2019“雅典娜杯”数据挖掘大赛

Team Smash 解题思路

1. 介绍

队员：Travis, None, Smash。

2. 赛题背景

本次比赛题目为贷款风险预测，基于用户基本信息、收入记录、支出记录、借贷信息、信用卡还款记录、浏览产品记录、产品基本信息等，解决用户的征信预测问题，预测用户是否逾期还款。

数据包含有用户银行流水记录、用户行为表、用户账单表、用户信息和标签五个部分。

3. 赛题理解

基于对赛题的分析以及金融风控业务的理解，本团队有以下理解：

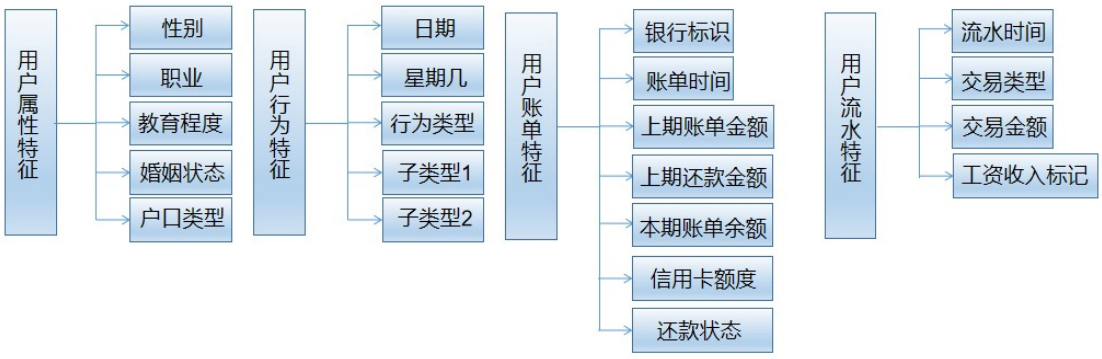
1、赛题重要性：准确地判断出用户未来是否有成为逾期还款用户的趋势，能够帮助金融企业更好地把控运营节奏，进行良性的资金配置和管理，最大化利用现有资金进行发展。

2、赛题目标：判断用户未来是否会逾期还款。

基于以上分析，本题应从用户自身出发，通过分析其个人信息、历史行为、账单、银行流水，寻找正常还款用户与准逾期违约用户的区别所在。

4. 特征工程

用户原始特征如下：



①用户属性特征工程：

用户的属性是构造用户画像的基本特征，尝试对基本属性进行组合后再编码，如“性别_职业”、“性别_教育程度_婚姻状态”等组合后，进行 LabelEncode 后作为新特征送入模

型；对属性特征进行 FrequencyEncode，即根据每一类的数量进行重新编码，以获取该类在模型中不同的表现形式。

```
for col in ['婚姻状态', '性别', '户口类型', '教育程度', '职业']:
    temp_df = data[[col]]
    fq_encode = temp_df[col].value_counts(dropna=False).to_dict()
    data[col+'_fq_enc'] = data[col].map(fq_encode)
```

②用户行为特征工程：

对行为特征做特征的基本想法是对用户的行为类型、子类型 1 和子类型 2 进行展开。本题的时间特征非 unix 系统时间，且每个月都有 31 号，猜想为公司系统自带时间，尝试对其根据每月 31 天这个规则并设定初始时间进行重编码。

- 对行为类型、子类型 1 和子类型 2 根据用户标识展开并计数，获取用户每个行为的累计特征；
- 对子类型 1、子类型 2 根据用户标识展开并进行唯一数统计(nunique)，通过统计某些没有进行过的行为加以区分不同用户；
- 根据用户和月份对行为进行计数，获取用户每月的行为计数特征；对子类型 1、子类型 2 根据用户标识和月份展开并进行唯一数统计。

③用户流水特征工程：

本表合并到主表后数据缺失较多，只考虑常规地做统计特征进行相应特征构建。

- 计算出流水时间间隔，对流水时间间隔做 max, min, mean, std 统计；

```
def time_diff_stat(ser):
    ts = ser.values // (3600 * 24)
    ts_diff = np.diff(ts)
    if ts_diff.shape[0] < 2:
        return 0, 0, 0, 0
    else:
        return ts_diff.min(), ts_diff.max(), ts_diff.mean(), ts_diff.std()

temp.sort_values(by=['用户标识', '流水时间'], inplace=True)
creditbill_stat_g = temp.groupby(['用户标识'])['流水时间'].apply(time_diff_stat).reset_index()
for i, st in enumerate(['min', 'max', 'mean', 'std']):
    creditbill_stat_g['流水时间间隔_' + st] = creditbill_stat_g['流水时间'].apply(lambda x: x[i])
del creditbill_stat_g['流水时间']

data = data.merge(creditbill_stat_g, on='用户标识', how='left')
```

