# 1.特征工程

## 1>为什么需要特征工程

|  |
| --- |
|  |

## 2>什么是特征工程

|  |
| --- |
|  |

## 3>特征工程的位置与数据处理的比较

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

# 2.特征抽取

|  |
| --- |
|  |
|  |

## 就是把一个不方便机器学习处理的数据格式转化为方便机器学习处理的数据格式

|  |
| --- |
|  |

# 3.字典特征提取

|  |
| --- |
|  |

## 新版本的获取特征名称的API是get\_feature\_names\_out()

## 示例

|  |
| --- |
|  |

## 实例代码

|  |
| --- |
|  |

### 返回一个稀疏矩阵

|  |
| --- |
|  |

## 如果先看到我们前面提到的样子,需要把sparse设置为False

|  |
| --- |
| *""" 字典特征抽取,使用feature\_extraction.DictVectorizer类来构造一个对象 然后使用该对象的方法 [{'city':'北京','temperature':100}, {'city':'上海','temperature':60},{'city':'深圳','temperature':30}, ] """* from sklearn.feature\_extraction import DictVectorizer  data = [  {'city':'北京','temperature':100},  {'city':'上海','temperature':60},  {'city':'深圳','temperature':30}, ] # 1.实例化一个转换器类对象 # dictVectorizer = DictVectorizer(sparse=True) dictVectorizer = DictVectorizer(sparse=False) # 2.调用fit\_transform方法 ret = dictVectorizer.fit\_transform(data) print(ret) |

### 效果:

|  |
| --- |
|  |

### 其实稀疏矩阵是由好处的,它只显示非0值,这样子可以节省内存

# 4.One-hot编码

|  |
| --- |
|  |
|  |

# 5.字典特征提取的应用场景

|  |
| --- |
|  |