

人工智能（研讨）大作业报告

数据清洗及预处理

传统机器学习的数据预处理

对给定的数据集，去除id列，将二元变量取值编码为01，对多类型变量进行独热编码，并对每一个数值变量标准化。

深度学习方法的数据预处理

- 读取与准备数据：从CSV文件加载训练和测试数据，并分离出特征和标签。
- 类别编码：使用LabelEncoder将字符串形式的标签转换为整数形式，以便用于训练模型。
- 处理 'CALC' 字段：确保所有可能的值（包括 'Always'）都被考虑在内，以保持训练集和测试集之间的一致性。如果某些值不在原始数据中，则通过临时添加这些值来进行独热编码，然后移除这些临时行。
- 独热编码：对分类变量进行独热编码（One-Hot Encoding），确保每个分类变量都转换成二进制向量。
- 标准化：使用StandardScaler对数值型特征进行标准化处理，使得它们具有零均值和单位方差。
- 数据集划分：使用train_test_split函数将训练数据划分为训练集和验证集，以便在训练过程中评估模型性能。创建 PyTorch 数据集和数据加载器：定义了自定义的ObesityDataset类来封装数据，并使用 DataLoader 创建了可以迭代的数据加载器，方便批量处理数据。

H2O与XGBoost的数据预处理

- 数据读取：我们首先使用 pandas 读取训练集 train.csv 和测试集 test.csv，并分别加载数据。由于id和 NObeyesdad不作为特征用于模型训练，因此它们被排除。
- 数据分割：为了提高模型性能并验证其泛化能力，使用 train_test_split 将训练数据按 9:1 的比例分为训练集和验证集。这样一方面可以保证足够的数据用于训练，另一方面又能用验证集来评估模型在真实数据上的表现。
- 特征处理：为了适应 H2O 自动化机器学习框架，我们将目标变量 NObeyesdad 转换为因子类型，这样有利于 H2O 中的分类模型处理。特征如 X_train 和 X_val 也同样将目标变量进行因子编码。
- 数据格式转换：最后将训练集和验证集转换为 H2O 的 H2OFrame 格式，以支持 H2O 自动化机器学习工具（如 H2O AutoML）。同时，测试集也进行相同转换。转换后，我们使用 H2O 提供的 asfactor 方法将目标列转换为因子，以确保模型能够正确处理分类目标。
- 对于XGBoost模型，我们合并 train_data 和 test_data 使得可以在同一个数据框中处理特征工程步骤，并使用 ignore_index=True 确保索引统一，从而避免因训练集和测试集索引差异导致的问题。然后将指定的类别变量进行独热编码，包括'Gender', 'family_history_with_overweight', 'FAVC', 'CAEC', 'SMOKE', 'SCC', 'CALC', 'MTRANS'并对目标变量 NObeyesdad使用LabelEncoder对其进行分类编码，将类标签转换为数值。使用 iloc 将合并后的数据分割回训练集和测试集，确保数据格式正确，将合并后的数据重新划分回原始训练集和测试集。选取 'Age', 'Height', 'Weight' 这三列作为数值特征，使用StandardScaler进行标准化，使数值特征的均值为0，标准差为1。由于在合并数据时可能会丢失一些特征，在测试集上补齐这些缺失特征。最后找出 X 和 test_data 中不同的列，并用 0 填充，确保测试集特征与训练集的特征一致。

预测具体方法描述

传统机器学习

方法一

从CSV文件读取数据集，分离特征X和目标变量y。使用StandardScaler对特征进行标准化。将数据集按8:2比例拆分为训练集和测试集。通过GridSearchCV在预定义的参数网格中选择最优参数。用测试集评估模型，输出分类准确率。对新数据进行分类，并保存预测结果至文件。

方法二

从文件processed_data.csv加载数据集，并分离出特征X和目标变量y。使用train_test_split将数据集按8:2比例拆分为训练集和测试集。创建KNN分类器，设置n_neighbors=5（即选择距离最近的5个邻居进行投票）。使用训练集（X_train和y_train）训练KNN分类器。对测试集（X_test）进行预测，并计算分类准确率。加载processed_test.csv中的新数据，使用训练好的模型进行预测，将预测结果添加为新列NObeyesdad，并将结果保存至predicted_results.csv。

