火星人

队伍成员：朱秋辉（四川大学）、郭柯娜（四川大学）、黄志标（中国科学院）

算法基本思路：

1、数据分析上，1138 个特征，其中 93 个 category、1045 个 numeric 型特征。样本中同时含有缺失值-1 和-2（极少）。

（1）由于分类器往往默认数据是连续的，并且是有序的。而对于 category 型特征，我们有两种方法来处理，第一种方法是人工地指定为一个顺序，但这样做工作量太大，不适合做。第二种方法就是进行 OneHotEncoding，对一些 category 型特征进行编码。于是我们选择特征取值大于 2 个取值（不含缺失值）的特征进行 OneHotEncoding, 大致选择了以下特征：

x417	1	2
x1134	7	8
x415	1	4
x416	1	4
x411	1	7
x1132	1	7
x1119	1	8
x1120	1	8
x1121	1	8
x1122	1	8
x1123	1	8
x1124	1	8
x1125	1	8
x1126	1	8
x1127	1	8
x1128	1	8
x1129	1	8
x1130	1	8
x1133	1	8
x1136	1	8
x1137	1	8
x1111	1	27
x1109	1	33
x1110	1	33
x1112	1	33
x1113	1	33
x1138	1	33
x1107	1	34
x1118	1	34
x1117	1	35
x1131	1	35
x1108	1	36

x1114	1	36
x1115	1	36
x1116	1	36
x1135	1	36

上述特征，第一列是特征名字，第二列是特征最小值，第三列是特征最大值。

（2）对缺失值的分析。如下图，对于 test_x.csv 进行每个样本的缺失值进行部分截图：

uid	test缺失值
1574	12
12148	14
12828	17
15881	18
16862	18
11518	19
8977	20
19950	20
143	21
1142	21
18041	21
19029	21
19213	21
3451	22
6560	22
8068	23
8248	23
10866	23
13462	23
1961	24
5785	24
7007	24
10699	24
12851	24
17957	24

可见，样本是以缺失值的多少为聚簇的，比如在测试集中样本缺失值为 24 的样本有 5 个，我们可以归为一簇。于是我们根据这个想法，新加入了两个特征，即每个样本的缺失值个数以及非缺失值个数。结果表明，这两个特征在线上的效果是很明显的。

（3）缺失值样本的发现。test_x.csv 中样本缺失值统计部分图如下：

uid	test缺失值个数
5992	1050
10083	1050
19541	1050
10127	583
16422	318

Uid 为 5992，10083，19541 的样本缺失大量的特征，于是我们把此类样本单独抽出来进行辨别。结合训练数据同样缺失 1050 的样本统计分析出有 19 个，uid 不再一一列出。我们把训练集中 19 个样本看为一个训练集（同时去掉了在特征上取相同值的列，去除后共计 47 个有用的特征），把测试集中 3 个样本看成测试集，运用随机森

林（适合多维数据的处理）来建模，得到的结果如下：

uid	5992	10083	19541
score	0.93475882	0.75566694	0.90268578

上面的结果表明：uid 为 10083 的样本与其他两个样本差异性很大，于是我们可以单独预测出 uid 为 5992，10083，19541 的类标分别为 1,0,1。此思路可以扩展到其他类样本上，但由于工作量比较大，只完成了最大缺失值样本的识别。

2、模型选择。在模型选择上我们根据数据集本身的特征，试采用了逻辑回归（特征选择，采用逐一搜索法），随机森林，GBDT(采用的 xgboost),结果表明逻辑回归表现了 66%，随机森林表现最高达到 66.8%，而 xgboost 表现最高 72.29%的识别率。

以上结果值得说明的是，在使用 xgboost 要注意的有几点：

（1） 随机种子的选择。多个种子的选择可以稳定分类器的性能，在当种子选择为 1220 时我们的分类效果最高。

（2） 参数的设置问题。其中正样本的权重设置问题十分重要，经过参数的设置，我们以 80%数据做为训练集，20%数据作为测试集，如下表：

正样本权重	0.11	0.12	0.13	0.14	0.15
score	0.71994	0.72069	0.72105	0.72084	0.72006

当正样本权重取 0.13 时候达到最优，同时我们设置构造 8000 棵树。以下是我们参数的选择：

```
params = {
    'booster': 'gbtree',
    'objective': 'binary:logistic',
    'early_stopping_rounds': 100,
    'scale_pos_weight': 0.13,   # 正样本权重
    'eval_metric': 'auc',
    'gamma': 0.1,
    'max_depth': 8,
    'lambda': 550,
    'subsample': 0.7,
    'colsample_bytree': 0.4,
    'min_child_weight': 3,
    'eta': 0.02,
    'seed': 1220,
    'nthread': 4
}
```

（3） 缺失值的指定参数。相信很多人都会忽略的一点，在 xgboost 说明中，我们在加载数据的时候我们可以指定缺失值，如代码：

```
dtrain = xgb.DMatrix(X, label=y, missing=-1)
```

除了单个模型外，我们试着考虑模型间的组合，如 bleeding。但由于很难找到与 xgboost 能达到同等效果的相异分类器，我们试着把随机森林与 GBDT 进行相组合，但由于随机森林本身效果不高，预测的结果与单个 GBDT 的效果相差不大，组合分类器工作就此放弃。