

# 机器学习实验评分和分析教程-财务重点

 [h2oai.github.io/tutorials/machine-learning-experiment-scoring-and-analysis-tutorial-financial-focus](https://h2oai.github.io/tutorials/machine-learning-experiment-scoring-and-analysis-tutorial-financial-focus)

## 4.任务2：浏览实验设置和专家设置

1.将鼠标悬停在“**实验设置**”上，并记下三个旋钮：“**准确性**”，“**时间**”和“**可解释性**”。

该**实验设置**描述精度，时间和具体实验的可解释性。实验设置上的旋钮是可调的，因为这些值会更改左下页面上设置的含义。

以下是“实验”设置的概述：

- **准确性** -相对准确性-较高的值应导致对模型性能（准确性）的较高置信度。
- **时间** -完成实验的相对时间。较高的值将需要更长的时间才能完成实验。
- **可解释性** -向人类解释或以可理解的术语呈现的能力。可解释性越高，将要提取的特征越简单。

### 准确性

通过增加精度设置，无人驾驶AI逐渐调整执行进化和集成的方法。机器学习集合由多种学习算法组成，以获得更好的预测性能，该性能可以从任何一种学习算法获得[1]。在低精度设置下，无人驾驶AI会改变功能（来自功能工程）和模型，但它们之间的竞争是平均的。在更高的精度下，每个独立的主模型将独立发展，并作为不同主模型上的合奏成为最终合奏的一部分。在更高的精度下，无人驾驶AI会不断发展和集成，例如目标编码的开和关，它们是独立发展的。最终，无人驾驶AI以最高的准确性执行模型和功能跟踪，并整合所有这些变化。

### 时间

时间指定完成实验的相对时间（即，较高的设置需要更长的时间）。如果实验在指定的迭代次数中没有提高得分，则会提前停止。时间值越高，分配给进一步迭代的时间就越多，这意味着该配方将有更多的时间来研究特征工程和模型的超参数调整中的新转换。

### 可解释性

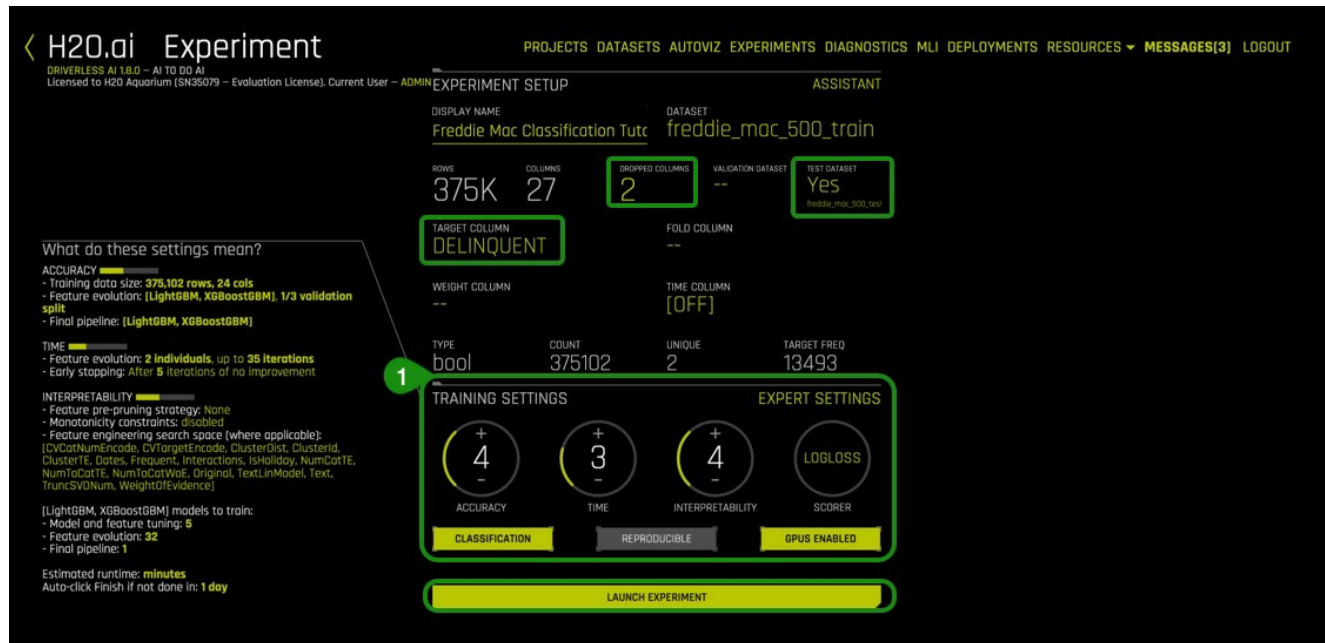
可解释性旋钮是可调的。可解释性越高，主要建模例程将从数据集中提取的特征越简单。如果可解释性足够高，则将生成单调约束模型。

