王茂霖：特征工程方法总结！

原创 王茂霖 Datawhale 5月1日

收录于话题
#王茂霖

11个

↑↑↑关注后"星标"Datawhale
每日干货 & 每月组队学习，不错过

Datawhale干货

作者：王茂霖，华中科技大学，Datawhale成员

内容概括

1.经典特征工程构造
2.特征工程案例实践

PPT完整下载：后台回复“**210501**”可获取

视频地址：https://www.bilibili.com/video/BV1sf4y1s7Fw

特征工程构造

TIANCHI天池 Datawhale

Part 1 经典特征工程构造

特征工程是数据科学最有创造力的部分

特征工程

TIANCHI天池 Datawhale

特征工程是数据科学中最有创造力的一部分

二分类问题：用逻辑回归，设计一个身材分类器。

输入数据X: **身高和体重** ；　　标签为Y: 瘦，不瘦。

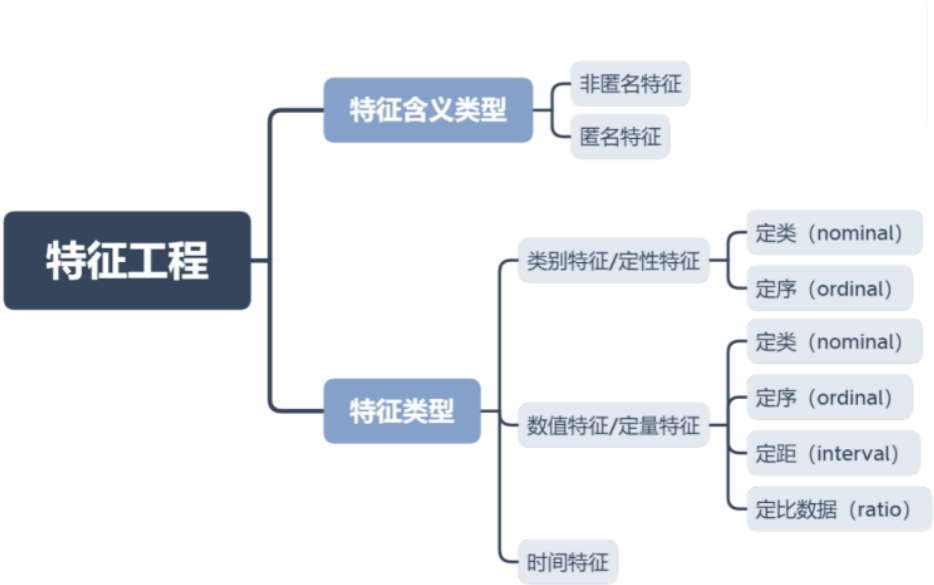
一个非常经典的特征工程是：**BMI指数，BMI=体重/(身高^2)。这样，通过BMI指数，就能非常显然地帮助我们，刻画一个人身材如何。甚至，你可以抛弃原始的体重和身高数据。**

