全国社会保险大数据应用创新大赛总结回顾

汪泽南

目录

[全国社会保险大数据应用创新大赛总结回顾 1](#_Toc494034328)

[1. 赛题分析 2](#_Toc494034329)

[2. 数据预处理 2](#_Toc494034330)

[2.1 缺失值处理 2](#_Toc494034331)

[2.2 降维 3](#_Toc494034332)

[2.3 原始特征转换 4](#_Toc494034333)

[3. 特征工程 5](#_Toc494034334)

[3.1 特征选择 5](#_Toc494034335)

[4. 模型设计与分析 6](#_Toc494034336)

[4.1模型选择 6](#_Toc494034337)

[4.2 数据划分 6](#_Toc494034338)

[4.3 调试模型参数 7](#_Toc494034339)

[4.4 预测 8](#_Toc494034340)

[5 附件 9](#_Toc494034341)

[附件1： 9](#_Toc494034342)

[附件2： 10](#_Toc494034343)

[附件3： 11](#_Toc494034344)

# 赛题分析

“选手需要基于给定的训练集数据得到模型，然后使用模型判定测试集中的人员是否为涉嫌造假人员”,从上面这对话中可以看出，这个赛题的是一个二分类问题，分别为“是涉嫌造假人员”和“不是涉嫌造假人员”。

题目中还给出了两个涉嫌造假人员的违规行为，如下：

（1）为了获得不当利益，部分人员从各种途径收集医疗保险参保人员的社保卡，通过社保卡到医院进行虚假诊疗，套取医保基金。

（2）在门诊特殊疾病的诊疗中，部分人员通过编造病历、诊疗过程，套取医保基金。

虽然还有很多别的形式的违规造假行为，但是这两个例子很好给我们这些几乎对造假现象没有概念的人一些提示。例如编造病历，那其中编造的病，应该是常见，高发的。这样才有利于造假人员进行多次造假而不容易被发现，这对我们后面的特征工程是有很好的指导意义的。当然还有很多的细节可以说，这里就不细细阐述了。

“以F1值作为最终的唯一评测标准”，可知，之后的评分需要采用F1值来做参考。

“训练集与测试集补充数据，涵盖人员在不同医保地址的消费金额明细与消费内容”，

上面这段是对fee\_detail表格的描述，是对训练集的一个补充。里面包含了消费内容，个人觉得这个对编造病历有一定的相关性，所以这个fee\_detail表格需要好好的加以利用。

# 数据预处理

## 2.1 缺失值处理

预处理中我主要对缺失值做了处理。缺失值会对我后面的特征工程带来比较大的干扰。所以必须处理。

首先我输出了所有样本的缺失值数量并用散点图进行了显示，我这样做的目的是想删除一些缺失值异常多的样本，但是并没有发现异常多的样本，反而发现几乎所有样本的缺失值是比较多的。所以放弃了删除异常样本的想法。

接下来分析一下缺失值处理的各种方式：

1. 删除。最简单最直接的方法，很多时候也是最有效的方法，这种做法的缺点是可能会导致信息丢失。
2. 补全。用规则或模型将缺失数据补全，这种做法的缺点是可能会引入噪声

* 用平均值、中值、分位数、众数、随机值等替代。效果一般，因为等于人为增加了噪声
* 用其他变量做预测模型来算出缺失变量。效果比方法1略好。有一个根本缺陷，如果其他变量和缺失变量无关，则预测的结果无意义。如果预测结果相当准确，则又说明这个变量是没必要加入建模的。一般情况下，介于两者之间。
* 最精确的做法，把变量映射到高维空间。这样做的好处是完整保留了原始数据的全部信息、不用考虑缺失值、不用考虑线性不可分之类的问题。缺点是计算量大大提升。而且只有在样本量非常大的时候效果才好，否则会因为过于稀疏，效果很差。