2.对于本教程，请更新以下实验设置，以使其与下图匹配：

- 准确度：4
- 时间：3
- 可解释性：4

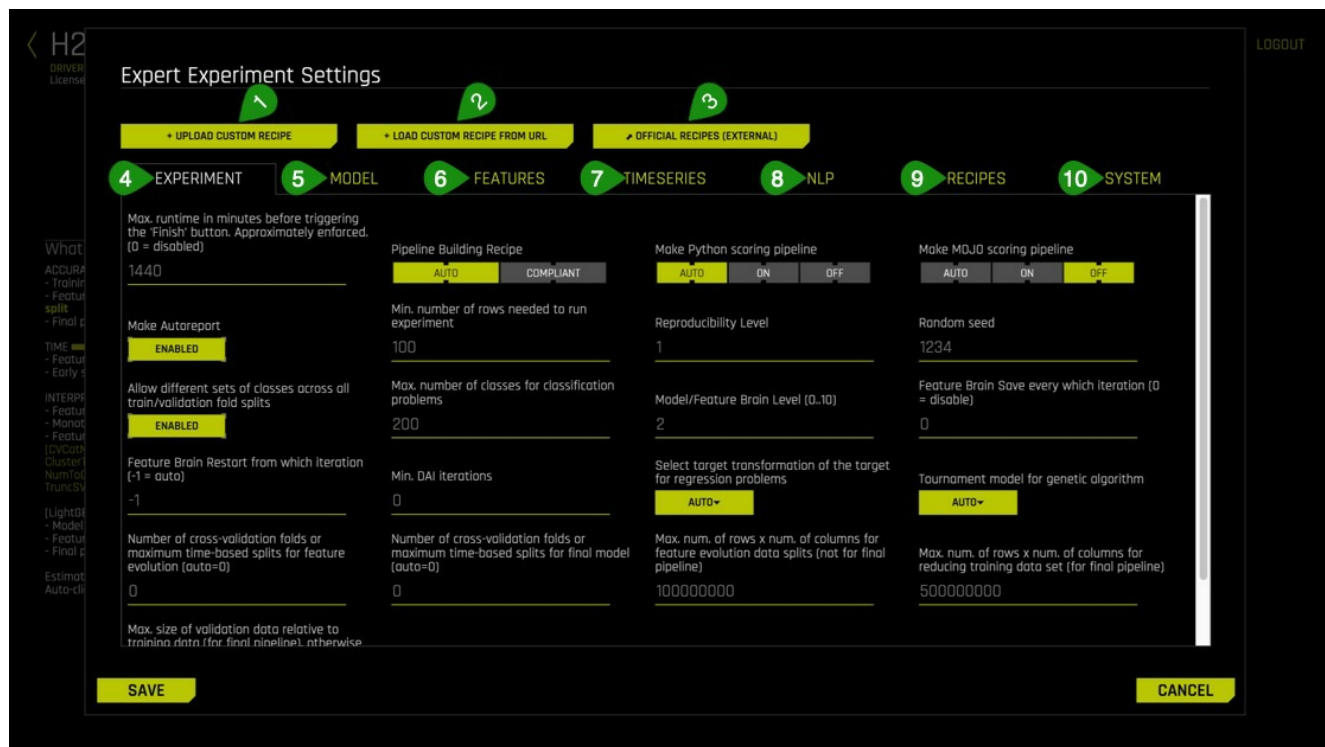
- 记分员：Logloss

选择此配置可在H2O无人驾驶测试驾驶环境中以足够的精度快速生成模型。



## 专家设定

3.将鼠标悬停在**专家设置**上，然后单击它。将会出现类似于以下图像的图像：



注意事项：

1. 上传自定义食谱
2. 从URL加载自定义配方
3. 官方食谱（外部）
4. 实验
5. 模型
6. 特征
7. 时间序列
8. 自然语言处理
9. 菜谱
10. 系统

**专家设置**是想要手动设置设置的人员可用的选项。通过单击页面顶部的选项卡，探索可用的专家设置。

**专家设置包括：**

### **实验设定**

- 触发“完成”按钮之前的最大运行时间（以分钟为单位）
- 管道建筑配方
- 制作Python计分管道
- 制作MOJO计分管道
- 进行自动报告
- 进行实验所需的最小行数
- 再现水平
- 随机种子
- 允许在所有火车/确认折叠拆分中使用不同的班级组
- 分类问题的最大类数
- 模型/功能大脑水平
- 功能大脑省去了所有迭代
- 从哪个迭代功能重新开始
- 即使重新训练最终模型，Feature Brain也会使用新列添加功能
- 最小无人驾驶AI迭代
- 为回归问题选择目标的目标转换
- 遗传算法的竞赛模型
- 为Ensemble Meta Learner启用额外日志记录
- 特征演化的交叉验证折叠数或基于时间的最大拆分数
- 最终模型的交叉验证折叠数或最大基于时间的拆分
- 日志中可显示的折叠ID的最大数量