Deep learning，作为一种强大的**自动化特征工程工具**

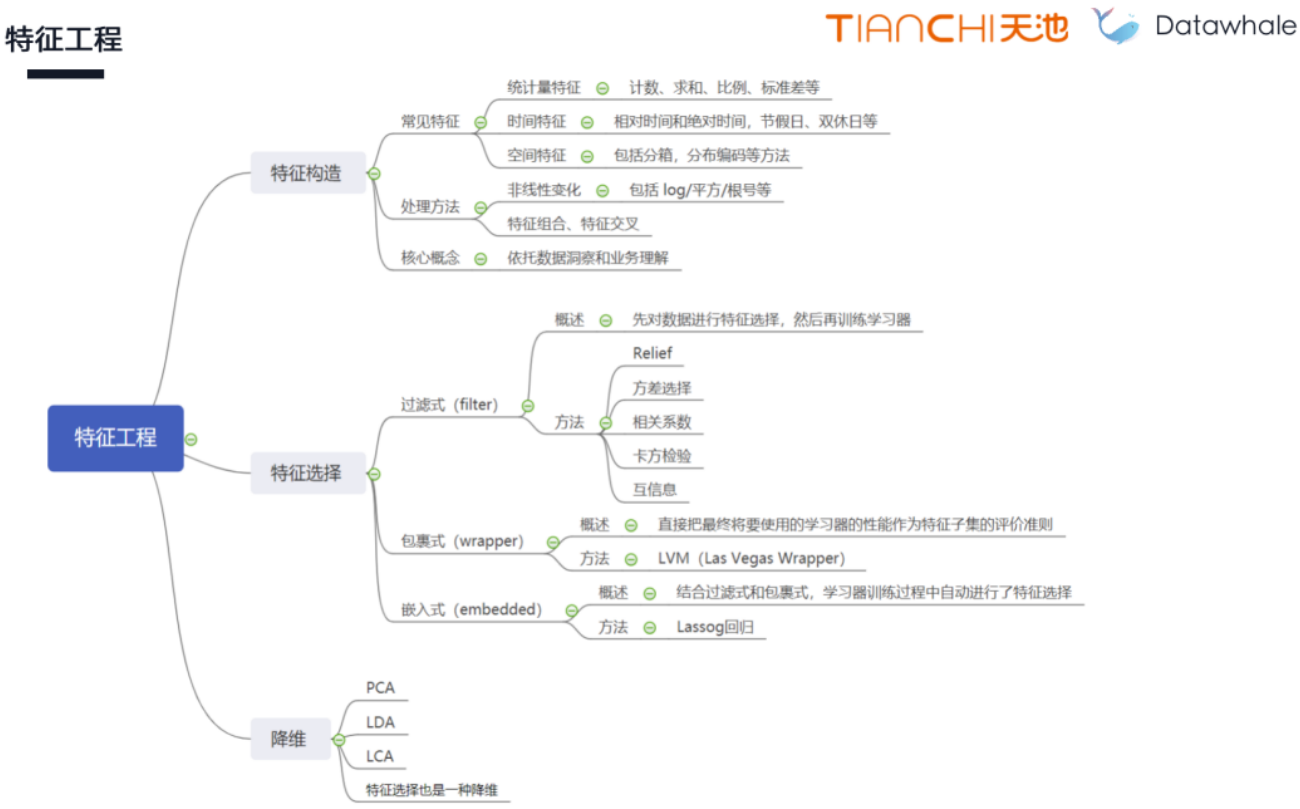
一、特征的类型汇总

特征工程

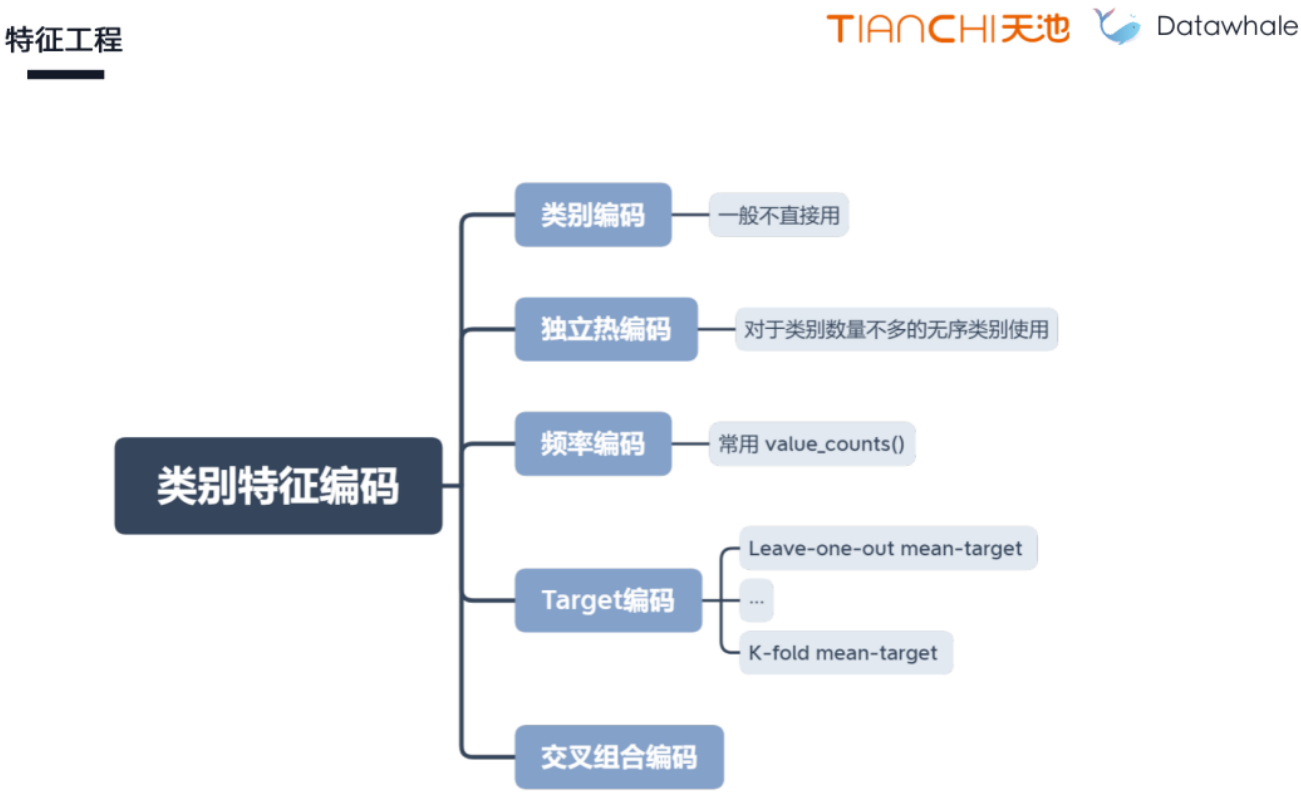
TIANCHI天池 Datawhale



二、特征工程方法总结



三、类别特征的常用编码方法



1.Label Encoder

特征工程：1.类别特征编码

TIANCHI天池 Datawhale

Label Encoder Label Encoder可以将类型为object的变量转变为数值形式

对应代码：preprocessing.LabelEncoder()

```
from sklearn import preprocessing
df = pd.DataFrame({'color':['red','blue','black','green']})
le = preprocessing.LabelEncoder()
le.fit(df['color'].values)
df['color_labelencode'] = le.transform(df['color'].values)
df
```

	color	color_labelencode
0	red	3
1	blue	1
2	black	0
3	green	2

2.One-Hot Encoder

特征工程：1.类别特征编码

TIANCHI天池 Datawhale

One-Hot Encoder One-Hot编码对于一个类别特征变量，我们对每个类别，使用二进制编码（0或1）创建一个新列（有时称为dummy变量），以表示特定行是否属于该类别

对应代码：pd.get_dummies(df['color'].values)

```
from sklearn import preprocessing
df = pd.DataFrame({'color':['red','blue','black','green']})
pd.get_dummies(df['color'].values)
```

	black	blue	green	red
0	0	0	0	1
1	0	1	0	0
2	1	0	0	0
3	0	0	1	0

3.Frequency 编码

特征工程：1.类别特征编码

TIANCHI天池 Datawhale

Frequency 编码

Frequency编码通过计算特征变量中每个值的出现次数来表示该特征的信息

对应代码：

```
from sklearn import preprocessing