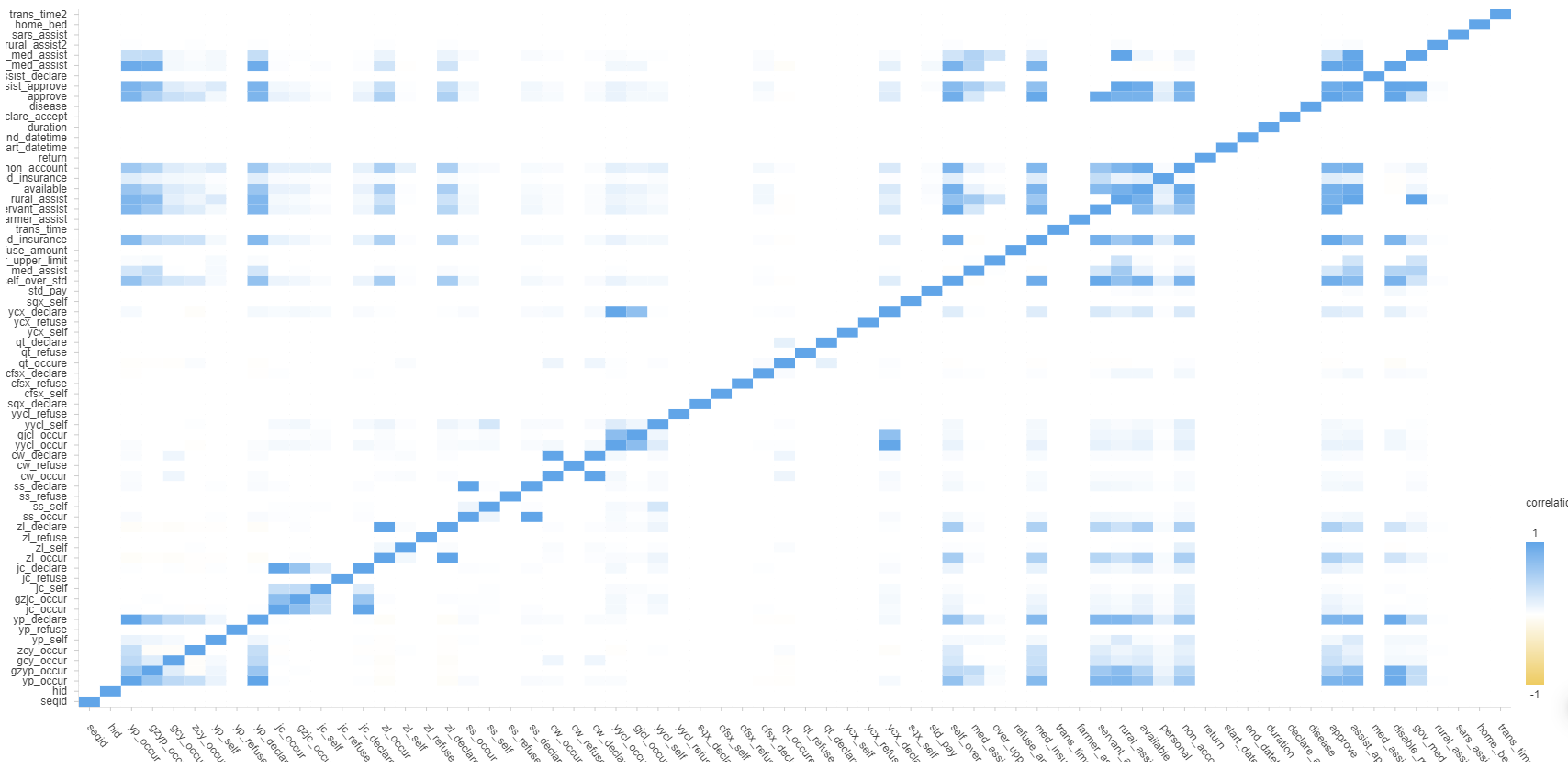
1. 忽略。有一些模型，如随机森林，自身能够处理数据缺失的情况，在这种情况下不需要对缺失数据做任何的处理，这种做法的缺点是在模型的选择上有局限。