- 特征演变数据拆分的最大行数乘以列数
- 最大行数乘以列数以减少训练数据集
- 相对于训练数据的验证数据的最大大小

- 通过toml字符串添加到config.toml

## 型号设定

- XGBoost GBM模型
- XGBoost Dart模型
- GLM模型
- LightGBM模型
- TensorFlow模型
- FTRL模型
- RuleFit模型
- LightGBM助推类型
- LightGBM分类支持
- 树/迭代的最大数量
- 最终集成GBM模型的最低学习率
- 最终集成GBM模型的最大学习率
- 特征演化过程中树/迭代数的减少因子
- 特征工程GBM模型的最低学习率
- 树模型的最大学习率
- TensorFlow / FTRL的最大纪元数
- RuleFit的最大规则数
- 最终建模管道的整体水平
- 调整阶段的模型数量
- 不平衡二元分类问题的抽样方法
- 二进制分类失衡触发特殊采样技术的多数与少数类的比率（如果启用）
- 对于严重不平衡的二元分类，如果启用，则仅启用特殊采样技术时，多数类与少数类的比率
- 二元分类不平衡抽样方法的包装袋数（如果启用）
- 特征演化阶段不平衡二元分类抽样方法的袋数硬限制
- 不平衡采样期间采样数据的最大大小
- 应用欠采样/过采样技术后的少数群体目标分数
- 二阶，三阶，四阶交互项的自动FTRL交互项的最大数量（每个）
- 启用详细的计分模型信息
- 对于具有这么多类的分类问题，默认为TensorFlow

## 功能设定

- 特色工程工作
- 数据分配移位检测
- 数据分布偏移检测功能下降
- 删除功能之前的最大允许功能移位（AUC）
- 泄漏检测
- 泄漏检测下降AUC / R2阈值