df = pd.DataFrame({'color':['red','red','red','blue','blue','black','green','green','green']})
df['color_cnt'] = df['color'].map(df['color'].value_counts())
df
```

	color	color_cnt
0	red	3
1	red	3
2	red	3
3	blue	2
4	blue	2
5	black	1
6	green	3
7	green	3
8	green	3

```
# 计算某品牌的销售统计量，同学们还可以计算其他特征的统计量
# 这里要以 train 的数据计算统计量
train_gb = train.groupby("brand")
all_info = {}
for kind, kind_data in train_gb:
    info = {}
    kind_data = kind_data[kind_data['price'] > 0]
    info['brand_amount'] = len(kind_data)
    info['brand_price_max'] = kind_data.price.max()
    info['brand_price_median'] = kind_data.price.median()
    info['brand_price_min'] = kind_data.price.min()
    info['brand_price_sum'] = kind_data.price.sum()
    info['brand_price_std'] = kind_data.price.std()
    info['brand_price_average'] = round(kind_data.price.su
    all_info[kind] = info
brand_fe = pd.DataFrame(all_info).T.reset_index().rename(c
data = data.merge(brand_fe, how='left', on='brand')
```

4.Target 编码

特征工程：1.类别特征编码

TIANCHI天池 Datawhale

Target 编码 （标签统计编码）

target编码是06年提出的一种结合标签进行编码的技术，它将类别特征替换为从标签衍生而来的特征，在类别特征为高基数的时候非常有效。该技术在非常多的数据竞赛中都取得了非常好的效果，但特别需要注意过拟合的问题。