想尝试上面的方案，但是，在观察所有的特征时，发现都是统计每一个样本的金额啊和消费项目等等的，那样本里面的缺失值可能是代表，该样本并没有使用该消费而没有记录到系统中，导出时，便是以缺失值的形式导出的。如此一来，那缺失值的意思和0是几乎一样的概念了，所以我最后采用的是以0来填充了所有的缺失值。

## 2.2 降维

观察数据内容可知，每一个人员根据时间不同有多个样本，为了预测每个人员是否为涉嫌造假人员，我们需要将每个人员的多个样本转化为统计量，而每一列特征都可以提取多个统计量，如此一来，最后生成的特征数量是庞大，这样不利于后期的特征工程和模型训练。

于是决定先对原始数据进行一次相关性分析，去除一些相关性非常高的特征，这样可以大大降低之后生成的特征维度。

表1 原始数据相关性表格：

首先我选定了几个目标特征，这几个目标特征的选择要求就是所在行对应的P值变换要尽可能的多。然后对每一个目标特征，进行了如下操作：

1. 去掉了与目标特征弱相关的特征：在上表中找到目标特征与任意非目标特征之间的相关度，如果相关度小于阈值(此阈值我没有采用比较完善的策略来进行选择，只是凭对上表的观察估计了一个阈值，0.05)则予以删除。
2. 去掉非目标特征之间强相关性的特征：在上表中找出相关性大于阈值(此阈值也是估计的，0.9)的每一对特征，然后再比较与目标特征相关度，去除相关度比较小的那一个。

最后，如果某一个特征在每一个目标值中至少一半以上被选中的话，将予以去除。

结果丢失，待补。

## 2.3 原始特征转换

对于trans\_time(交易时间)，我们用sql的to\_date转化为标准时间格式。随后统计出不同的时间，来计算出消费次数。统计出不同的天数，来计算出就诊天数。统计出最初和最终的时间，来计算出就诊时间跨度。

对于费用，大体分为两块，自费金额和报销金额(包括申报的金额)。对自费金额，统计了每个人费用总和，每个人的每天费用的均值，标准差，偏度和最大值。对于报销金额，统计了每个人费用总和，每个人的每天费用的均值，标准差和偏度。

对于fee\_detail这个表格，其中我觉得比较有用的就是sm\_stat(三目统计项目)，里面每一个数字代表一种类型的医疗项目。这应该是一个比较强特征。涉嫌造假人员在编撰病历时，一定是选有利于他们编造的特定的几个医疗项目的。那在这个包涵医疗项目的特征必然是很有用的。但是每个人员在不同天里的三目统计项目是有不相同的，也就是每个人会拥有多种三目统计项目。由于数字代表的是类别消息，所以不能用简单统计方法来处理。最后打算统计每一个数字类别的数量，选择出排名前5位的数字，分别为1,2,3,6,7。然后将三目统计项目用1,2,3,6,7这5列来代替，每一列的内容为每一个人拥有相应数字的数量。就尽可能保留了有用的数据，也让特征的维度不至于太大。

# 3. 特征工程

## 3.1 特征选择

特征的选择我采用过滤式(filter)。由于我后面会采用两个机器学习模型(分别是随机森林和xgboost)，并且这两个模型都能得出特征重要性。所以我打算采用这两个模型得出的特征重要性来进行特征选择。由于平台已经收回授权，无法运行，所以不能展示结果。

大体过程如下:



图1 特征选择大致步骤

图1 最后一步的详细步骤：

1. 根据特征重要性均值进行降序排序。
2. 确定要剔除的比例(更具我的特征数量，我选择了1:20的比例)，依据特征重要性剔除相应比例的特征，得到一个新的特征集。
3. 用新的特征集重复上述过程，直到剩下m个特征(m我设定为100)。
4. 最后根据前面几次的重复，选出两个模型平均最优的那一次。
5. 建立在自己对数据的观察上，对感觉可以删除或者可以保留的特征，进行保留或者删除，如果能提高预测效果的，说明前面采取的措施是正确的，否则撤销。

