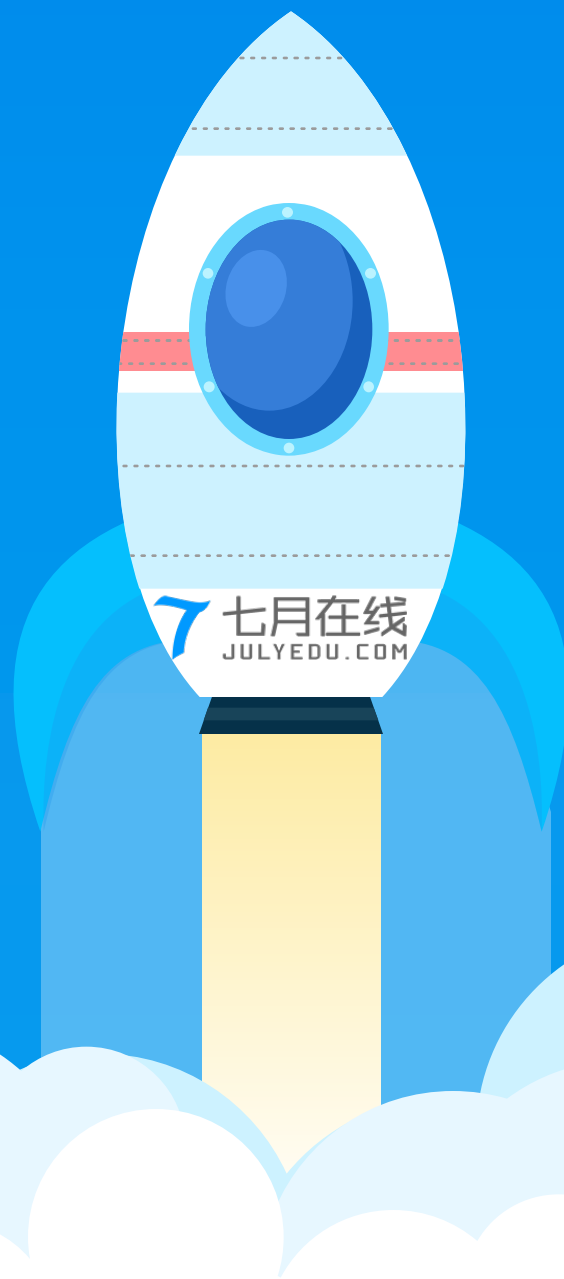


机器学习集训营

机器学习模型部署与案例

刘老师

<https://www.julyedu.com/>



目录



Part 1

机器学习的训练和部署



Part 2

XGBoost和LightGBM高阶使用



Part 3

模型调参与误差分析



Part 4

机器学习模型部署案例

/01 机器学习的训练与部署

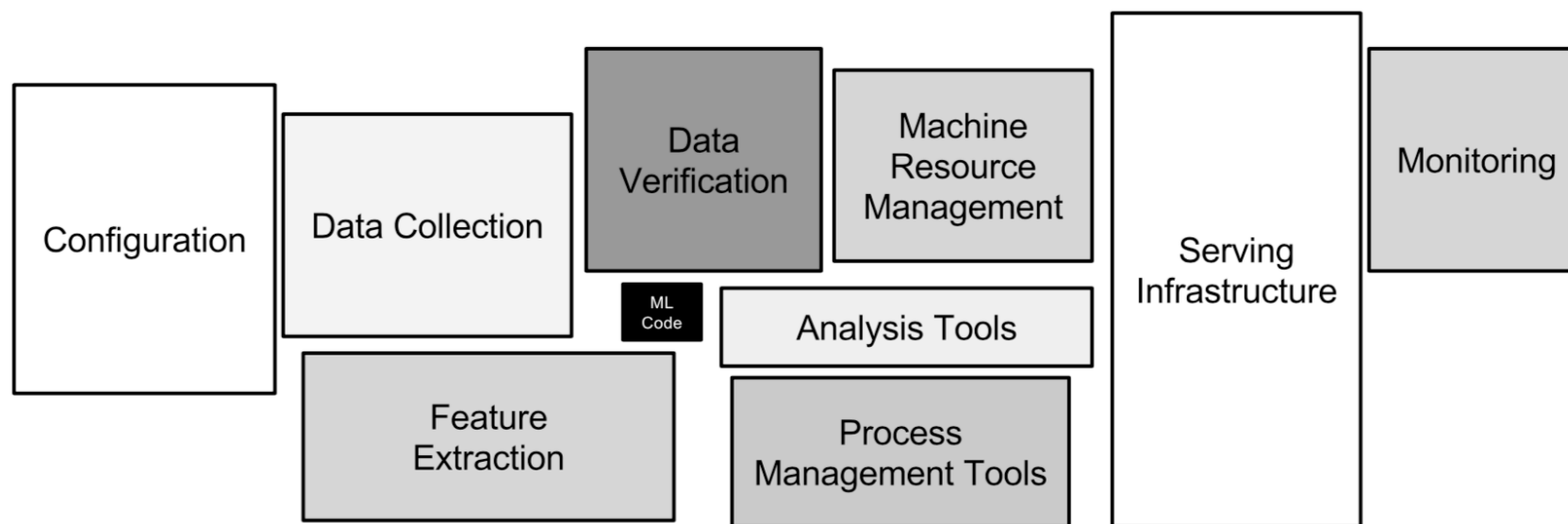
✓ 机器学习训练与部署细节

✓ 机器学习部署要点

Part 1 机器学习的训练与部署

机器学习系统是现实生活中重要的一部分，但也是较小的一部分，占据10%左右的代码；

Hidden Technical Debt in Machine Learning Systems, Google



Part 1 机器学习的训练与部署

机器学习系统是复杂的端到端的工作流程：

- ✓ 机器学习是复杂的，包括数据处理、多次实验构建和部署监控；
数据工程师（大数据开发）、算法工程师、数据运维工程师；
- ✓ 机器学习是大规模数据的应用，需要大规模训练和大规模部署；
- ✓ 机器学习是实验性应用，需要对实验进行跟踪并进行记录；

Part 1 机器学习的训练与部署

机器学习工作流的重要组成：


- ✓ 工作流需要迭代、拆分和记录保存；
- ✓ 工作流需要进行版本化和保证可复制；
- ✓ 工作流需要将训练和预测分开；

思考：如何保证模型可复现？如何保证？

Part 1 机器学习的训练与部署

机器学习部署方法，与具体的场景和要求相关。

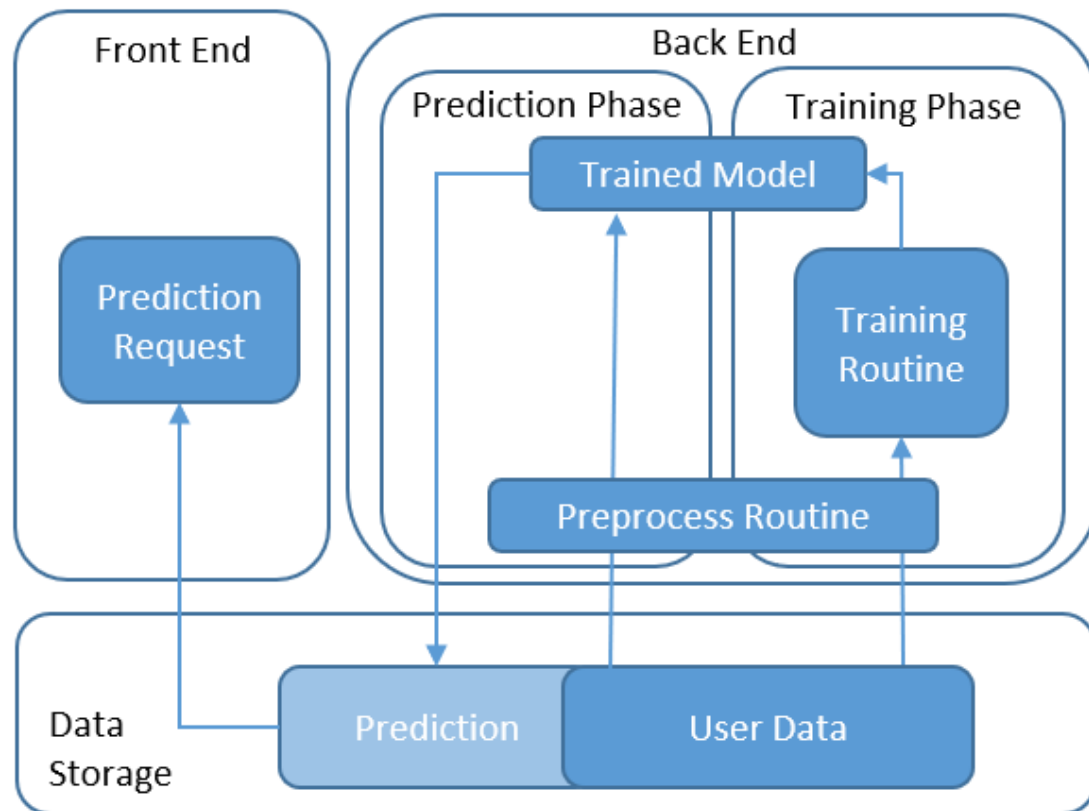
- ✓ 调用方法：调用API，REST API；
- ✓ 预测数量：批量预测，实时预测；
- ✓ 线上学习：是否支持线上增量学习；