方法三

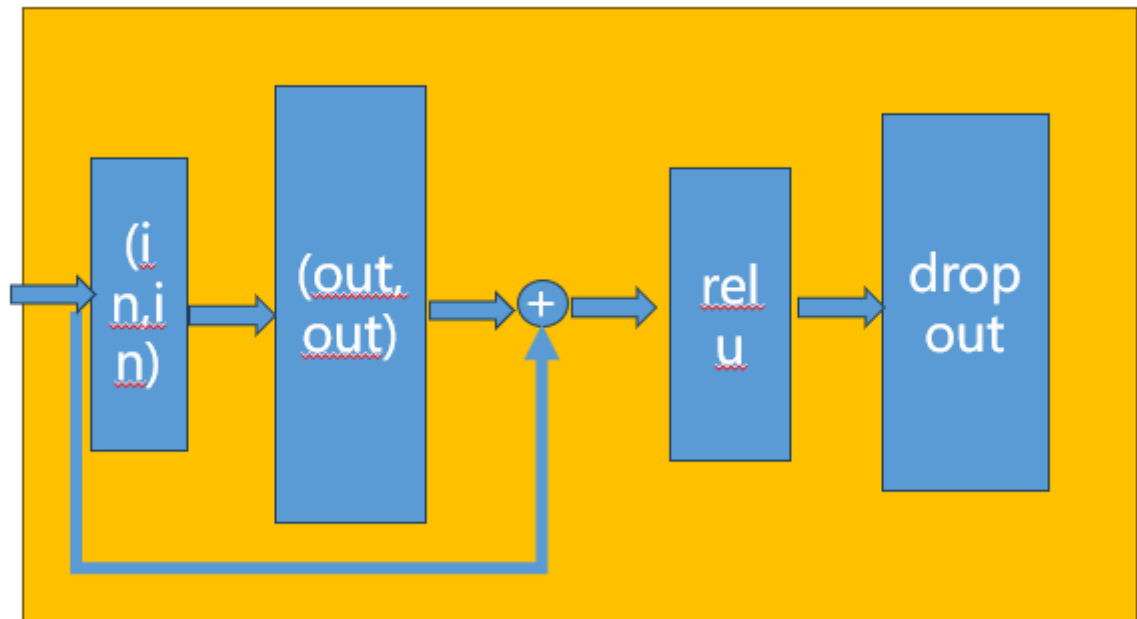
从文件processed_data.csv加载数据集，分离出特征X和目标变量y，并为特征添加一列CALC_Always默认值为False。使用train_test_split将数据集按8:2比例拆分为训练集和测试集。定义随机森林分类器并使用GridSearchCV对多个超参数进行调优，在训练集上寻找最佳参数组合。使用最佳模型对测试集进行预测，并计算分类准确率。同时提取模型的特征重要性，分析各特征的贡献。加载processed_test.csv中的新数据进行预测，将预测结果添加为新列NObeyesdad，并将结果保存至predicted_results.csv。

神经网络

自己搓的残差网络

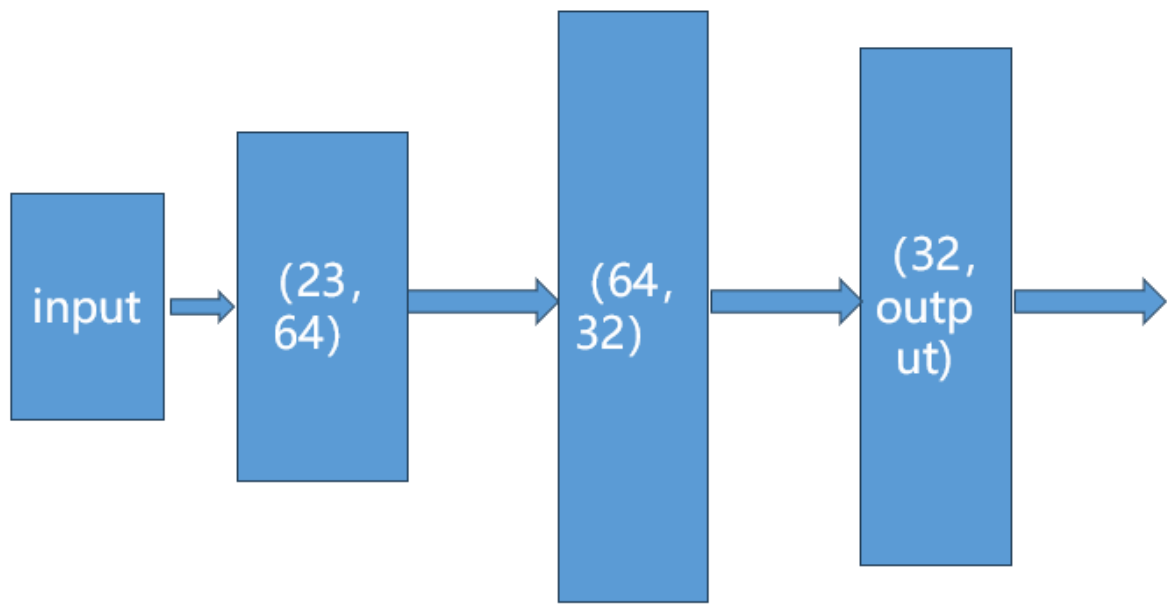
1. ResidualBlock (残差块):