## 4. 模型设计与分析

## 4.1模型选择

由于是标准的二分类问题，所以我们从分类算法中选择。由于提供的特征数量较大，类别也较多，所以选择了对特征要求不多的树模型。之前打算采用xgboost和随机森林双模型融合的方式。但是后来发现提升并不明显，反而工作量大大增加了，所以最后决定只采用xgboost模型。一开始使用的随机森林在PAI平台上的构建图可以见附件1(调试参数)，附件2(预测测试集)。

## 4.2 数据划分

通过对数据的观察，发现这是不平衡的数据。如果将所有样本一视同仁，不做任何处理，会造成了分类器在多数类的分类精度较高而在少数类的分类精度很低。

解决方法主要分为两个方面，第一种方案主要从数据的角度出发，主要方法为抽样，既然我们的样本是不平衡的，那么可以通过某种策略进行抽样，从而让我们的数据相对均衡一些；第二种方案从算法的角度出发，考虑不同误分类情况代价的差异性对算法进行优化，使得我们的算法在不平衡数据下也能有较好的效果。

我选择的是第一种方案。

采样算法通过某一种策略改变样本的类别分布，以达到将不平衡分布的样本转化为相对平衡分布的样本的目的，而随机采样是采样算法中最简单也最直观易懂的一种方法。随机采样主要分为两种类型，分别为随机欠采样和随机过采样两种。

随机采样通过改变多数类或少数类样本比例以达到修改样本分布的目的，从而让样本分布较为均衡，但是他们也存在一些问题。对于随机欠采样，由于采样的样本要少于原样本集合，因此会造成一些信息缺失，未被采样的样本往往带有很重要的信息。对于随机过采样，由于需要对少数类样本进行复制因此扩大了数据集，造成模型训练复杂度加大，另一方面也容易造成模型的过拟合问题。针对这些问题提出了几种其它的采样算法。

最终我选择随机欠采样，并做了一些改进。

在对多数样本采样时，我采取有放回的采样，这样最后的采样结果之间会有部分重复的可能，这样可以避免每个样本之间完全独立。在对少数样本进行采样时，我没有采用全部数据，也是采用有放回的采样，这样可以一定程度的抑制对少数样本的过拟合问题。

## 4.3 调试模型参数

调试模型参数的大致步骤如下：



图2 训练模型的步骤

由于PAI平台上并没有xgboost，所以在IDE平台中用PAI指令来完成模型的训练和预测，可见附件3。而且由于ODPS平台的局限，实现交叉验证需要大量的步骤，所以最终放弃交叉验证，采用手动修改参数，然后比较平均F1成绩，来选择最优的参数。

