^\")

      plt.axis(moonAxe)
      plt.xlabel(\"x\")
      plt.ylabel(\"y\")

# 显示决策线
def dispPredict(clf, moonAxe):
      #生成区间内的数据
      d0 = np.linspace(moonAxe[0], moonAxe[1], 200)
      d1 = np.linspace(moonAxe[2], moonAxe[3], 200)
      x0, x1 = np.meshgrid(d0,d1)
      X = np.c_[x0.ravel(), x1.ravel()]
      #进行预测并绘制预测结果
      y_pred = clf.predict(X).reshape(x0.shape)
      plt.contourf(x0, x1, y_pred, alpha=0.8)

# 1.显示样本
dispData(X, y, moonAxe)
# 2.构建模型组合，整合三个函数
polynomial_svm_clf=Pipeline(
                                    ((\"multiFeature\",PolynomialFeatures(degree=3)),
                                     (\"NumScale\",StandardScaler()),
                                     (\"SVC\",LinearSVC(C=100)))
                                  )
# 3.使用模型组合进行训练
polynomial_svm_clf.fit(X,y)
# 4.显示分类线
dispPredict(polynomial_svm_clf, moonAxe)
# 5.显示图表标题
plt.title('Linear SVM classifies Moons data')
plt.show()
```

第十章 神经网络

1、第一个神经网络模型——M-P 模型

2、如果一个神经网络中，每一层中每个神经元都和下一层的所有神经元相连，则这个神经网络是全连接的（full connected），称为全连接神经网络

3、BP 神经网络的功能：分类和回归；两个最重要的要素：网络结构、神经元模型

4、BP 神经网络的学习过程

(1)神经网络模型的初始化，为各个连接权重赋予初始值、设定内部函数、设定误差函数、给定预期精度、设置最大迭代次数等

(2)将数据集输入神经网络，计算输出结果

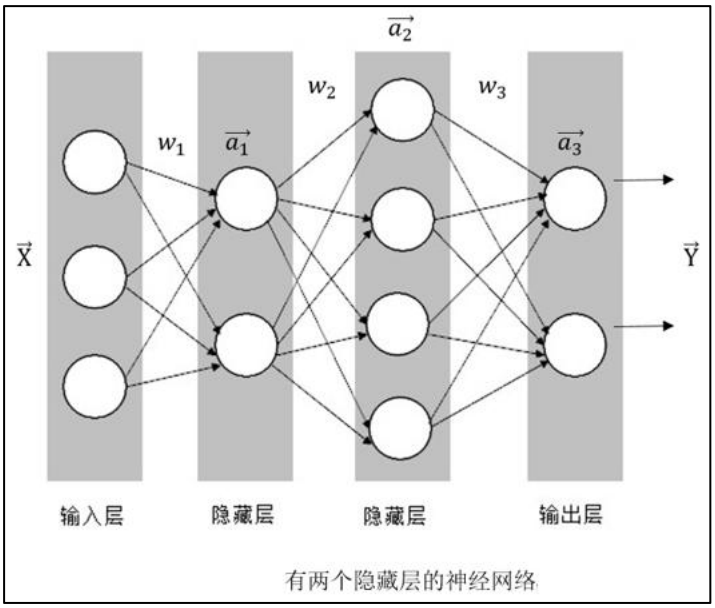
(3)求输出结果与期望值的差，作为误差

(4)将误差回传到与输出层相邻的隐藏层，同时根据误差减小的目标，依次调整各个连接权重，然后依次回传，直到第一个隐藏层

(5)使用新的权重作为神经网络的参数，重复步骤 2~4，使误差逐渐降低，达到预期精度

5、求参数个数

输入层到隐藏层的边数+隐藏层到隐藏层的边数+隐藏层到输出层的边数+隐藏层神经元数+输出层神经元数(下图为 6+8+8+2+4+2=30)



有两个隐藏层的神经网络

第十一章 深度学习

1、卷积能够实现图像的基本模糊、锐化、降低噪声、提取边缘特征等功能

2、什么是池化？两种方法是什么？

为了提取不同类型的特征，卷积核一般为多个，但多个卷积核使得连接大大增加，从而导致数据维度上升。所以卷积之后通常进行池化操作。池化是将图像按子区域进行压缩的操作，具有平移不变性

一般有两种方法：最大池化(线条加粗，突出纹理)和平均池化(可去噪声，突出背景)

注：卷积核↑→连接↑→维度↑

3、卷积神经网络 CNN

4、卷积神经网络中会有多个卷积核以及多个池化层，其中每一个卷积核的大小是不同的，VGG 有 13 个卷积核，每个卷积核大小都不同，可以是 3*3、5*5

重点

一、选择题

1、以下关于人工智能的说法中，错误的是(A)

A.人工智能是研究世界运行规律的科学

B.人工智能涵盖多个学科领城

C.人工智能包括自动推理、专家系统.机器学习等技术

D.现阶段的人工智能核心是机器学习

2、人工智能未来发展的三个层次包括(D)

A.弱人工智能

B.强人工智能

C.超人工智能

D.以上全对

3、下面能实现人工智能算法的开发环境有(D)

A. C 语言

B. Java 语言

C. Python 语言

D.以上都可以

4、20 世纪 70 年代开始，人工智能进人首次低谷期的原因不包括(B)

A.计算机内存有限

B.摄像设备没有出现

C.计算机处理速度不够快

D.理论基础薄弱

5、被广泛认为是 AI 诞生的标志的是(C)

A.计算机的产生

B.图灵机的出现

C.达特茅斯会议

D.神经网络的提出