Name	Learning	Prediction	Complexity
1. Store prediction in DB	Batch	Batch	Simple  Complex
2. Prediction is on model object	Batch	Real-time (Use model object)	
3. Prediction is on API	Batch	Real-time (REST API call)	
4. Real-time learning	Real-time	Real-time	

Part 1 机器学习的训练与部署

机器学习部署方法：数据库或离线批量预测

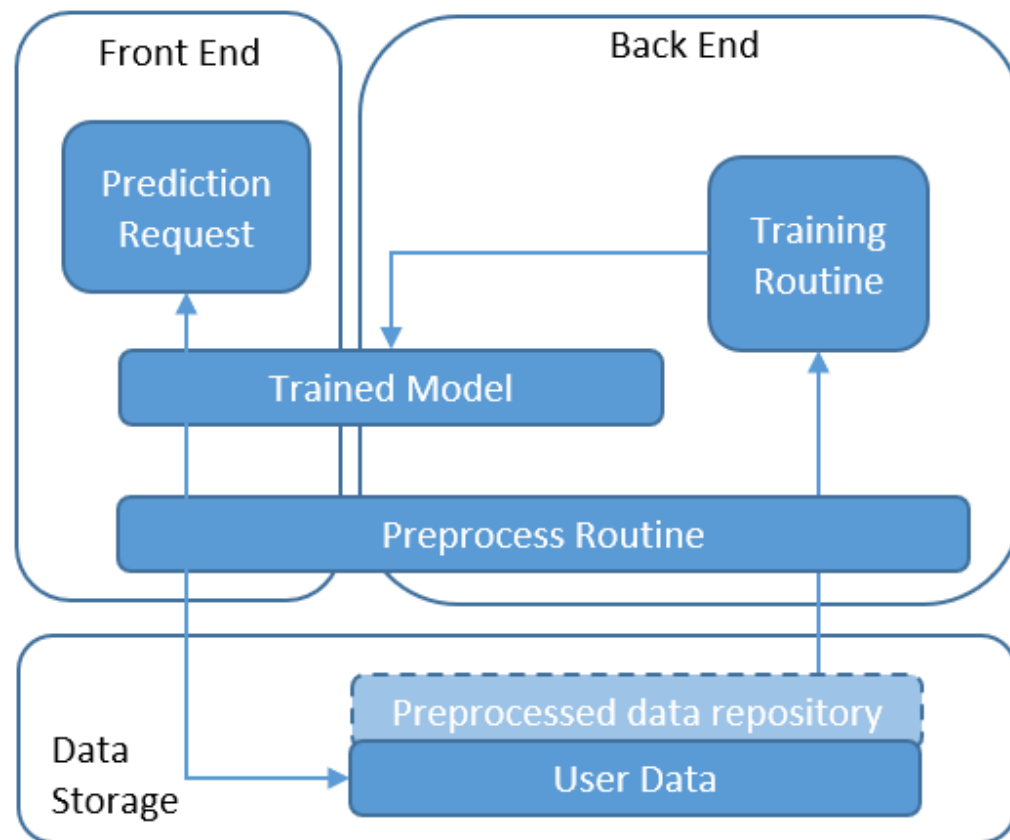
- ✓ 通过离线任务完成训练与预测；
- ✓ 使用评率少，周期长，运行时间长；
- ✓ 简单可控适合周期性任务；



Part 1 机器学习的训练与部署

机器学习部署方法：基于模型完成预测

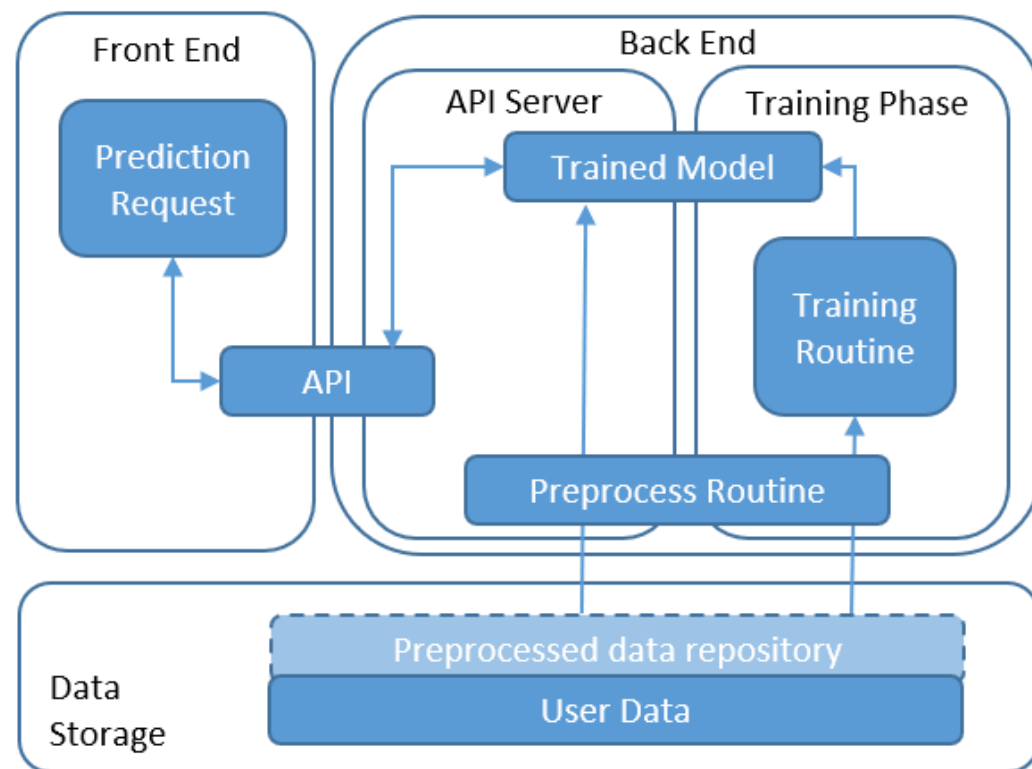
- ✓ 将模型放在后端响应中；
- ✓ 可以实时对请求进行预测；
- ✓ 如果模型更新，需要重新部署；



Part 1 机器学习的训练与部署

机器学习部署方法：基于REST API预测

- ✓ 模型部署与调用分开；
- ✓ 通过请求调用服务；
- ✓ 比较灵活但需要额外的设计；



/02 XGBoost与LightGBM高阶使用

- ✓ XGBoost高阶使用；
- ✓ LightGBM高阶使用；
- ✓ CatBoost高阶使用

Part 2 XGBoost与LightGBM高阶使用

- XGBoost , <https://xgboost.readthedocs.io/>
- 参数介绍 : <https://xgboost.readthedocs.io/en/latest/parameter.html>

- LightGBM , <https://lightgbm.readthedocs.io/en/latest/>
- 参数介绍 : <https://lightgbm.readthedocs.io/en/latest/Parameters.html>

- CatBoost , <https://yandex.com/dev/catboost/>
- 参数介绍 : <https://catboost.ai/docs/>

常见面试题 : XGBoost、LightGBM、CatBoost三者之间的区别和联系是什么 ?