## 4.4 预测

对测试集的预测步骤如下：

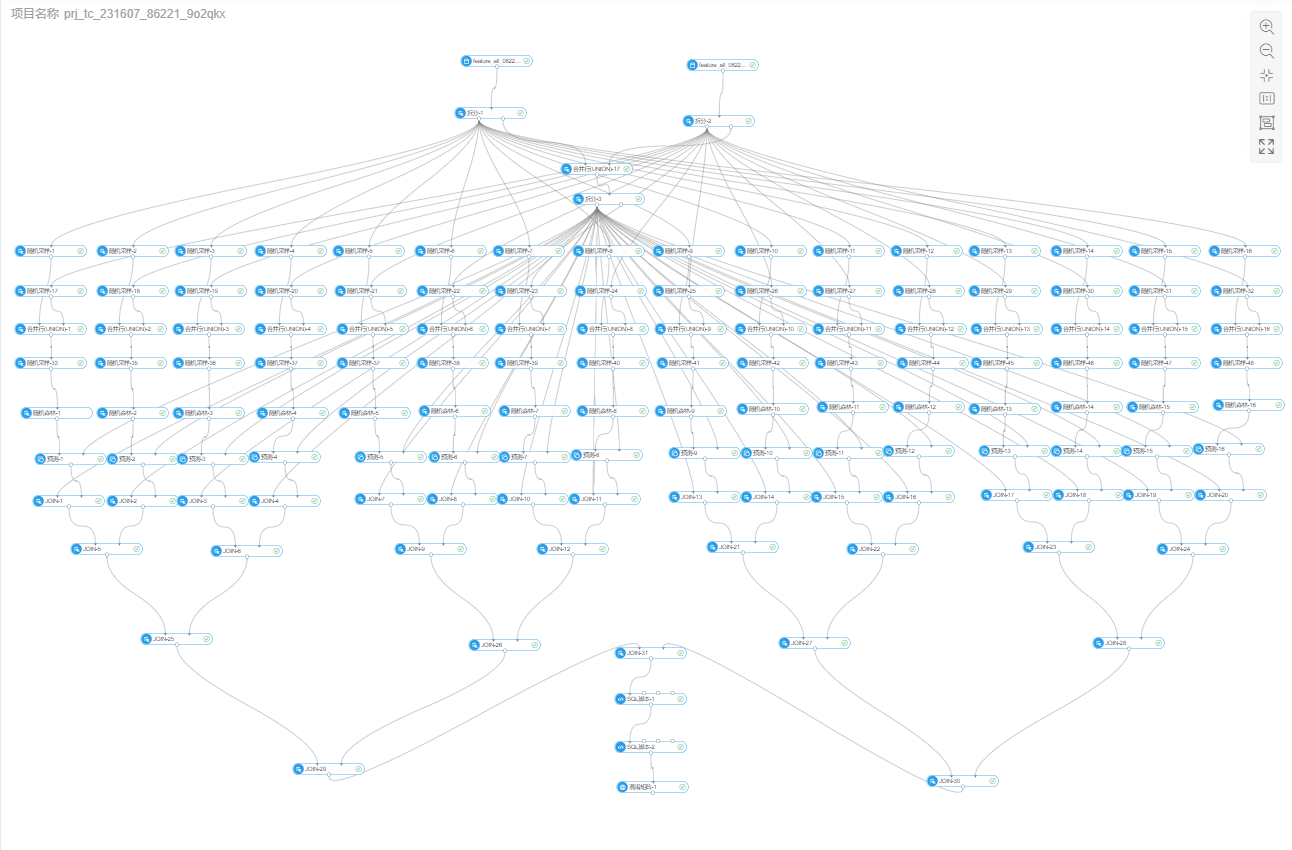


图3 预测步骤图

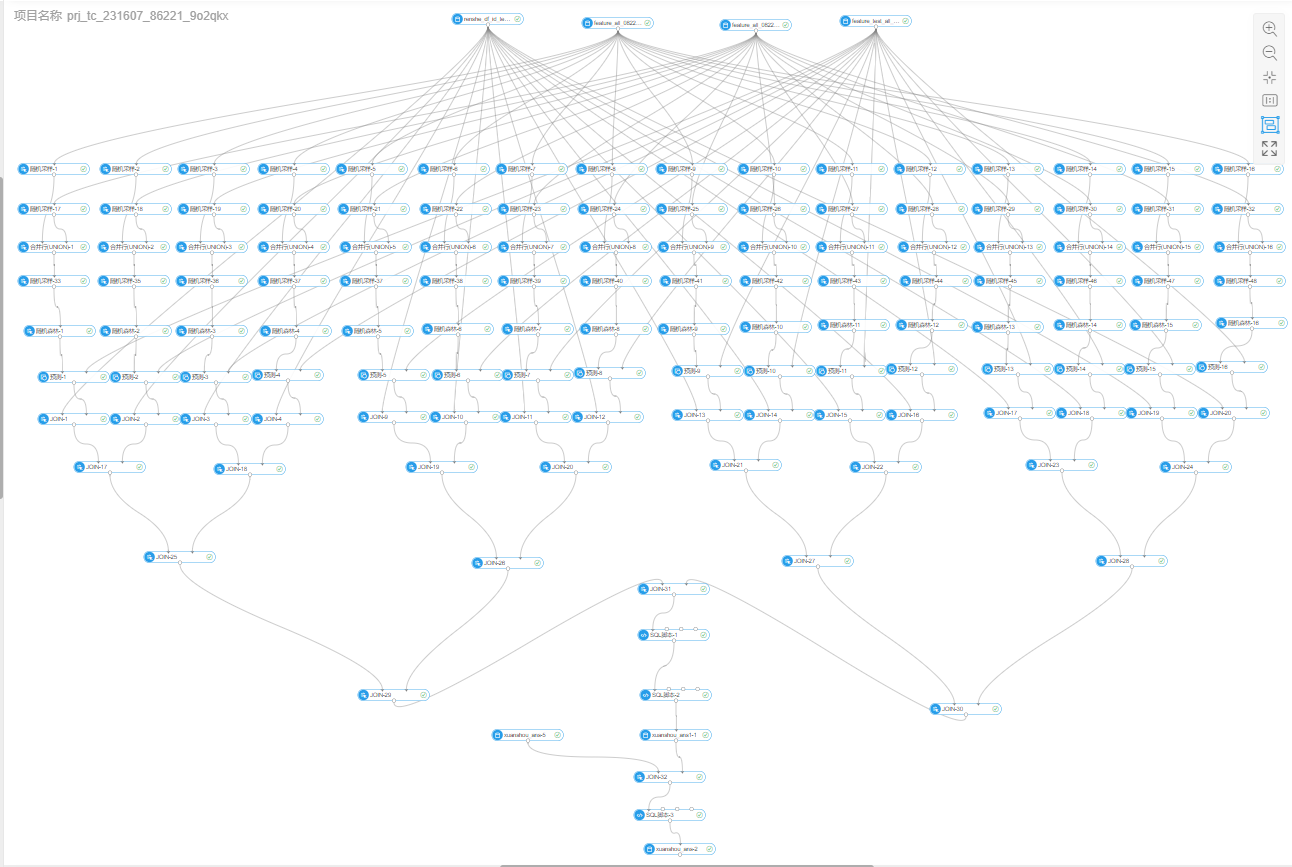
基本步骤和训练时，基本一致，不同之处就是在分出16分训练集时，采用的是全部的数据集。

# 5 附件

## 附件1：



## 附件2：



## 附件3：

|  |  |
| --- | --- |
|  | DROP OFFLINEMODEL IF EXISTS xgboost\_1; |
|  |
|  | -- train |
|  | PAI |
|  | -name xgboost |
|  | -project algo\_public |
|  | -Deta="0.01" |
|  | -Dobjective="binary:logistic" |
|  | -DitemDelimiter="," |
|  | -Dseed="33" |
|  | -Dnum\_round="5000" |
|  | -DlabelColName="label" |
|  | -DinputTableName="feature\_1" |
|  | -DenableSparse="false" |
|  | -Dmax\_depth="6" |
|  | -Dsubsample="0.8" |
|  | -Dcolsample\_bytree="0.8" |
|  | -DmodelName="xgboost\_1" |
|  | -Dgamma="0" |
|  | -Dlambda="1"  -DfeatureColNames="省略" |
|  |
|  | -Dbase\_score="0.25" |
|  | -Dmin\_child\_weight="20" |
|  | -DkvDelimiter=":"; |
|  |  |
|  |
|  | -- predict |
|  | PAI |
|  | -name prediction |
|  | -project algo\_public |
|  | -DdetailColName="prediction\_detail" |
|  | -DappendColNames="pid" |
|  | -DmodelName="xgboost\_1" |
|  | -DitemDelimiter="," |
|  | -DresultColName="prediction\_result1" |
|  | -Dlifecycle="28" |
|  | -DoutputTableName="prediction\_1" |
|  | -DscoreColName="prediction\_score" |
|  | -DkvDelimiter=":"  -DfeatureColNames="省略" |
|  | -DinputTableName="xgb\_test" |
|  | -DenableSparse="false"; |