# “魔镜杯”风控算法大赛

团队：NEMO

成员：燕鹏，陈朝才，王哲

# 目录

# 题目背景

预测内容：用户信用好坏

直接预测目标：用户是否正常还款

应用场景：互联网金融借贷

二分类，评价指标：Auc

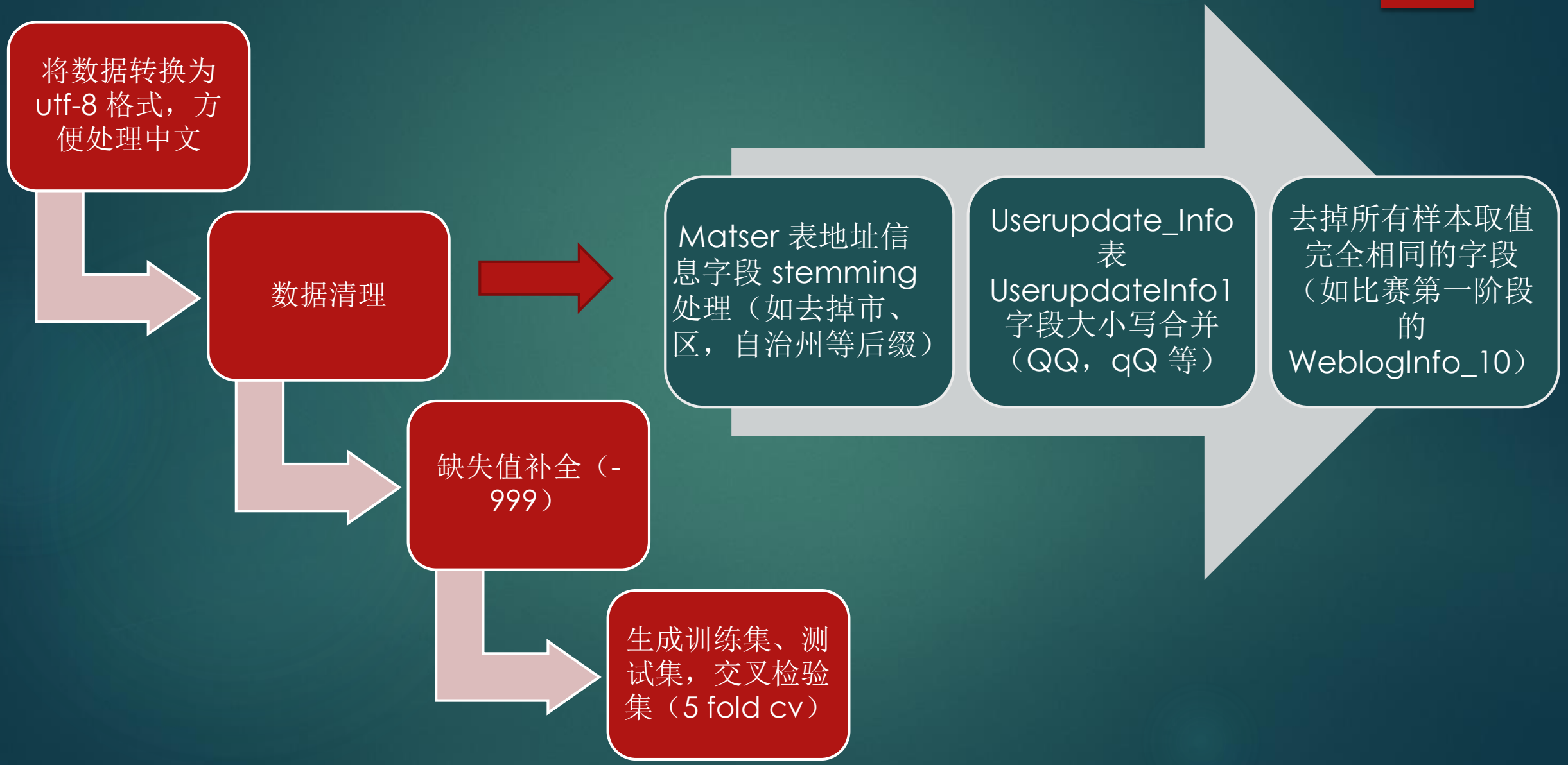
## 数据信息

Master 表：借款人特征字段、网络行为、学历学籍、社交网络信息、第三方信息以及借款成交时间

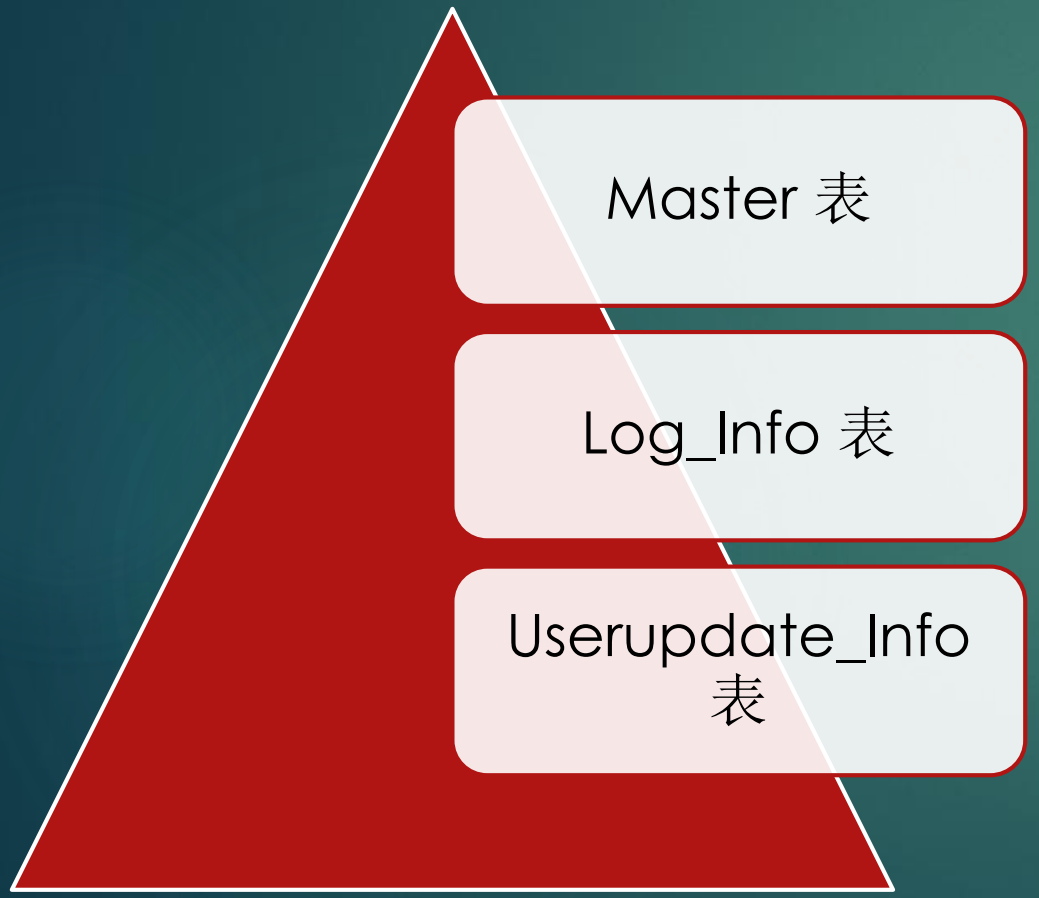
Log_Info 表：借款人登录信息（具体操作、操作所属类别，对应时间）

Userupdate_Info 表：借款人修改信息操作记录（修改内容和修改时间）

# 数据预处理


将数据转换为utf-8 格式，方便处理中文
数据清理
缺失值补全（-999）
生成训练集、测试集，交叉检验集（5 fold cv）
Matser 表地址信息字段 stemming 处理（如去掉市、区，自治州等后缀）
Userupdate_Info 表 UserupdateInfo1 字段大小写合并（QQ，qQ 等）
去掉所有样本取值完全相同的字段（如比赛第一阶段的 WeblogInfo_10）