类似于留一验证和交叉验证

1. Leave-one-out mean-target 编码

2. K-fold mean-target 编码

对应代码：

```
from sklearn import preprocessing
from pandas import pandas
from category_encoders.leave_one_out import LeaveOneOutEncoder
df_tr = pd.DataFrame({'color':['red','red','red','red','red','red','black','black'], 'label':[1,0
df_te = pd.DataFrame({'color':['red','red','black'] })
```

```
loo = LeaveOneOutEncoder()
loo.fit_transform(df_tr['color'], df_tr['label'])
```

	color
0	0.6
1	0.8
2	0.6

```
loo.transform(df_te['color'])
```

	color
0	0.666667
1	0.666667
2	0.500000

5.其他编码

特征工程：1.类别特征编码-有序特征

TIANCHI天池 Datawhale

有序类别特征（有相对顺序的类别特征）

示例：
- 年龄段特征 (分箱特征)："1-10,11-20,21-30,31-40"年龄段；
- 评分等级特征：“A,B,C,D"；

字典编码

有序字典编码，就是将特征中的每个字段按照相对大小构建字典，再进行转化。

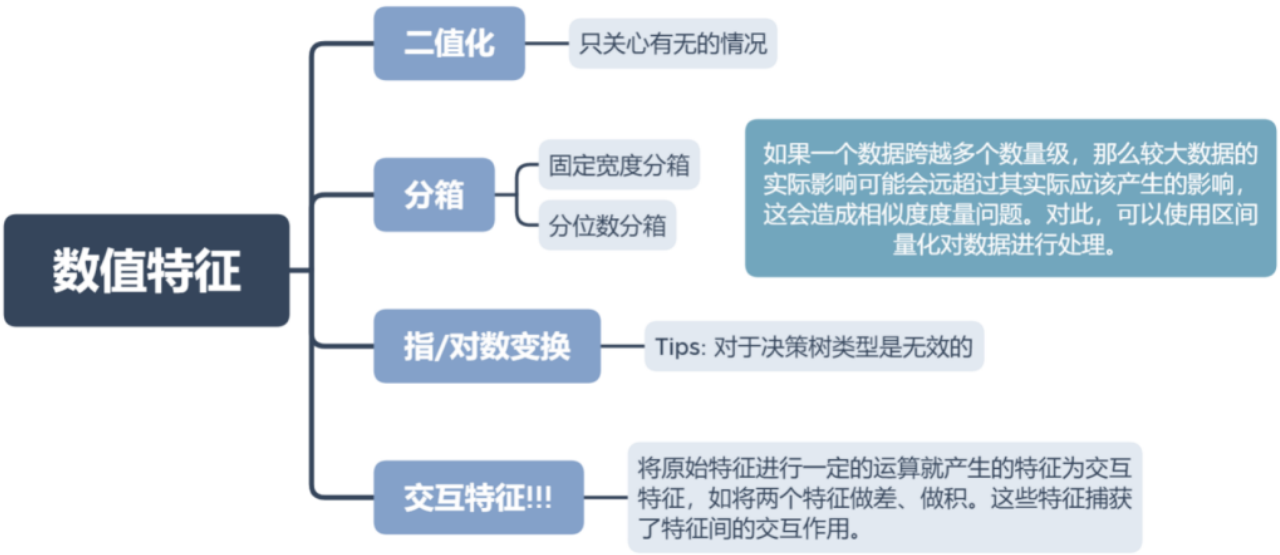
分段编码

分段聚类编码也是一种分箱的策略，它主要基于数据的相对大小并结合业务背景知识对类别特征进行分段分组重新编码。

四、数值特征的常用编码方法

特征工程：2.数值特征工程

TIANCHI天池 Datawhale

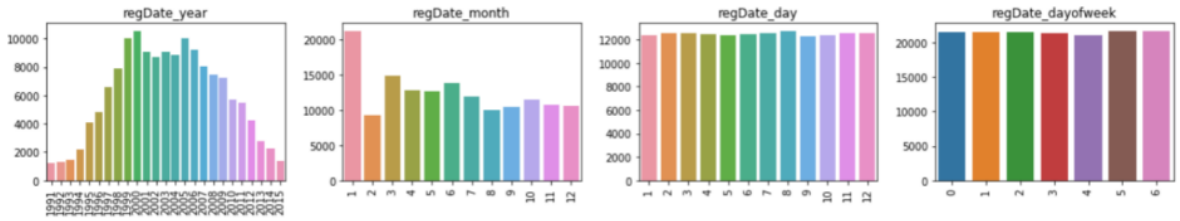


五、时间特征的常用编码方法

特征工程：3.时间特征工程

TIANCHI天池 Datawhale

1. 基础周期特征(年月日特征拆解)