- 包含两层全连接层，每层后面跟有ReLU激活函数和Dropout层，以防止过拟合。
- 如果输入和输出维度不同，则通过线性变换调整维度大小。
- 最后加上输入作为残差，实现跳跃连接（skip connection），有助于缓解深层网络中的梯度消失问题。



2. **ObesityClassifier (肥胖症分类器):**

- 由两个连续的残差块组成，逐渐减少神经元数量（64 -> 32），最后通过一个全连接层映射到输出维度（即类别数）。
- 每个残差块内部都包含了ReLU激活函数和Dropout层，以增强非线性和提高泛化能力。



3. **损失函数和优化器:**

- 使用交叉熵损失函数 (**CrossEntropyLoss**) 作为多分类任务的标准损失函数。
- 使用Adam优化器，并设置了初始学习率和L2正则化参数 (权重衰减)，以帮助模型更好地收敛。

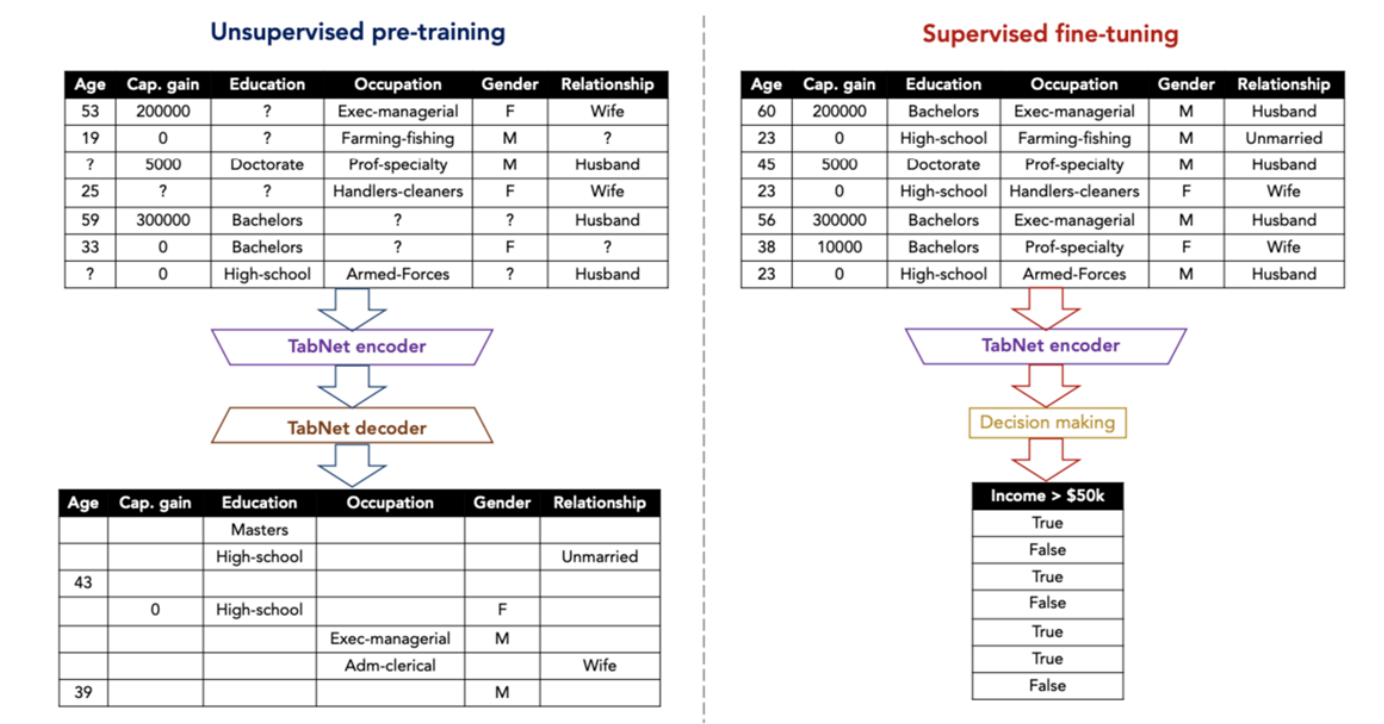
4. **模型训练:**

- 训练过程中，每隔一个epoch计算一次验证集上的准确率。
- 训练完成后，保存模型权重到指定路径，以便后续加载或部署。

5. **测试与预测:**

- 加载训练好的模型，在测试集上进行预测，并将预测结果解码回原始标签。
- 创建提交文件，包括测试样本ID和对应的预测类别，以便参加竞赛或提交结果。

TableNet



TabNet 是专为处理 表格数据（Tabular Data） 设计的深度学习模型，利用注意力机制动态选择特征，能够实现高效建模和可解释性。它将深度学习的优势引入表格数据，弥补了传统神经网络在这方面的劣势。

整体架构

- 编码器（Encoder）：包含特征变换模块（Feature Transformer）和注意力变换模块（Attentive Transformer）。
- 解码器（Decoder）：用于无监督学习和缺失数据的填充。
- 决策步骤（Decision Steps）：类似决策树的节点，动态选择特征并逐步累加特征权重。

主要组件

- Feature Transformer 对输入特征进行特征处理，包含共享模块（全局共享）和步骤依赖模块（仅作用于当前决策步骤）。
- Attentive Transformer 通过生成特征 Mask，完成特征选择，决定哪些特征对当前步骤更重要。

关键机制

- 特征选择：通过 Sparsemax 生成稀疏特征 Mask，限制每个特征在不同决策步骤中的使用。
- 累加权重：每个步骤的输出逐步累加，用于最终预测。

XGBoost