Part 2 XGBoost与LightGBM高阶使用

LightGBM高阶使用：

- ✓ 模型参数理解与参数调整；
- ✓ 模型保存与加载；
- ✓ 模型微调；
- ✓ 模型可视化与特征重要性；

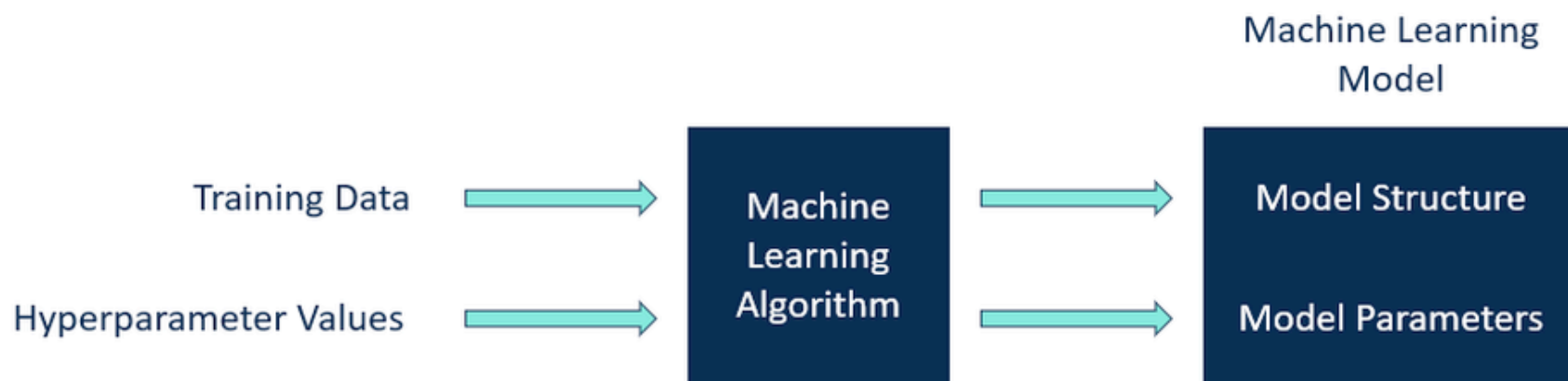
/03 模型调参与误差分析

- ✓ 模型调参方法
- ✓ 误差分析方法
- ✓ 特征筛选方法

Part 2 模型调参方法

模型参数 vs 模型超参数：

- ✓ 模型参数（Model parameter）：通过数据可以学习到的参数；
- ✓ 模型超参数（Model hyperparameters）：需要人为设定的参数，无法通过数据进行学习；



<https://towardsdatascience.com/hyperparameters-optimization-526348bb8e2d>

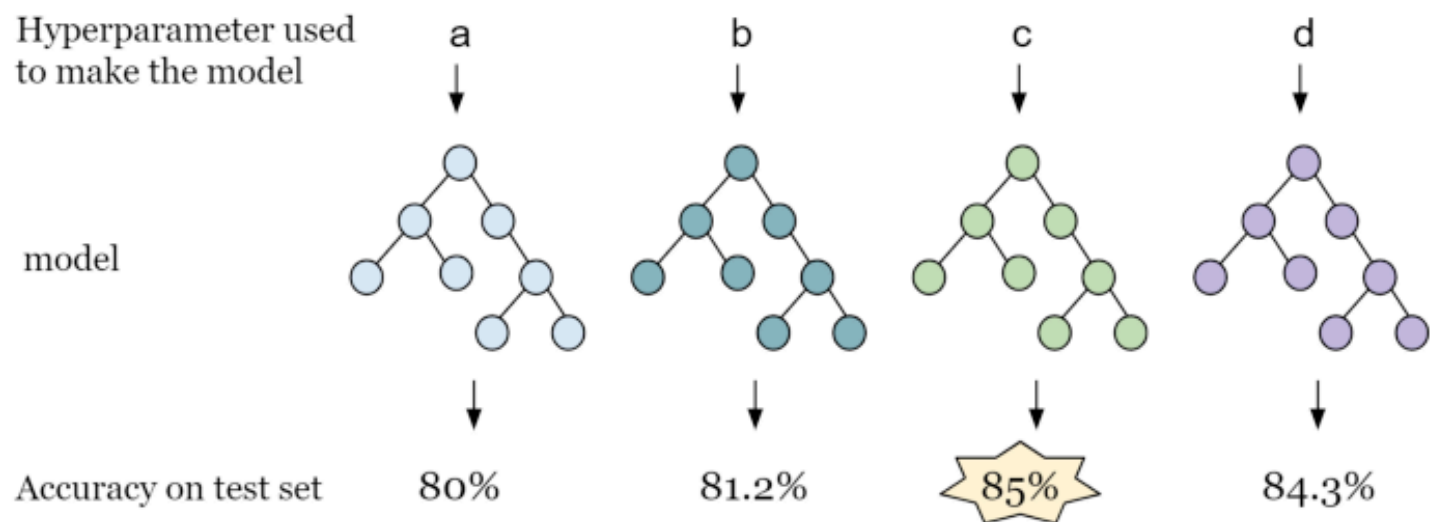
Part 3 模型调参与误差分析

模型超参数选择：通过验证集精度选择模型参数，类似人工筛选；

✓ 优点：靠谱的方法，需要较少的计算资源；

✓ 缺点：需要人工参与和人工知识；

```
Test_Hyperparameters = [a, b, c, d]
```



Part 3 模型调参与误差分析

模型超参数选择：通过网格搜索和随机搜索选择参数；

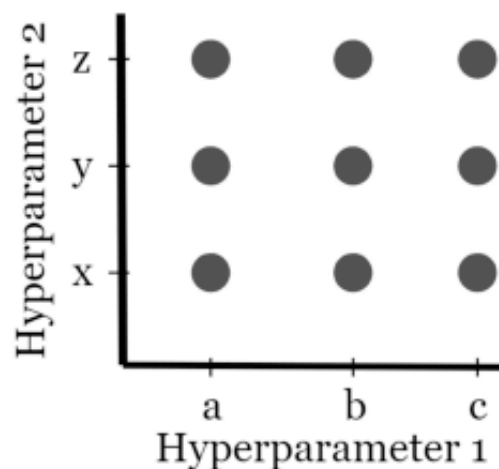
✓ 优点：对参数空间进行完备的搜索；

✓ 缺点：计算量比较大；

Grid Search

Pseudocode

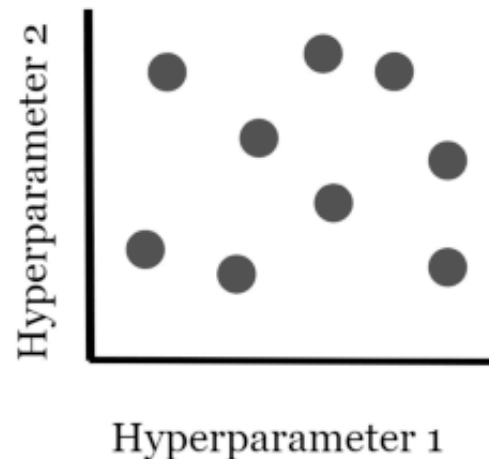
```
Hyperparameter_One = [a, b, c]  
Hyperparameter_Two = [x, y, z]
```



