**打卡三-特征工程的构建**

**特征工程主要包括以下几步：**

**特征预处理、缺失值、异常值处理**

**1）特征预处理：**

**数据预处理部分一般我们要处理一些EDA阶段分析出来的问题，这里介绍了数据缺失值的填充，时间格式特征的转化处理，某些对象类别特征的处理。**

**首先我们查找出数据中的对象特征和数值特征：**

**numerical\_fea = list(data\_train.select\_dtypes(exclude=['object']).columns)**

**category\_fea = list(filter(lambda x: x not in numerical\_fea,list(data\_train.columns)))**

**label = 'isDefault'**

**numerical\_fea.remove(label)**

**2）缺失值处理：**

**首先我们必须查看缺失值的情况：**

**#查看缺失值情况**

**data\_train.isnull().sum()**

**然后我们可以有几种选择策略：**

1. **重新采集数据，但对于一个比赛这明显不现实**
2. **数据填充，一个方向是根据专业知识填充，另一方向是用统计值来填充（平均值，众数，中值，特定值）**
3. **数据丢弃，如果出现缺失值的很少，按我们完全可以丢弃这小部分数据**

a把所有缺失值替换为指定的值0

data\_train = data\_train.fillna(0)

b)向用缺失值上面的值替换缺失值

data\_train = data\_train.fillna(axis=0,method='ffill')

c)纵向用缺失值下面的值替换缺失值,且设置最多只填充两个连续的缺失值

data\_train = data\_train.fillna(axis=0,method='bfill',limit=2)

**3)异常值处理**

当你发现异常值后，一定要先分清是什么原因导致的异常值，然后再考虑如何处理。首先，如果这一异常值并不代表一种规律性的，而是极其偶然的现象，或者说你并不想研究这种偶然的现象，这时可以将其删除。其次，如果异常值存在且代表了一种真实存在的现象，那就不能随便删除。在现有的欺诈场景中很多时候欺诈数据本身相对于正常数据勒说就是异常的，我们要把这些异常点纳入，重新拟合模型，研究其规律。能用监督的用监督模型，不能用的还可以考虑用异常检测的算法来做。

**检测异常的方法一：均方差**

在统计学中，如果一个数据分布近似正态，那么大约 68% 的数据值会在均值的一个标准差范围内，大约 95% 会在两个标准差范围内，大约 99.7% 会在三个标准差范围内。

def find\_outliers\_by\_3segama(data,fea):

data\_std = np.std(data[fea])

data\_mean = np.mean(data[fea])

outliers\_cut\_off = data\_std \* 3

lower\_rule = data\_mean - outliers\_cut\_off

upper\_rule = data\_mean + outliers\_cut\_off

data[fea+'\_outliers'] = data[fea].apply(lambda x:str('异常值') if x > upper\_rule or x < lower\_rule else '正常值')

return data

data\_train = data\_train.copy()

for fea in numerical\_fea:

data\_train = find\_outliers\_by\_3segama(data\_train,fea)

print(data\_train[fea+'\_outliers'].value\_counts())

print(data\_train.groupby(fea+'\_outliers')['isDefault'].sum())

print('\*'\*10)