6、机器学习是研究如何使用计算机(C)的一门学科，

A.模拟生物行为

B.模拟人类解决问题

C.模拟人类学习活动

D.模拟人类生产活动

7、使用小括号定义的数据类型是(D)

A.列表

B.集合

C.字典

D.元组

8、使用大括号定义的数据类型是(D)
A.字典
B.集合
C.列表
D.字典或集合

9、以下关于字典中的键值的说法，正确的是(D)
A.键值不可修改
B.键值不能重复
C.键值必须是字符串
D.以上都不对

10、以下描述中,属于集合特点的是(A)
A.集合中的数据是无序的
B.集合中的数据是可以重复的
C.集合中的数据是严格有序的
D.集合中必须嵌套一个子集合

11、机器学习研究的目标有三个，不包括(D)
A.人类学习过程的认知模型
B.通用学习算住
C.构造面向任务的专用学习系统
D. 制作长相接近人类的机器系统

12、按学习方式划分，机器学习通常分为(C)
A.监督学习、非监督学习、聚类
B 监督学习、非监督学习、神经网络
C.监督学习、非监督学习、强化学习
D.监督学习、非监督学习、有数师学习

13、下面关于非监督学习算法的说法，正确的是(C)
A.数据要是成对的
B.算法准确本非常高
C.没有经验数据可供学习
D.需要一定的经验数据

14、强化学习(D)
A.也称为有教师学习
B.需要经验数据
C.数据要是成对的
D.不需要预备知识

15、机器学习模型包括四个组成部分，不包含(A)
A.模型结构(应该是**环境**)
B.知识库
C.学习单元
D.执行单元

16、关于机器学习模型中的数据。以下说法正确的是(C)
A.数据越多越好
B.数据只要质量好，这少越好
C.数据的数量和质量都很重要
D.模型选择最重要，数据影响不大

17、(B)是指机器学习算法对新鲜样本的适应能力。
A.模型测试
B.泛化能力
C.过拟合
D.模型训练

18.训练集、验证集和测试集在使用过程中的顺序是(B)
A.测试集、训练集、验证集
B.训练集、测试集、验证集
C.验证集、训练集、测试集
D.训练集、验证集、测试集

19.关于过拟合的说法,正确的是(D)
A.指模型学习不足
B.会使得模型泛化能力高
C.会强化欠拟合
D.可以通过交叉验证改善

二、填空题
1、神经网络两个最重要的要素是：**网络结构、神经元模型**
2、给出两个点，如何找出最优分类面方程：**中垂线/面**
3、SVM 与最佳超平面有什么关系：SVM 的主要目标是**寻找最佳超平面**
4、支持向量机的核函数有：**线性核、多项式核、径向积核、S 型函数核**
5、Sigmoid 函数的表达式为：$h(x) = 1/(1 + e^{-x})$
6、KNN 什么情况下分类最好？**样本较少但典型性好**
7、机器学习过程：**样本标注（收集数据），训练（学习），得到模型**
8、机器学习的方法包括：**损失函数、优化算法、模型、模型评估指标**等要素
9、评估指标有：**准确率、错误率、精确率、召回率、F-measure 指数、均方误差**
10、图表类型有：**scatter（散点图）、bar（条形图）、pie（饼图）、hist（直方图）、plot（坐标图）**
11、智能是人类**智力**和**能力**的总称
12、人工智能的英文缩写为 **AI**， 是研究人类智能的技术科学。

13、**1956** 年，人工智能诞生于美国达特茅斯的一次夏季学术研讨会
14、20 世纪 80 年代第二次人工智能浪潮中，最为代表性的系统是**专家系统**
15、在人工智能领域，**NLP** 的全称是自然语言处理
16、公司为方便员工考勤，安装了种考勤设备，员工只罗将某个特定的手指放在设备上即可实现专勤。这个设备使用了人工智能中的**指纹识别**技术
17、首先要收集大量样本图像，并标明这些图像的类别，这个过程称为**样本标注**
18、把样本和标注送给算法学习的处理称为模型的**训练**
19、以多隐层神经网络为代表的深度学习模型近年来得到快速的发展，属于**深层**(浅层/深层)模型
20、随机森林是一种**集成**(单一/集成)学习算法。
21、**监督**学习在学习的时候需要标签，也称为有教师学习。

三、简答题
1、什么是池化？两种方法是什么？
(1)为了提取不同类型的特征，卷积核一般为多个，但多个卷积核使得连接大大增加，从而导致数据维度上升。所以卷积之后通常进行池化操作。池化是将图像按子区域进行压缩的操作，具有平移不变性
(2)一般有两种方法：最大池化(线条加粗，突出纹理)和平均池化(可去噪声，突出背景)\

2、SVM 的优缺点是什么
(1)优点
- 小样本：并不是需要很少的样本，而是与问题的复杂程度比起来，需要的样本数量相对少
- 在高维空间中有效：样本的维度很高的情况下也可以处理
- 非线性：SVM 擅长处理非线性问题，主要通过核函数和惩罚变量完成
- 理论基础简单，分类效果较好
- 通用性好，可以自定义核函数

(2)缺点