2. 特殊周期特征(星期&节假日等)

3. 时间差：

```
# 使用时间：data['creatDate'] - data['regDate']，反应汽车使用时间，一般来说价格与使用时间成反比
# 不过要注意，数据里有时间出错的格式，所以我们需要 errors='coerce'
data['used_time'] = (pd.to_datetime(data['creatDate'], format='%Y%m%d', errors='coerce') -
                     pd.to_datetime(data['regDate'], format='%Y%m%d', errors='coerce')).dt.days
```

特征工程实践

TIANCHI天池 Datawhale

Part 2 特征工程案例实践

特征工程-示例

TIANCHI天池 Datawhale

Field	Description
SaleID	交易ID，唯一编码
name	汽车交易名称，已脱敏
regDate	汽车注册日期，例如20160101，2016年01月01日
model	车型编码，已脱敏
brand	汽车品牌，已脱敏
bodyType	车身类型：豪华轿车：0，微型车：1，厢型车：2，大巴车：3，敞篷车：4，双门汽车：5，商务车：6，搅拌车：7
fuelType	燃油类型：汽油：0，柴油：1，液化石油气：2，天然气：3，混合动力：4，其他：5，电动：6
gearbox	变速箱：手动：0，自动：1
power	发动机功率：范围 [0, 600]
kilometer	汽车已行驶公里，单位万km
notRepairedDamage	汽车有尚未修复的损坏：是：0，否：1
regionCode	地区编码，已脱敏
seller	销售方：个体：0，非个体：1
offerType	报价类型：提供：0，请求：1
creatDate	汽车上线时间，即开始售卖时间
price	二手车交易价格（预测目标）
v系列特征	匿名特征，包含v0-23在内24个匿名特征