Random Search

Pseudocode

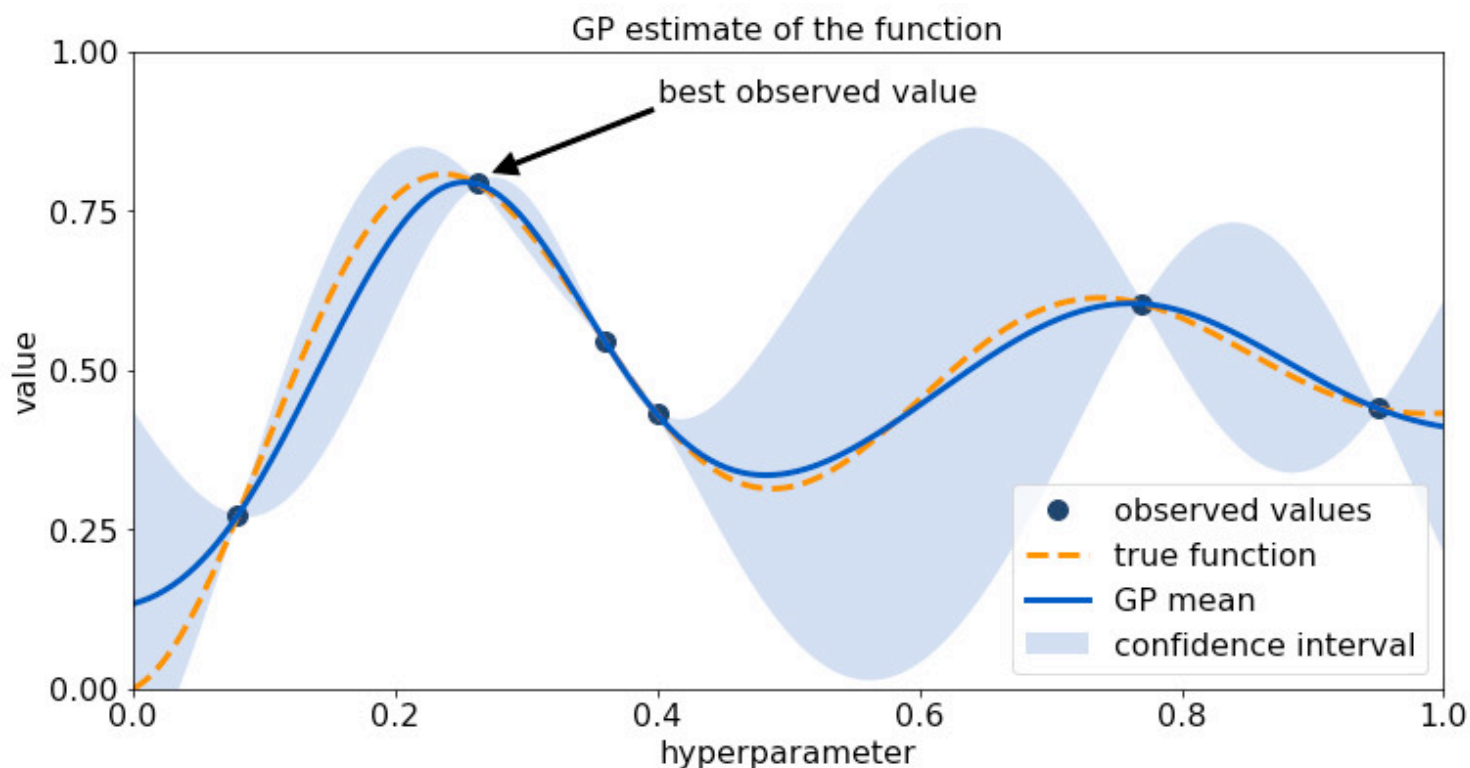
```
Hyperparameter_One = random.num(range)  
Hyperparameter_Two = random.num(range)
```



Part 3 模型调参与误差分析

模型超参数选择：通过贝叶斯优化或遗传算法

- ✓ 优点：能够减少参数搜索空间；
- ✓ 缺点：计算量较大；



Part 3 模型调参与误差分析

误差如何定量分析 & 特征如何筛选

尽可能不要产生很多的特征，控制特征维度；其次如果有很多特征了，可以筛选一部分特征；

- ✓ Mean Decrease Impurity
- ✓ Permutation importance
- ✓ Partial dependence plots
- ✓ SHAP Values
- ✓ Boruta
- ✓ ELI5
- ✓ Null Importance

<https://www.kaggle.com/ml-for-insights-signup>

<https://www.jianshu.com/p/324a7c982034>

<https://www.kaggle.com/ogreliier/feature-selection-with-null-importances>

Part 3 模型调参与误差分析

LightGBM 由微软提出，主要解决 GDBT 在海量数据中遇到的问题，可以更好更快地用于工业实践中。

□ LightGBM的贡献

- 单边梯度抽样算法；
- 直方图算法；
- 互斥特征捆绑算法；
- 深度限制的 Leaf-wise 算法；
- 类别特征最优分割；
- 特征并行和数据并行；
- 缓存优化；



<https://papers.nips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf>

<https://lightgbm.readthedocs.io/en/latest/>

Part 3 模型调参与误差分析

□ LightGBM的贡献：单边梯度抽样算法

- ✓ 对样本进行采样，选择部分梯度小的样本；
- ✓ 让模型关注梯度高的样本，减少计算量；

Algorithm 2: Gradient-based One-Side Sampling

Input: I : training data, d : iterations
Input: a : sampling ratio of large gradient data
Input: b : sampling ratio of small gradient data
Input: $loss$: loss function, L : weak learner
 $models \leftarrow \{\}$, $fact \leftarrow \frac{1-a}{b}$
 $topN \leftarrow a \times \text{len}(I)$, $randN \leftarrow b \times \text{len}(I)$
for $i = 1$ **to** d **do**
 $preds \leftarrow models.predict(I)$
 $g \leftarrow loss(I, preds)$, $w \leftarrow \{1, 1, \dots\}$
 $sorted \leftarrow \text{GetSortedIndices}(\text{abs}(g))$
 $topSet \leftarrow sorted[1:topN]$
 $randSet \leftarrow \text{RandomPick}(sorted[topN:\text{len}(I)], randN)$
 $usedSet \leftarrow topSet + randSet$
 $w[randSet] \times = fact$ ▷ Assign weight $fact$ to the small gradient data.
 $newModel \leftarrow L(I[usedSet], -g[usedSet], w[usedSet])$
 $models.append(newModel)$

Part 3 模型调参与误差分析

□ LightGBM的贡献：直方图算法

- ✓ 将连续特征离散化，用直方图统计信息；
- ✓ 对内存、速度都友好；

Algorithm 1: Histogram-based Algorithm

Input: I : training data, d : max depth

Input: m : feature dimension

$nodeSet \leftarrow \{0\}$ \triangleright tree nodes in current level

$rowSet \leftarrow \{\{0, 1, 2, \dots\}\}$ \triangleright data indices in tree nodes

for $i = 1$ **to** d **do**

for $node$ **in** $nodeSet$ **do**

$usedRows \leftarrow rowSet[node]$

for $k = 1$ **to** m **do**

$H \leftarrow \text{new Histogram}()$

\triangleright Build histogram

for j **in** $usedRows$ **do**

$bin \leftarrow I.f[k][j].bin$

$H[bin].y \leftarrow H[bin].y + I.y[j]$

$H[bin].n \leftarrow H[bin].n + 1$

 Find the best split on histogram H .

 ...

 Update $rowSet$ and $nodeSet$ according to the best split points.

 ...