- 根据用户标识、交易类型对交易金额做 max, min, mean, std 统计；
- 根据用户对交易金额做 max, min, mean, std 统计；

- 用户最近一个月的交易金额 max,mean;
- 根据用户和工资收入标记对交易金额进行统计,以获取用户农行卡的收入与支出的信息。

④用户账单特征工程：

账单特征为本题的提分关键所在，根据用户历史每月账单信息构建特征可以更好地区分出潜在逾期还款用户。

- 根据用户标识对上期账单金额、上期还款金额、本期账单余额和信用卡额度进行 max,min,mean,std 统计，并计算(上期还款金额、上期账单金额)和('信用卡额度','本期账单余额')统计值的差值；
- 根据用户标识和银行标识对用户的还款状态进行计数统计，获取用户还款次数的信息；
- 统计上期还款金额为零且上期账单金额为零的次数；统计上期账单金额为零的次数；统计信用卡额度为零且本期账单余额为零的次数；
- 根据用户标识对(上期账单金额-上期还款金额),(本期账单余额-信用卡额度)进行 max,min,mean,std 统计；对上期账单金额==上期还款金额和上期还款金额>上期账单金额进行计数统计；
- 根据用户标识和账单时间月份对本期账单余额展开后进行 std 统计；
- 考虑账单时间的发放时间，对账单时间是否小于凌晨 3 点和凌晨 6 点进行次数统计，竟然有点用；
- 根据用户标识对信用卡额度和银行标识分别做唯一值统计,企图得到用户各种银行卡的数量；
- 计算出账单时间戳间隔，对账单时间间隔做 max,min,mean,std 统计；
- 考虑到农行的风控模型应该随着时间推移越来越优秀,后来申请信用卡的用户逾期违约的概率理应更低，因此根据用户标识和银行标识对账单时间戳做 min 统计，求出每个用户对应每个银行标识的最初一条账单的记录时间。(这应该是本赛题的最强特征，最初直接根据用户标识对账单时间戳进行统计，会出现线下 KS 奇高而线上降很多分的现象，对实际业务分析后做出了修改，使得账单时间戳成为了强特之一)；
- 根据上一个特征，对强特继续做深入挖掘，继续根据用户标识和银行标识对时间戳做 max,std,mean,maxmin/2,std/mean 等特征，也能使得 KS 值得到一定的提升；

```
for tag in ['min','max','std','mean']:
    tmp = temp.groupby(['用户标识','银行标识'])['账单时间戳'].agg({tag}).reset_index()
    tmp = tmp.pivot(index='用户标识', columns='银行标识', values='账单时间戳').reset_index()
    tmp.columns = ['用户标识'] + ['用户标识_银行标识_账单时间戳_{}_{}'.format(i,tag) for i in range(13)]
    data = data.merge(tmp, on='用户标识', how='left')
```

• 考虑用户的还款习惯，对用户的历史还款做统计，即最近一个月/最近六个月的平均还款金额，也是一个不错的强特；

```
creditbill_month = temp.groupby(['用户标识', '银行标识', '月'])['上期还款金额'].sum().reset_index()
creditbill_habit = creditbill_month.groupby(['用户标识', '银行标识'])['上期还款金额'].apply(lambda x: x.values[-1] / x.tail(6).mean())
creditbill_habit = creditbill_habit.pivot('用户标识', '银行标识', '上期还款金额').reset_index()
creditbill_habit.columns = ['用户标识'] + ['银行标识%d_还款习惯1/6' % i for i in range(13)]
data = data.merge(creditbill_habit, on='用户标识', how='left')
```

⑤最后的再统计

在完成上述的所有特征构建后，再次对展开后的特征进行常规统计也能使得 KS 得到不少的提升，另外求其 gap 和变异系数也可以获取一定的提升。

• 对行为类型、子类型 1、子类型 2 的 count 特征进行 max,min,mean,std 统计；

• 对统计的特征求 gap，即 max-min 和变异系数即 std/mean；

```
data['行为类型_count_max'] = data[['行为类型{}_count'.format(i) for i in range(8)]].max(axis=1)
data['行为类型_count_min'] = data[['行为类型{}_count'.format(i) for i in range(8)]].min(axis=1)
data['行为类型_count_std'] = data[['行为类型{}_count'.format(i) for i in range(8)]].std(axis=1)
data['行为类型_count_mean'] = data[['行为类型{}_count'.format(i) for i in range(8)]].mean(axis=1)

data['交易金额_gap'] = data['交易金额_max'] - data['交易金额_min']
data['交易金额_变异系数'] = data['交易金额_std'] / data['交易金额_mean']
```

5. 模型设计

模型主要分为两部分，为两组不同的特征所构建，所用模型为 LightGBM，验证集构建方式为五折单种子交叉验证，并无进行 stacking，最终只进行简单的平均融合。