# 1.主成分分析

|  |
| --- |
|  |
|  |

# 2.计算案例

|  |
| --- |
|  |
|  |
| 下面这种降维损失很大 |
|  |
| 用斜线,损失就比较小 |
|  |
|  |

# 3.使用的API模块

|  |
| --- |
|  |

## 小例子

|  |
| --- |
| from sklearn.decomposition import PCA  data = [[2,8,4,5],[6,3,0,8],[5,4,9,1]]  transfer = PCA(n\_components=0.9) data\_new = transfer.fit\_transform(data) print(data\_new) |

### 效果

|  |
| --- |
|  |

### 然后我们xiug 一下代码

|  |
| --- |
| from sklearn.decomposition import PCA  data = [[2,8,4,5],[6,3,0,8],[5,4,9,1]] # 3行4列,说明有4个特征  # transfer = PCA(n\_components=0.9) transfer = PCA(n\_components=2) # 设置特征数为2 data\_new = transfer.fit\_transform(data) print(data\_new) |

### 效果

|  |
| --- |
|  |

## 案例

|  |
| --- |
|  |
|  |
|  |

### 数据处理流程

|  |
| --- |
|  |

### 注意:这个案例的原始数据太多,我们需要把它减少一点,方法是把原来数据的一小部分复制到另外一个文件,把order\_products\_\_prior.csv复制为order\_products\_\_prior2.csv.把orders.csv复制为orders2.csv这样子数据就少很多,当然我们还是需要保留原来的文件,方便以后学习

### 程序名称:pca-project.py,代码如下,使用小一点的两个数据

|  |
| --- |
| # 主成分分析案例: instacart 市场篮子分析:消费者将购买哪些产品? # 这里有4个文件,我们需要找出用户和物品之间的关系,也就是user\_id和aisle的关系,但是他们分布在不同的表格里面我们需要想办法把他们 # 放到同一张表格里面,实际上,我们需要把用户id和其他表格的数据合成为一个表格.这样子才方便分析 from sklearn.decomposition import PCA import pandas as pd  # 1.加载数据 # 需要先设置pandas的数据显示格式否则有些列看不到 pd.set\_option("display.max\_columns",None) # 显示所有列 pd.set\_option("display.width",200) # 显示200列后换行 aisles = pd.read\_csv("./aisles.csv") opp = pd.read\_csv("./order\_products\_prior2.csv") orders = pd.read\_csv("./orders2.csv") products = pd.read\_csv("./products.csv") # print(aisles) # 2.合并表格 # 1>把aisle和procucts合并,这两张表都有aisle\_id需要这么写 table1 = pd.merge(aisles,products,on=["aisle\_id","aisle\_id"]) # 2>把table1 表格和opp表格合并,用product\_id作为连接的条件 table2 = pd.merge(table1,opp,on=["product\_id","product\_id"]) # 3>把table2和orders表格合并 table3 = pd.merge(table2,orders,on=["order\_id","order\_id"]) # print(table3) # 3.找到user\_id和aisle的关系,使用pandas.crosstab方法,也就是交叉表方法 table = pd.crosstab(table3["user\_id"],table3["aisle"]) # print(table) # 4.PCA降维 # 1>实例化转换器对象 transfer = PCA(n\_components=0.95) # n\_components使用小数的情况比使用整数的情况要多 # 2>调用fit\_transform方法 new\_data = transfer.fit\_transform(table) print(new\_data.shape) |

#### 结果

|  |
| --- |
|  |