XGBoost 的 K 折交叉验证与网格调参过程旨在优化模型性能，通过选择最佳超参数来减少过拟合，提高模型的泛化能力。加载数据集后，首先合并训练集和测试集，以确保数据统一处理。利用 StratifiedKFold 进行 5 折交叉验证，确保在每次折叠中类别分布保持一致，有助于避免因数据集划分不均导致的偏差。接着，网格搜索用于调节 XGBoost 模型的超参数，帮助寻找最佳的参数组合：

- 针对 n_estimators（树的个数），网格搜索设定了一系列迭代次数，如 50、100、200、300 和 500，结合交叉验证和准确率指标，选择了模型的最优迭代次数。适当调整 n_estimators 可以提高模型的训练效率，同时减少过拟合和加快训练速度；

- 针对 max_depth（最大深度），设置了不同的取值，如 3、5、10、15、20，通过网格搜索和交叉验证来选择最佳深度。合理的最大深度可以有效控制模型复杂度，防止过度拟合，提升模型的泛化能力；
- 针对 min_child_weight（最小样本权重）和 gamma（最小丧失函数精度调整），网格搜索设定了多种取值。min_child_weight控制每个叶子节点最小样本数量，避免过于复杂的树结构；而gamma控制了进一步减少丧失函数值所需的最小样本数量。这一步有助于模型在减少复杂性的同时，保持良好的预测性能。
- 将选择出的最佳超参数组合应用到 XGBoost 模型中进行最终训练，使用训练集提高模型性能，并在验证集上进行性能评估，输出分类报告来检验模型的准确性。将最佳模型应用于测试集，生成最终预测结果，并将这些预测结果保存为提交文件，以供后续评估。这种方法可以有效地提升 XGBoost 模型的预测能力，同时减少调参和选择的时间，让整个机器学习过程更加高效和简单。

AutoDL

AutoDL通过智能算法自动选择和优化深度学习模型高效完成复杂的机器学习任务。首先，AutoDL 自动探索神经网络、卷积神经网络以及循环神经网络等模型类型，根据数据特性选择最合适的模型结构，从而提高模型的表现。它自动调节超参数，如学习率、层数和单元数等，确保模型在验证集上表现最优。数据集存在类别不平衡时，AutoDL通过调整样本权重或引入采样策略来平衡数据分布，提升模型预测的准确性并还会自动处理数据中的特征，比如类别变量的独热编码、数值特征的标准化的以及缺失值的填补，从而确保模型输入的特征尽可能精简且有用。后面在模型训练过程中会自动评估所有训练出的模型，并根据性能指标选择表现最优的模型，避免了用户手动选择模型的繁琐过程。最后使用选出的最佳模型进行测试集预测，并生成预测结果，保存为提交文件，方便后续使用。这些全自动化步骤，极大地节省了调参和模型选择的时间，让用户能够专注于更高层次的分析和决策。