Part 3 模型调参与误差分析

□ LightGBM的贡献：互斥特征捆绑算法

- ✓ 使用互斥捆绑算法将特征绑定，降低复杂度；
- ✓ 将特征绑定视为图着色问题，计算特征之间的冲突值；
- ✓ 将特征增加增加偏移量，然后一起相加分桶；

Algorithm 4: Merge Exclusive Features

Input: $numData$: number of data

Input: F : One bundle of exclusive features

$binRanges \leftarrow \{0\}$, $totalBin \leftarrow 0$

for f **in** F **do**

$totalBin += f.numBin$

$binRanges.append(totalBin)$

$newBin \leftarrow new \text{ Bin}(numData)$

for $i = 1$ **to** $numData$ **do**

$newBin[i] \leftarrow 0$

for $j = 1$ **to** $len(F)$ **do**

if $F[j].bin[i] \neq 0$ **then**

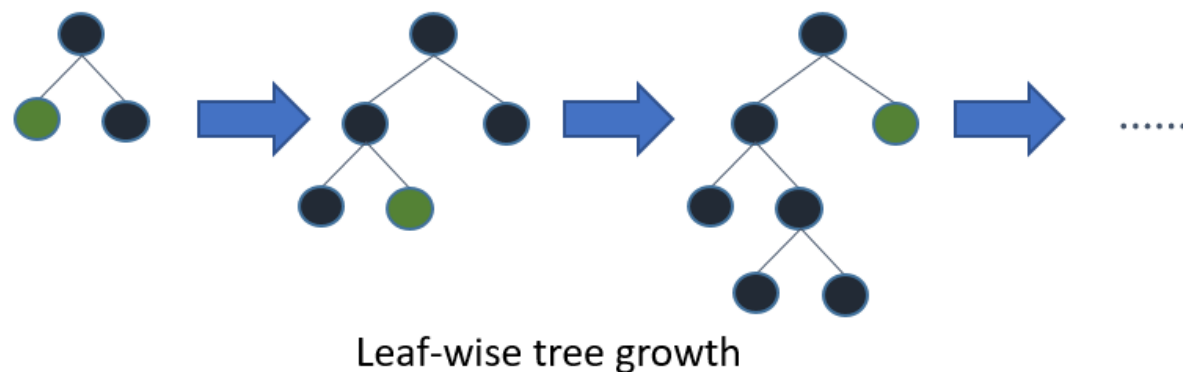
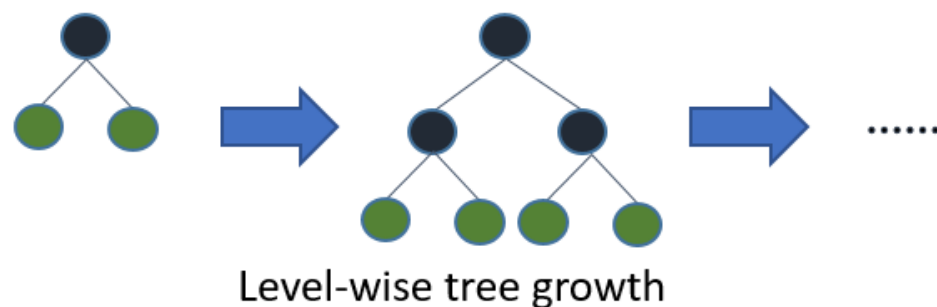
$newBin[i] \leftarrow F[j].bin[i] + binRanges[j]$

Output: $newBin, binRanges$

Part 3 模型调参与误差分析

□ LightGBM的贡献：深度限制的 Leaf-wise 算法

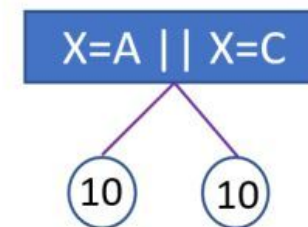
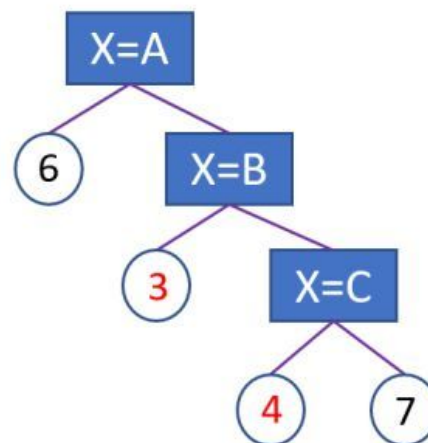
- ✓ 每次分裂增益最大的叶子节点，直到达到停止条件；
- ✓ 限制树模型深度，每次都需要计算增益最大的节点；



Part 3 模型调参与误差分析

□ LightGBM的贡献：类别特征最优分割

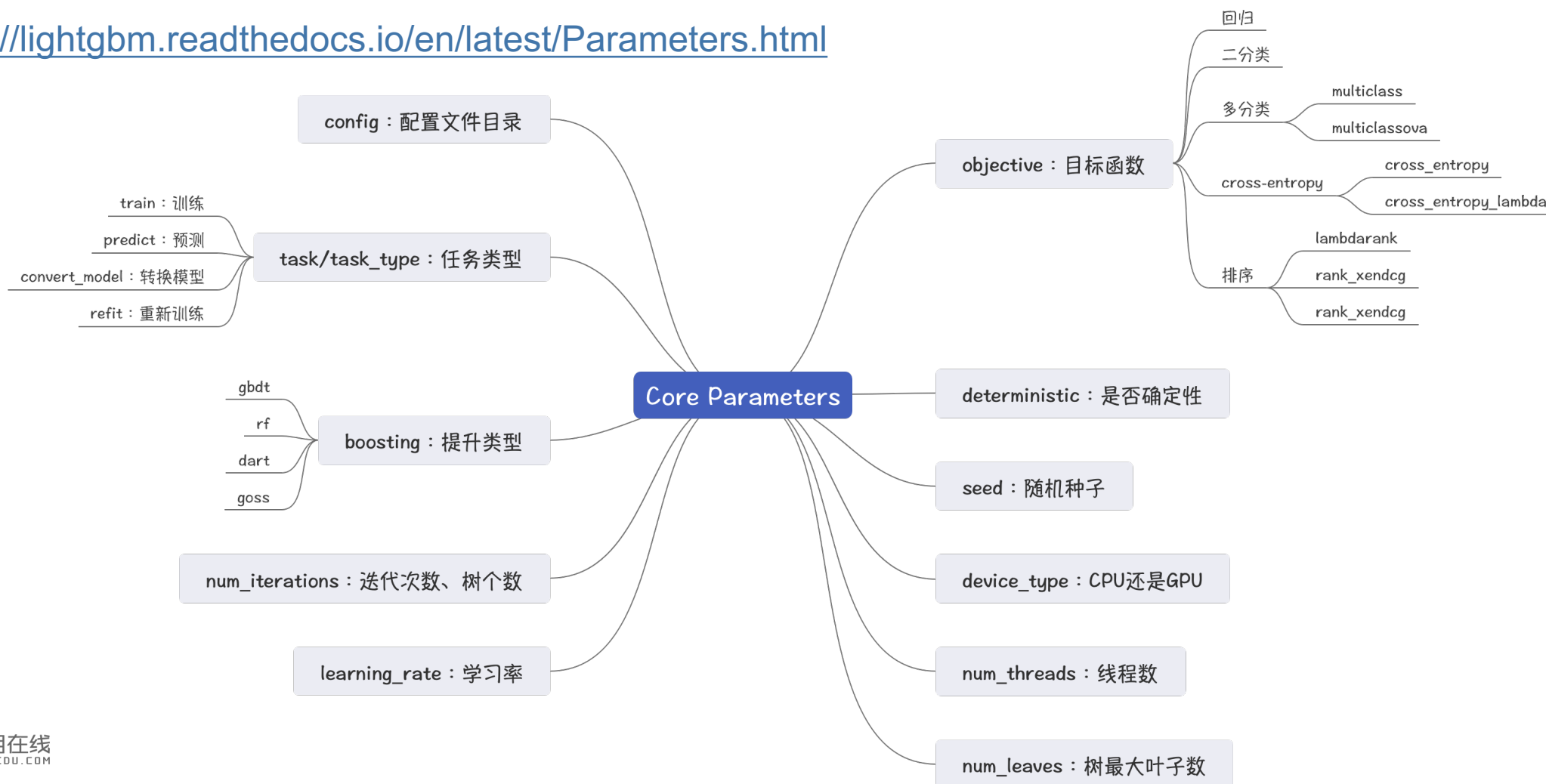
- ✓ 不用提前将类别one-hot；
- ✓ 将类别多对多的分类；
- ✓ 分裂过程考虑到类别对应的标签分布情况；