# 特征工程 — 特征提取




| Master 表 | Log_Info 表 | Userupdate_Info 表 |
| --- | --- | --- |
| • 基础特征<br>  • 基础数值特征<br>  • 字符串特征转换<br>• 统计值特征<br>• 地理信息特征<br>• 第三方数据深度挖掘 | • 以用户 id 为基本单元，提取<br>  • LogInfo1，LogInfo2，LogInfo3 中不同特征字段的出现次数<br>  • LogInfo3 时间信息中的最早时间、最晚时间，以及时间差 | • 以用户 id 为基本单元，提取<br>  • LogInfo1，LogInfo2，LogInfo3 中不同特征字段的出现次数<br>  • LogInfo3 时间信息中的最早时间、最晚时间，以及时间差 |

# 特征工程 — 特征提取（Master 表）

## 基础数值特征

- 树模型：直接使用
- 其他模型（LR FFM NN 等）数据归一化

## 字符串特征转换

- 转换为连续整数值特征（pandas 的 factorize 函数或 sklearn 的 LabelEncoder 函数）
- 转换为多维向量（one-hot）

## 统计特征

- 从字符串特征、类特征两个角度进行特征统计。以某一维特征为基础统计单元，将该维度特征中所有不重复字段在数据样本中出现的次数作为特征
- 将连续数值特征做上述处理，不同的是，我们需要做两项特殊操作：
  - 连续数值离散化；
  - 设置特征选择的阈值，只保留那些不同值个数在一定范围的特征

# 特征工程 — 特征提取（Master 表）

## 地理信息特征

- 地理位置信息映射到城市等级
- 地理位置信息映射到经纬度
- 各个层级的地理位置对应的 GDP 情况

## 第三方数据深度挖掘

- 数据描述：ThirdParty_Info_Period[1-7]_[1-17] 共 119 维特征
- 从 ThirdParty_Info_Period[1-7]_j(j from 1 to 17) 和 ThirdParty_Info_Periodi(i from 1 to 7)_[1-17] 两个角度统计 7 维特征和 17 维特征中的所有样本数据的 max、min、avg、median 作为特征
- 经过大量实验和分析，ThirdParty_Info_Period[1-7]_j(j from 1 to 17) 角度的统计特征更为重要，这个字段很可能表示的是不同时间段（1-7）不同类型的第三方数据（1-17）的用户信息