- 计算复杂度高，对大规模数据的训练困难
- 不支持多分类，多分类问题需要间接实现
- 对不同的核函数敏感

3、当样本被检测出异常值，怎么处理？
- 删除含有异常值的记录
- 视为缺失值
- 平均值修正：用前后两个观测值的平均值修正该异常值
- 不处理：直接在具有异常值的数据集上进行数据挖掘

4、聚类和分类的区别是什么？
(1)分类：监督、有标签
(2)聚类：无监督、训练数据无标签，**最终目标是获得紧凑、独立的簇集合**(簇中心也叫聚类中心)

5、训练集、验证集和测试集合的区别

(1)训练集：训练用的数据集

(2)测试集：测试用的数据集

(3)验证集：模型训练过程中单独留出的样本集，用于调整模型的超参数和初步评估模型的能力

6、过拟合和欠拟合的区别

(1)过拟合(过学习)，指模型过度学习了训练数据的固有关系。直观表现是算法在训练集上表现好，但在测试集上表现不好，泛化性能差(原因：训练集的数量级和模型复杂度不匹配)

避免过拟合：交叉验证法(循环估计)

(2)欠拟合(欠学习)，指模型没有学到训练数据的内在关系，对样本的一般性质学习不足。例如耳朵长度超过 56 毫米的是狗(原因：模型学习不足、模型过于简单)

7、监督学习、非监督学习、强化学习的本质区别

(1)监督学习：使用已有的数据进行学习的机器学习方法。如：KNN 分类算法

(2)非监督学习：直接对没有标记的训练数据进行建模学习，算法可以在缺乏经验数据下使用，如：K-Means 聚类算法

（两者区别是建模的数据有没有标签）

(3)强化学习(增强学习)：根据系统状态和优化目标进行自主学习，不需要预备知识，也不事先知道要采取什么动作，通过尝试，根据环境的反馈去确定哪一个动作获得最大受益。

四、计算题

计算用户 U2 对物品 I2 的评分

物品 / 用户	I1	I2	I3	I4
U1	5	-	4	4
U2	3	0	3	3
U3	2	5	2	1
U4	4	3	5	4

先计算$U1$与各个用户的相似度，使用欧式距离：

$$d(U1,U2)=\sqrt{(5-3)^2+(4-3)^2+(4-3)^2}=\sqrt{6}$$
$$d(U1,U3)=\sqrt{(5-2)^2+(4-2)^2+(4-1)^2}=\sqrt{22}$$
$$d(U1,U4)=\sqrt{(5-4)^2+(4-5)^2+(4-4)^2}=\sqrt{2}$$
$$sim(U1,U2)=1/(1+\sqrt{6})\approx 0.29$$
$$sim(U1,U3)=1/(1+\sqrt{22})\approx 0.18$$
$$sim(U1,U4)=1/(1+\sqrt{2})\approx 0.41$$

各个用户对物品评价均值为：

$$\overline{r_{U1}}=(5+4+4)/3=4.33$$
$$\overline{r_{U2}}=(3+0+3+3)/4=2.75$$
$$\overline{r_{U3}}=(2+5+2+1)/4=2.5$$
$$\overline{r_{U4}}=(4+3+5+4)/4=4$$

预测用户$U1$对物品$I2$的评价为：

$$r(U1,I2)=\overline{r_{U1}}+\frac{\sum[sim(U1,U)*(r(U,I2)-\overline{r_U})]}{\sum sim(U1,U)}$$
$$=4.33+\frac{0.29*(0-2.75)+0.18*(5-2.5)+0.41*(3-4)}{0.29+0.18+0.41}$$

五、编程题

(1)问题描述：

假设有三个样本，特征坐标分别为(2,0),(1,1),(2,3)，标签依次为 0、0、1。使用 SVC 模型建立分类器，并预测数据点(2,0)的类别。

(2)代码：

```
#-*- coding: utf-8 -*-
from sklearn import svm

# 样本特征
x = [[2, 0], [1, 1], [2, 3]]
# 样本的标签
y = [0, 0, 1]
# 建立 SVC 分类器
clf = svm.SVC(kernel='linear')
# 训练模型
clf.fit(x, y)
print(clf)
# 获得支持向量
print(clf.support_vectors_)
# 获得支持向量点在原数据中的下标
print(clf.support_)
# 获得每个类支持向量的个数
print(clf.n_support_)
# 预测(2,0)的类别
print( clf.predict( [[2, 0]] ) )
```

(3)运行结果

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto', kernel='linear',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
[[1. 1.]
 [2. 3.]]
[1 2]
[1 1]
[0]
```