1.特征构造

特征工程构建大概可以从三个方面入手：领域特征，交叉特征和多项式特征。

特征工程-特征构造

TIANCHI天池 Datawhale

特征工程构建大概可以从三个方面入手：领域特征；**交叉特征；多项式特征。**

统计特征常规操作，如count、ratio、nunique 等相关特征。

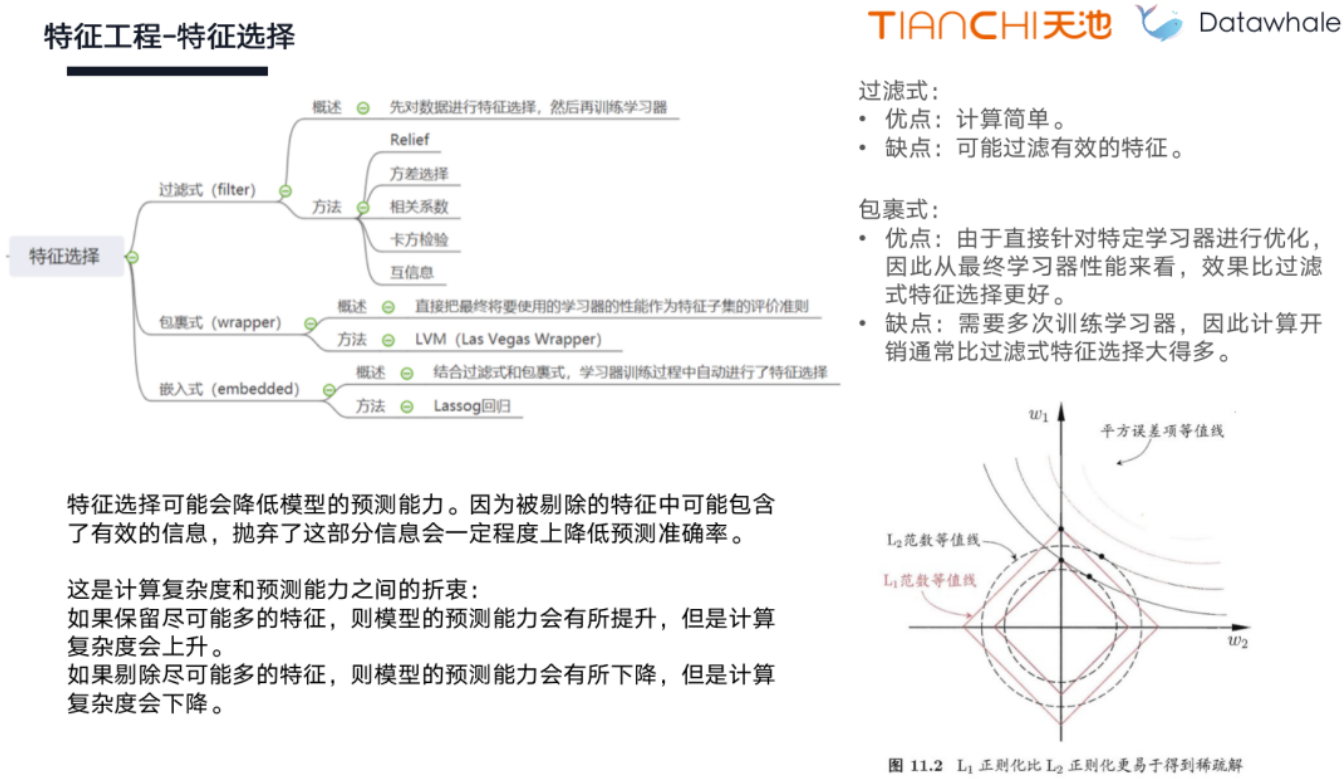
```
# 计算某品牌的销售统计量，这种特征不能细粒度的统计，防止发生信息过多的泄露
train_gb = Train_data.groupby("brand")
all_info = {}
for kind, kind_data in train_gb:
    info = {}
    kind_data = kind_data[kind_data['price'] > 0]
    info['brand_amount'] = len(kind_data)
    info['brand_price_max'] = kind_data.price.max()
    info['brand_price_median'] = kind_data.price.median()
    info['brand_price_min'] = kind_data.price.min()
    info['brand_price_sum'] = kind_data.price.sum()
    info['brand_price_std'] = kind_data.price.std()
    info['brand_price_average'] = round(kind_data.price.sum() / (len(kind_data) + 1), 2)
    all_info[kind] = info
brand_fe = pd.DataFrame(all_info).T.reset_index().rename(columns={"index": "brand"})
data = data.merge(brand_fe, how='left', on='brand')
```

```
### 类别特征的二阶交叉
for f_pair in tqdm([['model', 'brand'], ['model', 'regionCode'], ['brand', 'regionCode']]):
    ### 共现次数
    data['_'.join(f_pair) + '_count'] = data.groupby(f_pair)['SaleID'].transform('count')
    ### nunique、熵
    data = data.merge(data.groupby(f_pair[0], as_index=False)[f_pair[1]].agg({
        '{}_{}_nunique'.format(f_pair[0], f_pair[1]): 'nunique',
        '{}_{}_ent'.format(f_pair[0], f_pair[1]): lambda x: entropy(x.value_counts() / x.sha
    }), on=f_pair[0], how='left')
    data = data.merge(data.groupby(f_pair[1], as_index=False)[f_pair[0]].agg({
        '{}_{}_nunique'.format(f_pair[1], f_pair[0]): 'nunique',
        '{}_{}_ent'.format(f_pair[1], f_pair[0]): lambda x: entropy(x.value_counts() / x.sha
    }), on=f_pair[1], how='left')
    ### 比例偏好
    data['{}_in_{}_prop'.format(f_pair[0], f_pair[1])] = data['_'.join(f_pair) + '_count'] /
    data['{}_in_{}_prop'.format(f_pair[1], f_pair[0])] = data['_'.join(f_pair) + '_count'] /

    feat_cols.extend([
        '_'.join(f_pair) + '_count',
        '{}_{}_nunique'.format(f_pair[0], f_pair[1]), '{}_{}_ent'.format(f_pair[0], f_pair[1
        '{}_{}_nunique'.format(f_pair[1], f_pair[0]), '{}_{}_ent'.format(f_pair[1], f_pair[0
        '{}_in_{}_prop'.format(f_pair[0], f_pair[1]), '{}_in_{}_prop'.format(f_pair[1], f_pa
    ])
```

2.特征选择

特征选择可能会降低模型的预测能力。因为被剔除的特征中可能包含了有效的信息，抛弃了这部分信息会一定程度上降低预测准确率。



本文作者

王茂霖，Datawhale重要贡献成员，Datawhale&天池数据挖掘学习赛开源内容发起人，全网阅读超10w。

参赛30余次，获得DCIC-数字中国创新创业大赛亚军，全球城市计算AI挑战赛，Alibaba Cloud German AI Challenge等多项Top10。

分享地址

- 复制链接打开（或阅读原文）
- https://www.bilibili.com/video/BV1sf4y1s7Fw

整理不易，点赞三连↓