实验结果分析

传统机器学习

从实验结果来看，随机森林算法表现最好，这可能是因为随机森林能够通过集成多个决策树来减少过拟合，并且对特征之间的复杂关系具有较强的建模能力。它在处理大规模数据集时通常表现出较好的鲁棒性和高准确度，尤其在特征之间的相关性较强时，能够有效利用数据中的信息。支持向量机（SVC）表现居中，原因可能在于它对于高维特征的处理较为高效，尤其适用于数据具有较为复杂边界的情况。然而，SVC通常对数据的尺度和核函数选择较为敏感，可能需要较长的训练时间和细致的超参数调整才能发挥出最优性能。K近邻

（KNN）算法的效果较差，主要是因为它是一个基于距离的算法，对数据的尺度非常敏感，并且在数据量大时计算复杂度较高。在特征空间较大或者数据分布不均匀时，KNN的表现会受到影响，容易导致较高的误分类率。

残差网络

通过残差连接，减缓了过拟合，最终在参数为：Lr=0.0001，weight_decay=0.0001，Epoch=1000下得到了最佳效果

TabNet

我们实际用下来发现，对于这个任务来说，表现并不如刚刚的残差链接，我们也分析了一下，这个场景的训练集只有两万条，对于这样一个模型来说还是不太合适，更加先进的机器学习模型应该是更好的选择

XGBoost

- 模型性能分析

通过对验证集的分类报告来看，模型的主要性能指标如准确率、召回率、F1 分数等都得到了提升。交叉验证和网格搜索的过程帮助我们选择了合适的超参数，使模型在验证集上的性能表现最佳。例如，在验证阶段，准确率为 0.90、F1 分数为 0.91，这表明模型在分类能力上得到了显著提升。

特定超参数选择能够有效控制模型复杂度，从而减少了过拟合的风险，提升了泛化能力。调整 min_child_weight 和 gamma 也增强了模型对样本的不平衡性处理能力，使得模型在实际应用中表现更加稳定。

Validation Report:				
	precision	recall	f1-score	support
0	0.95	0.92	0.93	255
1	0.88	0.88	0.88	330
2	0.89	0.89	0.89	260
3	0.98	0.98	0.98	337
4	1.00	1.00	1.00	404
5	0.78	0.81	0.79	244
6	0.82	0.82	0.82	246
accuracy			0.91	2076
macro avg	0.90	0.90	0.90	2076
weighted avg	0.91	0.91	0.91	2076