# 特征工程 — 特征选择

## 树模型

利用树模型进行特征重要性排序（决策树分叉过程中选择某个特征的次数），作为特征选择的依据

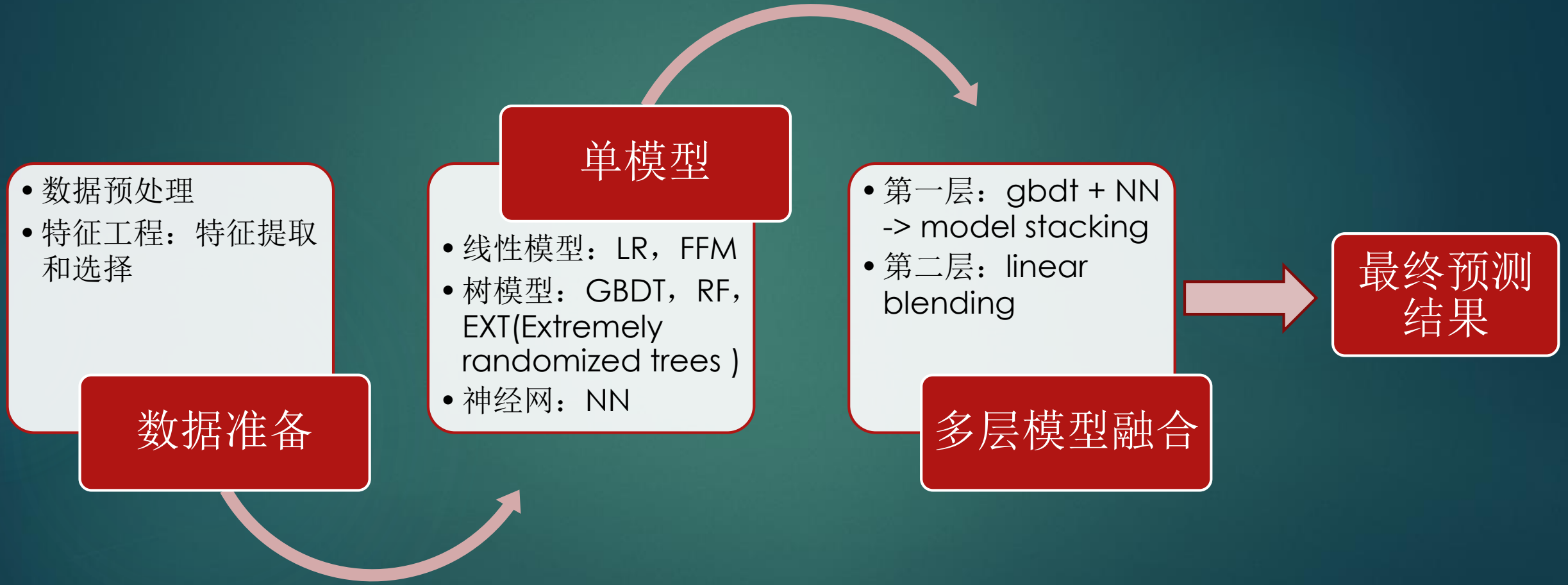
## 特征 bagging

在全部特征集合中随机选取一定比例特征，训练多个模型，并进行线性融合

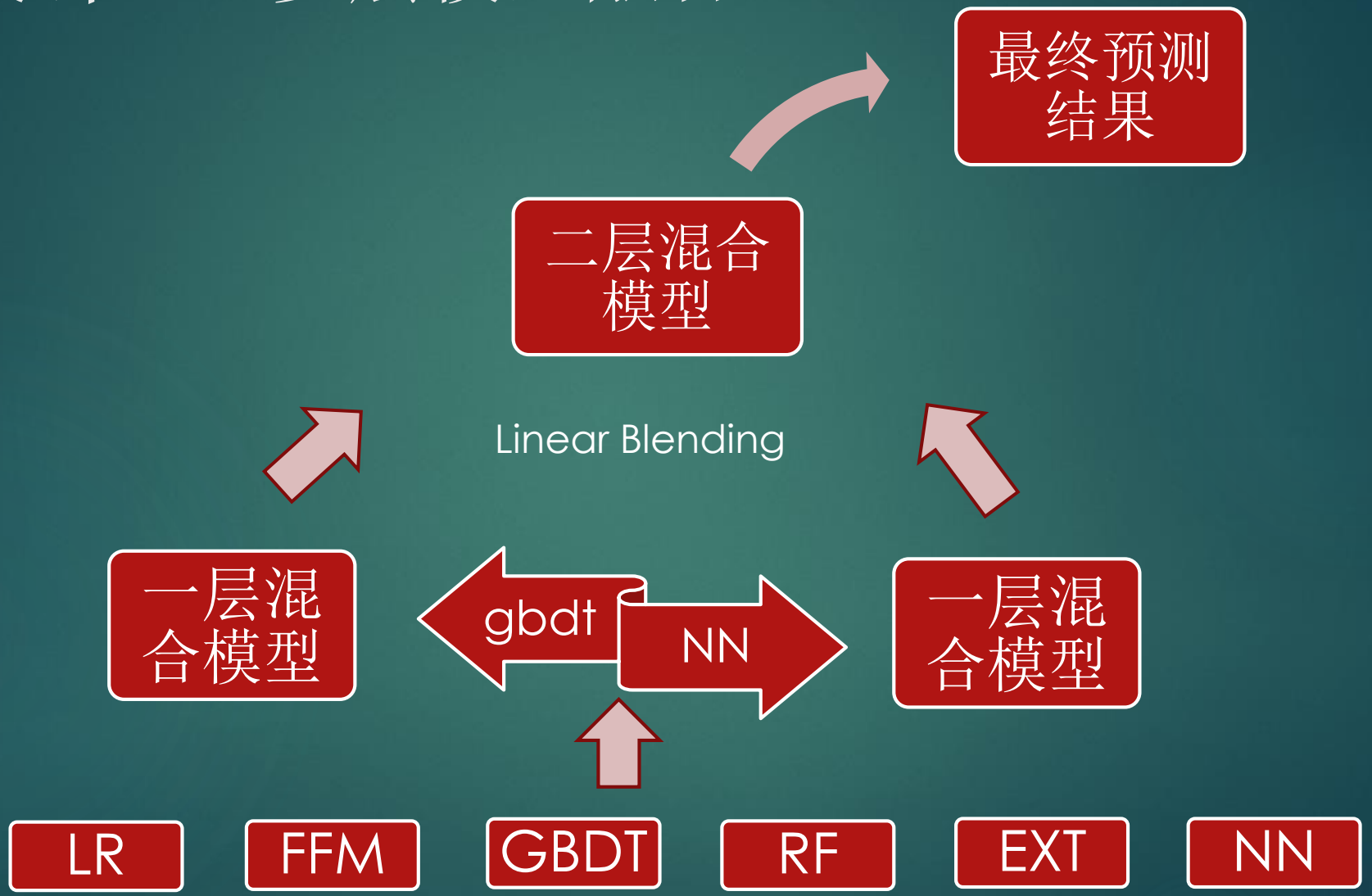
## 随机 seed

通过选取多个随机 seed 的方式增加模型训练过程中特征选择的随机性，并通过模型融合的方法增强预测能力的稳定性

# 模型设计 — 算法框架


• 数据预处理
• 特征工程：特征提取和选择
数据准备
单模型
• 线性模型：LR，FFM
• 树模型：GBDT，RF，EXT(Extremely randomized trees )
• 神经网：NN
• 第一层：gbdt + NN -> model stacking
• 第二层：linear blending
多层模型融合
最终预测结果

# 模型设计 — 多层模型融合


最终预测结果
二层混合模型
Linear Blending
一层混合模型
gbdt
NN
一层混合模型
LR
FFM
GBDT
RF
EXT
NN

# 模型设计 — 多层模型融合（详解）

- 首先，各个单模型独立完成训练，并得出预测结果
- 接下来，分别用 gbdt 和 NN 作为第一层融合模型，并将上述预测结果作为输入特征，训练模型（model stacking）
- 将新得到的两个模型（gbdt 和 NN）做线性融合（linear blending），作为最终的预测模型
- 利用上述模型对测试数据进行预测，得出最终预测结果

# 算法总结

- 1. 准确性：采用 5/10 fold 交叉验证集进行线下评估确保模型预测准确性；
- 2. 稳定性：采用特征 bagging、随机 seed 以及多层模型融合的机制提高算法稳定性；
- 3. 创新性：深入细致的特征工程：数据清理、统计特征、地理位置的映射，第三方数据的深度挖掘以及合理有效的特征选择；
- 4. 实用性：所有算法都基于开源代码实现，实用性强；
- 5.探索性：算法框架涉及线性模型、树模型以及神经网络等多个领域，并采用多层模型融合机制，很多都是工业界尚未推广使用却十分有效的方法，具有很强的探索性和前瞻性；
- 6. 科学性：特征提取过程中，没有引入未来数据，且线下实验流程规范合理可复现；

# Thanks

谢谢！