Note: Numbers in circles represent to the #data in that node

Part 3 模型调参与误差分析

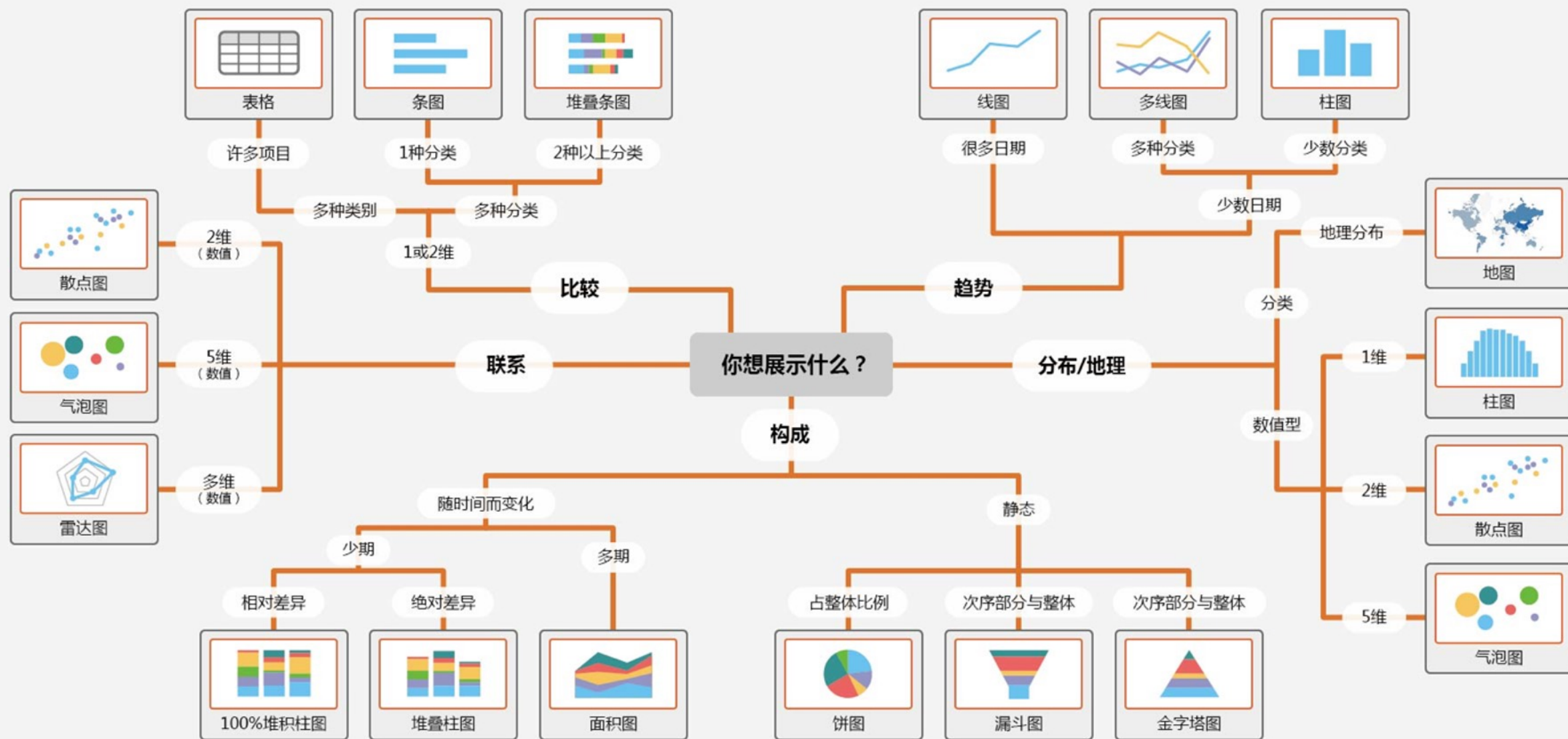
<https://lightgbm.readthedocs.io/en/latest/Parameters.html>