• 超参数选择效果

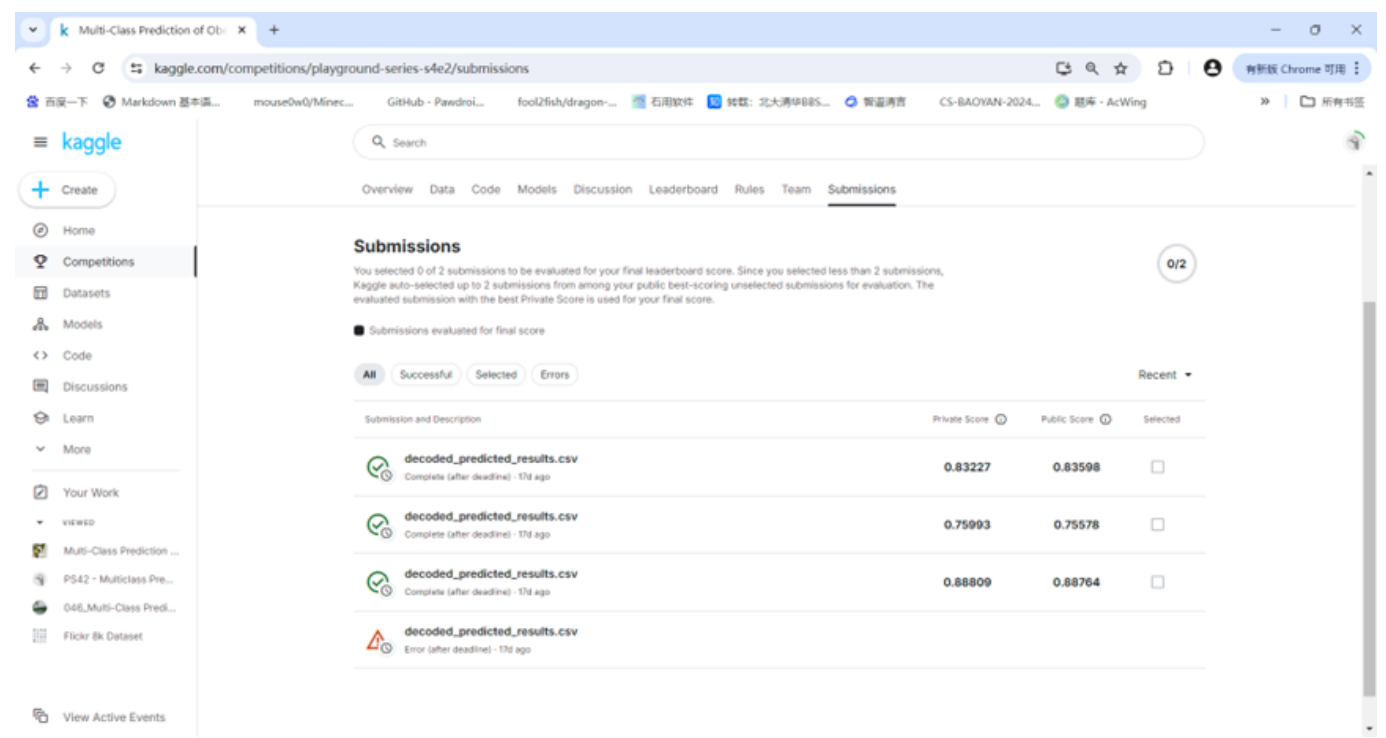
通过网格搜索和交叉验证，选择了最佳的超参数组合。对于 n_estimators，经过多轮测试，最终选择了 50 作为最佳值，平衡了模型的训练时间和效果。对于 max_depth，选择了 5，这一深度有助于控制过拟合，而对于 min_child_weight 和 gamma，分别选择了 1 和 0.1，这些参数帮助模型平衡样本的复杂度和丧失函数。超参数优化的效果显著提升了模型的稳定性和预测能力，减少了过拟合，同时提高了在验证集上的分类性能。

AutoDL




AutoDL（自动深度学习）通过自动化模型选择、超参数调节和特征处理，大幅度提升了机器学习任务的效率和模型性能。其自动化过程有效减少了人工操作的复杂性，使得模型能够在大数据集和高维特征空间下表现更为稳定和准确。在实验中，AutoDL 自动选择了最优模型结构并优化了超参数，如学习率、层数和单元数，从而显著提升了验证集和测试集上的表现。通过自动化特征处理，模型输入的特征质量得到了优化，进一步提高了模型的泛化能力。实验结果表明，AutoDL 在降低调参难度、提升模型准确性和泛化能力方面取得了显著效果，能够有效推动机器学习任务的高效实现。

系统给出的score证明

传统机器学习



残差连接





	submission.csv Complete (after deadline) · 18d ago	0.86994	0.86632	<input type="checkbox"/>
	submission.csv Complete (after deadline) · 18d ago	0.88050	0.88439	<input type="checkbox"/>
	submission.csv Complete (after deadline) · 18d ago	0.88069	0.88114	<input type="checkbox"/>

TableNet

0.87463	0.87210	<input type="checkbox"/>
0.87463	0.87210	<input type="checkbox"/>
0.87319	0.87716	<input type="checkbox"/>
0.86217	0.87174	<input type="checkbox"/>

XGBoost与H2o-automl

从上到下分别为： XGBoost结果； 三次迭代网格搜索调参后结果； K折交叉验证调参后结果； H2o-automl结果：

 submission.csv Complete (after deadline) · 41m ago · 未调参	0.90092	0.90968	<input type="checkbox"/>
 submission.csv Complete (after deadline) · 9m ago · 三次分别调参	0.90299	0.90859	<input type="checkbox"/>
 submission.csv Complete (after deadline) · 22s ago · k折交叉验证	0.90335	0.90895	<input type="checkbox"/>
 submission2.csv Complete (after deadline) · 17d ago	0.90606	0.91040	<input type="checkbox"/>

最终最高得分： private-0.90606 public-0.91040