三态笔试题报告

数据集分析

1. 典型的不平衡二分类问题
2. 变量较少（只有 8 个），且基本全是相关变量
3. 数据集完整程度比较好，没有缺失值，也没发现异常填充值

分析策略：

1. 由于数据集没有缺失值，且较完整，所以给特征工程节省了较多时间。关于特征工程中的特征选择，根据业务分析，基本全是相关的特征。**而且特征的筛选最好在种类变量的重新热编码之前做，不然会导致模型可解释性降低**。热编码是为了给种类变量重新编码而生成一个稀疏向量，往往这么做是为了满足模型的输入要求，而且如果不做么做，把种类变量当成连续变量处理，很容易在训练过程中导致模型过拟合。这里说的为何要在热编码之前做特征选择，是因为热编码之后往往之后会产生稀疏数据，导致数据维度增加，此时我们如果做一个降维处理，会大几率导致某些生成的变量被 drop，即使我们最后得到了一个高精度模型，此时我们预测一个新的样本，它的其中一个属性恰好是被 drop 掉的一个变量，所以模型无法预测。举一个简单的例子，一个地区种类变量被重新热编码之后生成 50 个新变量代表 50 个不同的地区，由于某些地区的样本很少，降维之后这些地区的变量被处理掉了，然后我们训练了一个模型，此时有一个新样本恰好来自那些被忽略掉的地区，所以样本就不符合模型的输入要求而导致做不了预测（可解释性差）。

2. 典型的不平衡二分类问题，我有 3 步策略：
    1. 改变评价指标：着重看 F1 或者 ROC_AUC
    2. 算法层面上惩罚那些样本量多的种类，ie sklearn 算法中把 class_weight 设置成 balanced，反比惩罚。此时利用 AutoML 训练一遍所有模型选择出最优模型。
    3. 数据层面上用过采样或者欠采样算法（SMOTE 过采样），再利用 AutoML 训练一遍所有没模型，选择出最优模型。
    4. 利用 2 和 3 中得出的结果交叉验证得出最优模型。根据我的经验，往往 2 中表现好的模型依然在 3 中表现也好。
    5. 在这个 case 中，逻辑回归表现最后，2 和 3 的结果都是最优的。

总结

1. 参与选择的模型：逻辑回归/随机森林/梯度提升树（GBDT）/XGBOOST
2. 最后选择的最优模型：逻辑回归
3. 整个过程的随机种子已经固定，结果可以复现：random_state = 2020

由于时间问题，没有用更复杂的模型或者是模型融合上的策略。并且我发现基于梯度提升框架的树模型在这个 case 上表现不好，比如 GBDT 和 XGboost，只有在应用数据采样算法之后效果才好起来，但是结果依然不如解释性很强的逻辑回归。

整个训练过程对于参数空间不大的算法，用到了 GridSearchCV 超参数优化，对于参数空间过大的算法用到了 RandomizedSearchCV 超参数优化。

利用我的框架，每 50 行就可以加个新算法从而系统比较每个算法。每个算法的效果都记录在了文件中，这个题目已经放到了我的 github 中：https://github.com/DimonYin/Predict-the-purchase-probability-of-a-room