/04 机器学习模型部署案例

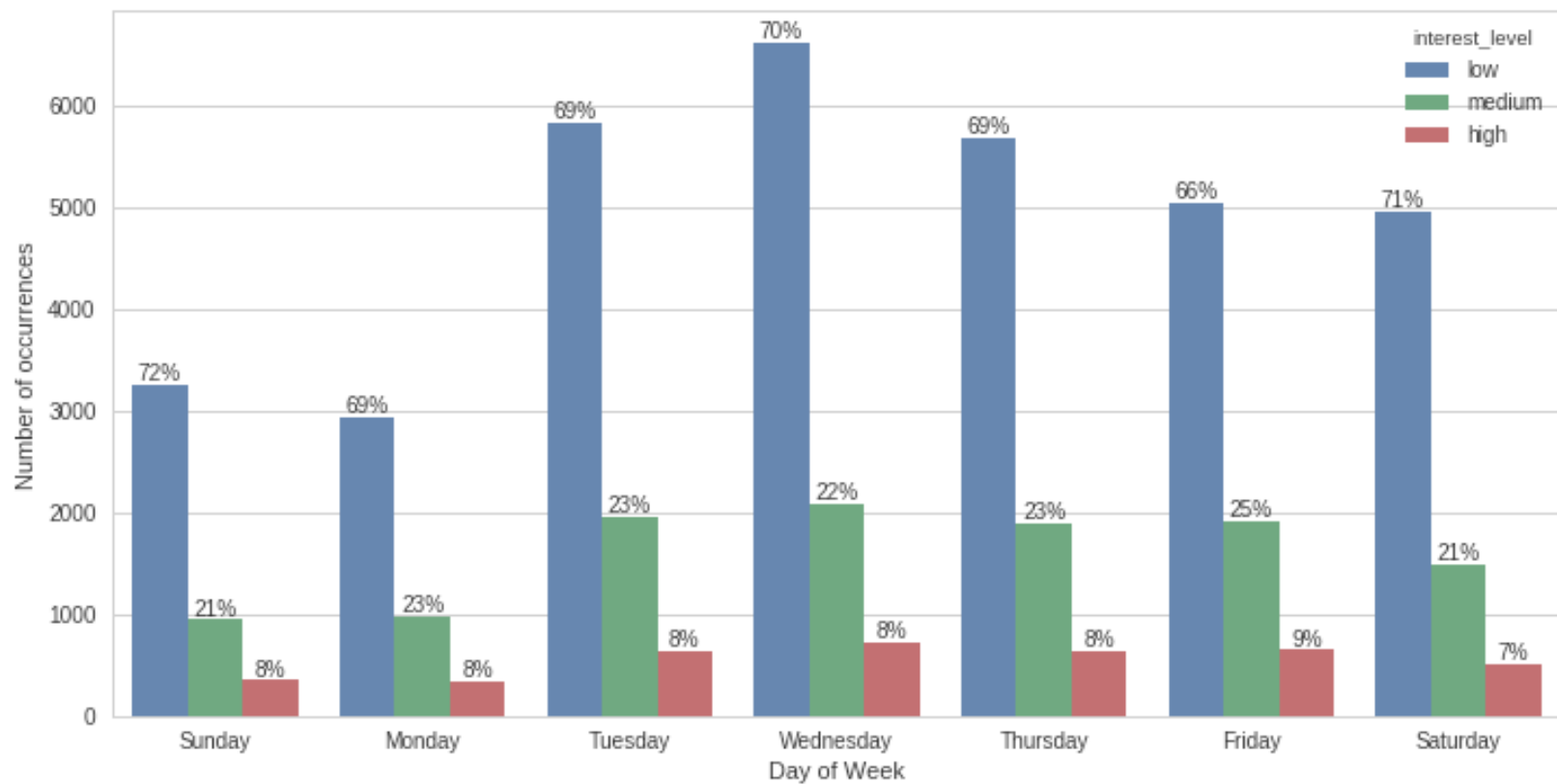


如何选择图表的类型？



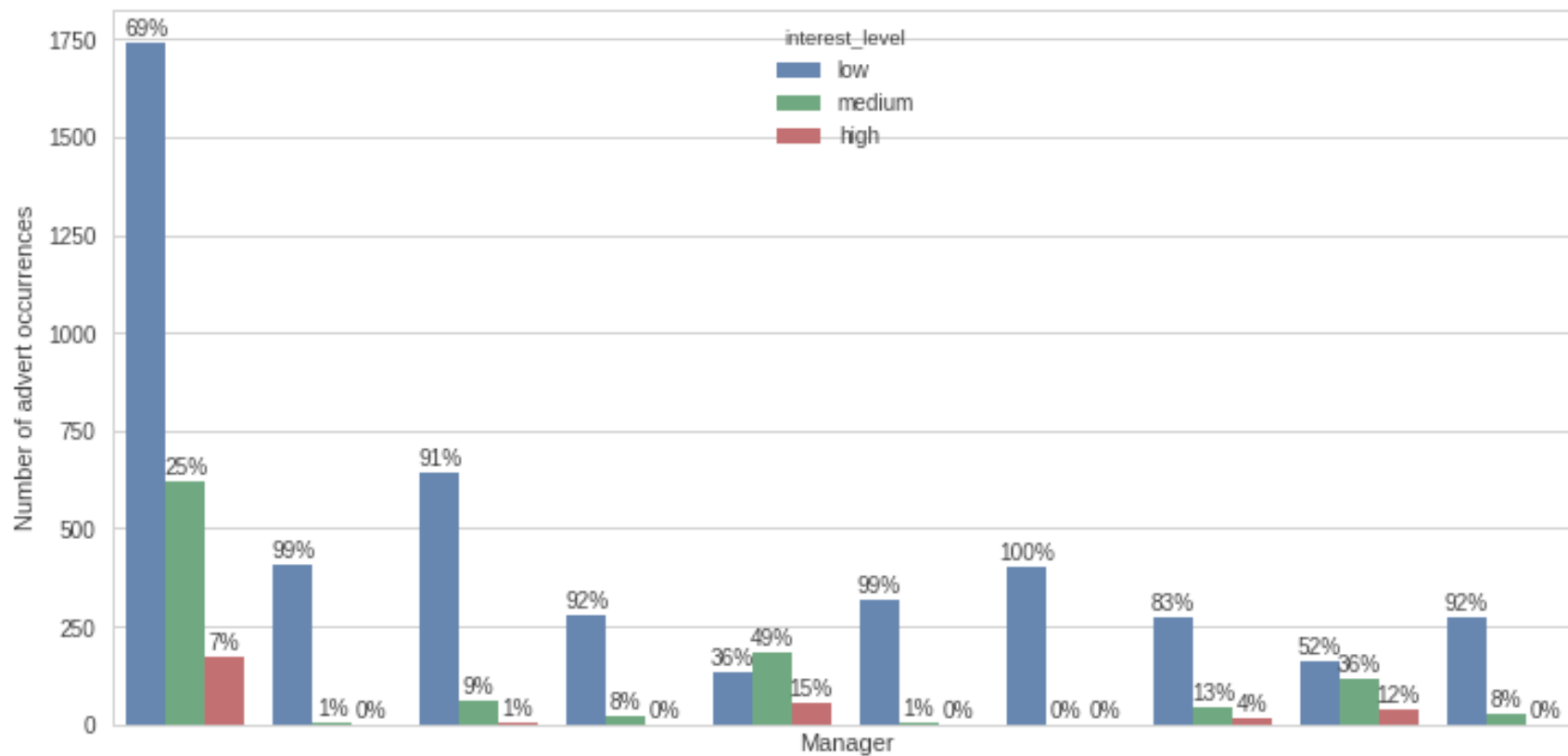
课程总结

Two Sigma可视化案例：



课程总结

Two Sigma可视化案例：



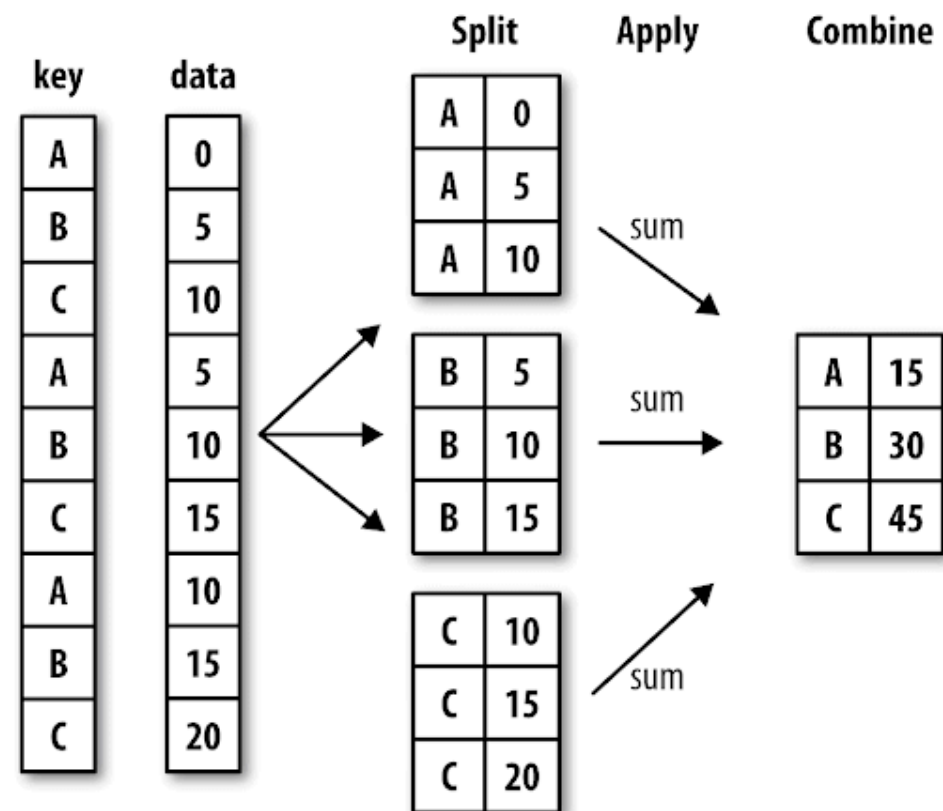
课程总结

要有对比（分组、聚合）的思路：

- ❑ 房子与本小区相比价格怎么样？
- ❑ 房子与同manager下价格相比怎么样？