- 最大行泄漏次数列
- 报告原始特征的排列重要性
- 执行基于排列的特征选择的最大行数
- 使用的原始功能的最大数量
- 原始非数值功能的最大数量
- FS个人使用的最大原始功能数
- 触发要素选择模型类型的原始数字要素数量
- 触发要素选择模型类型的原始非数值要素数量
- 整数列和分类列的最大允许唯一分数
- 整数/浮点数归类的最大唯一值数
- 最大工程数量
- 超出触发单调性约束的相关性（如果启用）
- 最大功能交互深度
- 启用目标编码
- 启用字典标签编码
- 启用隔离林异常分数编码
- 启用一种HotEncoding
- 隔离林编码的估计量
- 删除常量列
- 删除ID列
- 不要删除任何列
- 删除功能
- 启用详细的评分功能信息
- 为生成的功能的时间和类型启用详细日志

## 时间序列设定

- 时间序列滞后配方
- 生成假期功能
- 时间序列滞后覆盖
- 最小的滞后尺寸
- 从时间列启用要素工程
- 从“整数时间”列启用要素工程
- 允许将日期或时间要素直接转换为数字表示形式
- 将“时间组”列视为独立功能
- 哪些TGC功能类型被视为独立功能
- 启用时间未知的变压器
- 始终按所有时间组分组列以创建滞后特征
- 生成时间序列保持预测
- 用于创建最终时间序列模型的保留预测的最大拆分数
- 滞后功能的辍学模式
- 创建非目标滞后特征的可能性

- 创建滚动测试集预测的方法
- 新的时序变压器使用默认滞后的可能性
- 探索基于交互的滞后变压器的可能性
- 探索基于聚集的滞后变压器的可能性

## NLP设置

- NLP的Max TensorFlow时代
- 默认情况下，所有型号的准确性均高于启用TensorFlow NLP
- 为NLP启用基于单词的CNN TensorFlow模型
- 为NLP启用基于单词的BiGRU TensorFlow模型
- 为NLP启用基于字符的CNN TensorFlow模型
- TensorFlow NLP模型的预训练嵌入路径
- 允许训练未冻结的预训练嵌入
- Python / MOJO评分运行时是否将具有GPU
- 所有特征中的文本列的分数均被认为是文本主导问题
- 所有变形金刚的分数，以触发该文本占主导地位
- 字符串列的阈值被视为文本

## 食谱设置

- 包括特定的变压器
- 包括特定型号
- 包括特定得分手
- 添加变压器的可能性
- 添加最佳共享变压器的可能性
- 修剪变压器的概率
- 突变模型参数的可能性
- 修剪弱功能的可能性
- 以分钟为单位的超时时间，用于测试每个食谱的接受程度
- 是否跳过变压器故障
- 是否跳过模型故障
- 记录跳过故障的级别

## 系统设置

- 使用的核心数
- 用于模型拟合的最大芯数
- 用于模型预测的最大核数
- 在执行MLI，自动报告，对另一个数据集评分时用于模型转换和预测的最大内核数
- 每批为CPU调整工人
- # GPU /实验
- # GPU /型号

- 用于数据表和用于修补和模型训练的OpenBLAS的最大线程数
- GPU起始编号
- 启用详细跟踪
- 启用调试日志级别

4.对于此实验，请打开RuleFit模型，在“模型”选项卡下，选择“保存”。

RuleFit [2]算法通过首先拟合树模型，然后拟合Lasso（L1正规化）GLM模型来创建由最重要的树叶（规则）组成的线性模型，从而创建最佳的决策规则集。RuleFit模型有助于在保留决策树的可解释性的同时，超越随机森林的准确性。

打开RuleFit模型将被添加到无人驾驶AI将为实验考虑的算法列表中。算法的选择取决于所选的数据和配置。

5.在选择启动之前，请确保您的实验页面看起来与下面的相似，一旦准备好，请点击启动。

通过访问H2O的文档- [专家设置](#)，了解有关每种设置的含义以及如何从默认值进行更新的更多信息

## 资源资源

[1] 合奏学习

[2] J. Friedman, B. Popescu. “通过规则集合进行预测性学习”。2005年

## 深潜

---

背部下一个