- ❑ 房子与同等配置下价格相比怎么样？

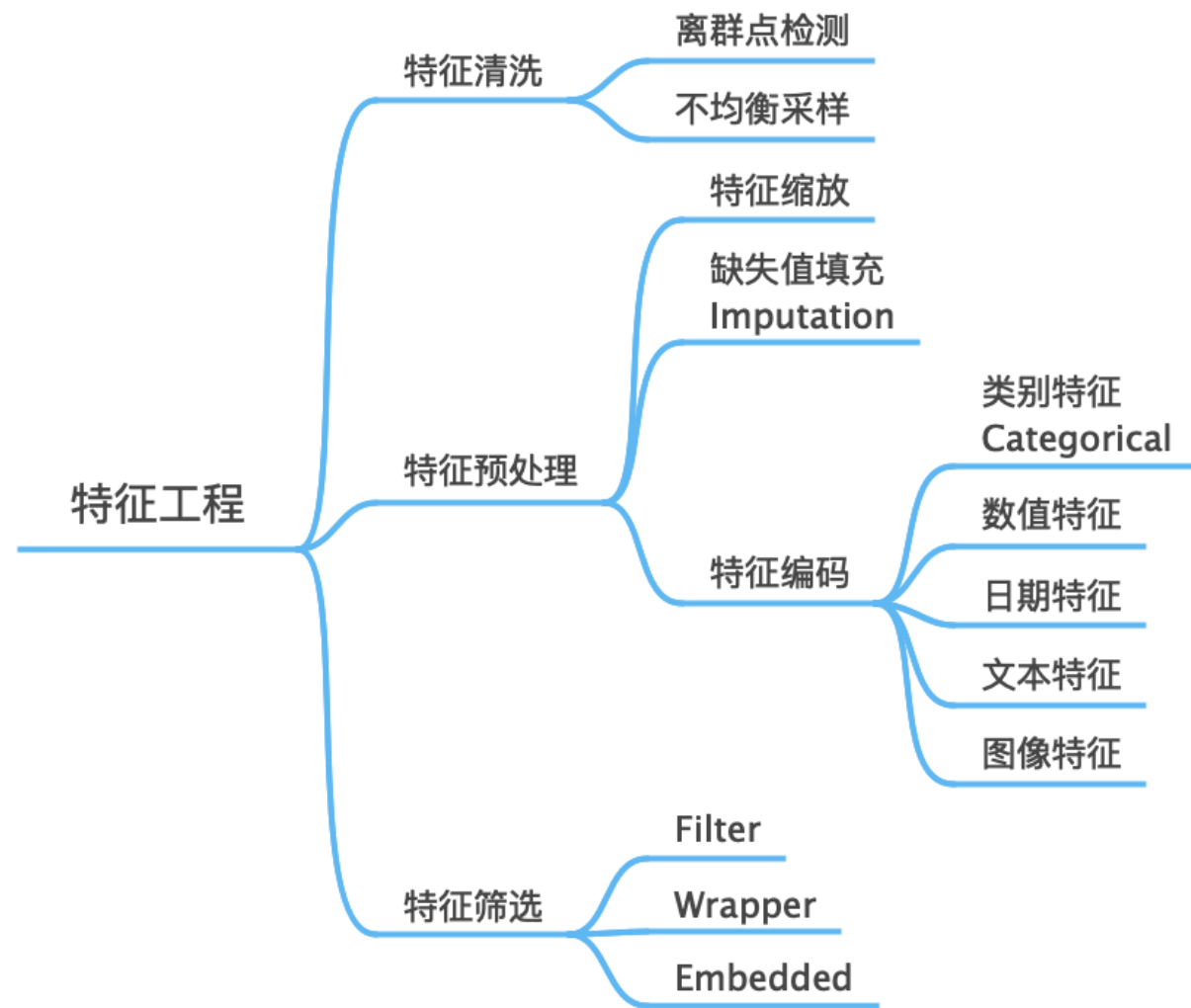
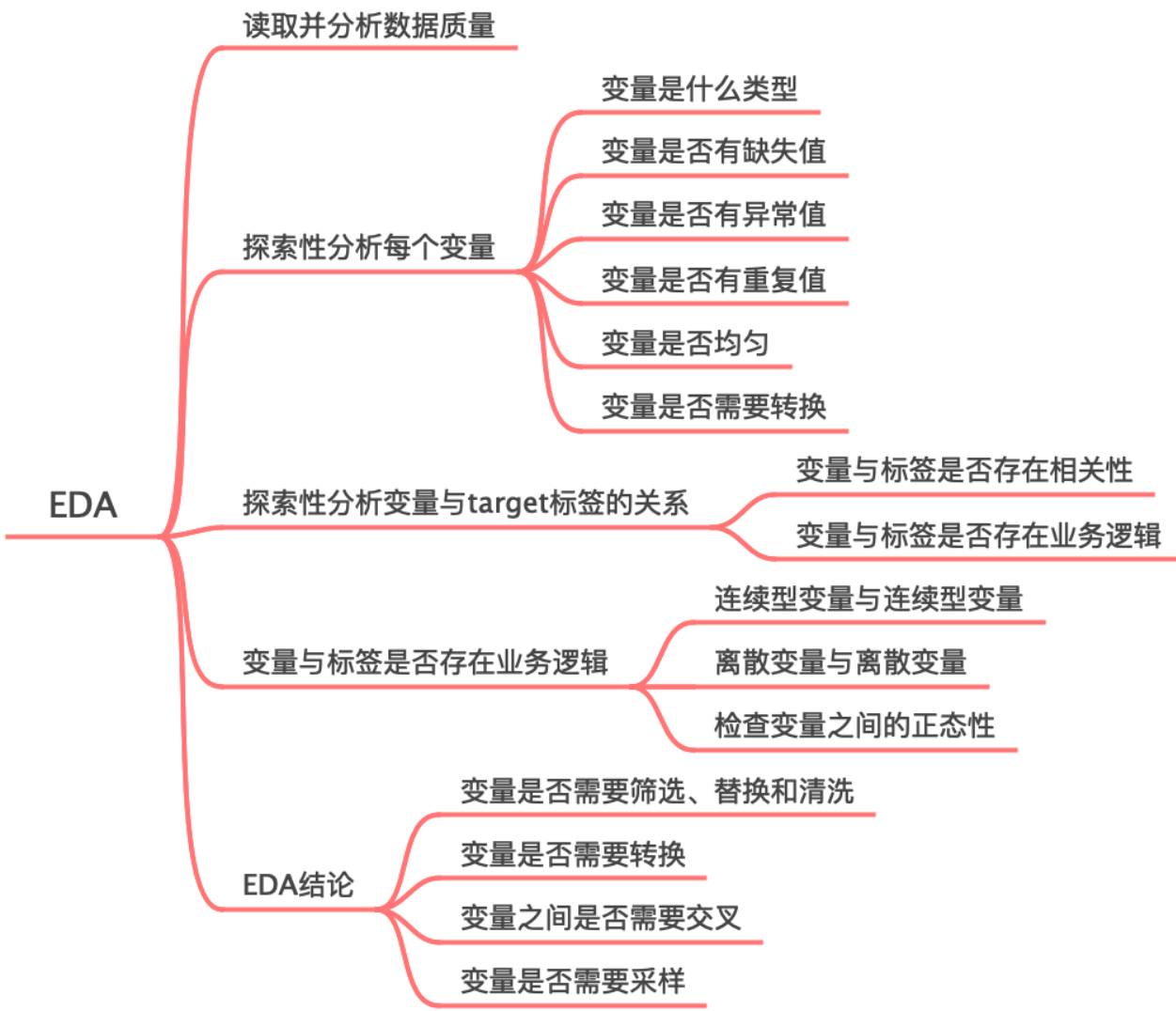
思考🤔：

- ❑ 特征如何编码，模型能快速学习；
- ❑ 模型哪些能自己学习，哪些学习不到；



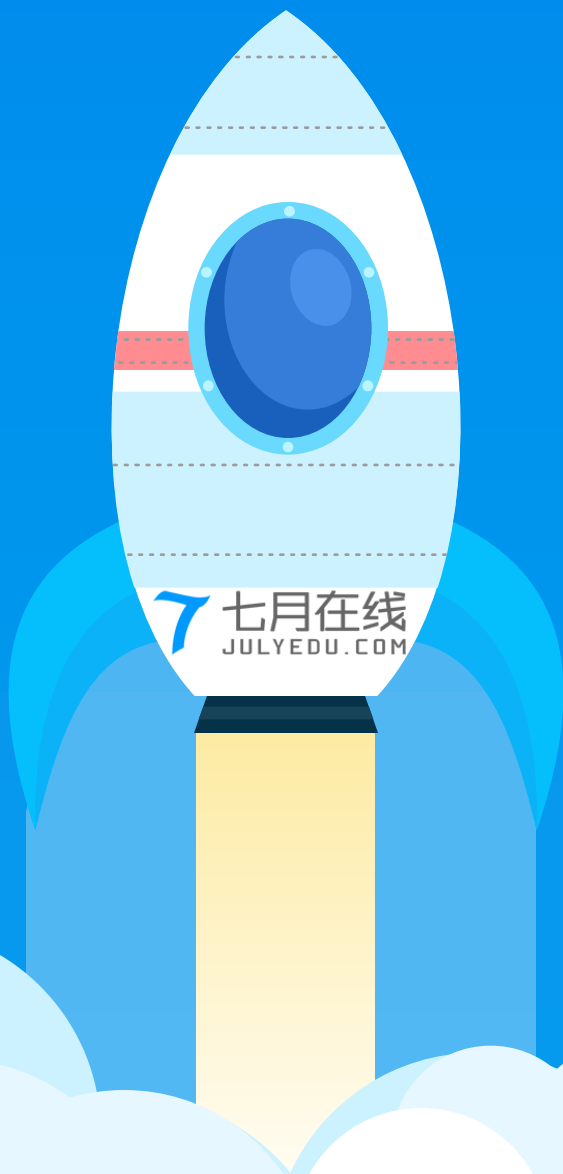
课程总结

Kaggle竞赛是学习知识的一种很有效的形式：





微信扫一扫关注我们



THANKS

刘老师

<https://www.julyedu.com>