# Module 3总结: Market Basket Analysis

**1. Lesson 1: What is in Your Basket?**

**目标：理解购物篮分析的商业价值 —— 用于 交叉销售 (Cross-Selling)、追加销售 (Up-Selling)、捆绑销售 (Bundling)。**

* **为什么重要？**
  + **保留老客户比获取新客户更省成本。**
  + **提高客户保留率能显著提升利润。**
  + **交叉/追加销售和口碑传播比广告更有效。**
* **实现方式：**
  + **Cross-Sell：推荐互补商品（例：买酱油 → 推荐蚝油）。**
  + **Up-Sell：推荐高端或更大规格的商品。**
  + **Bundle：把多个商品组合销售，提高总价值。**
* **数据来源：POS 系统、线上电商、电话/邮件订单。**
* **数据结构：**
  + **交易-商品表格（两列格式）**
  + **Item List（每笔交易一条记录，含商品数组）**
  + **Item Indicator（稀疏布尔矩阵，字段多，存储低效）**

**👉 关键：发现 哪些商品经常一起出现，从而指导营销策略。**

**2. Lesson 2: How Are Association Rules Discovered?**

**(a) 基本概念**

* **项集 (Itemset)：一组商品。**
* **频繁项集 (Frequent Itemset)：支持度 ≥ 最小支持度阈值的项集。**
* **关联规则 (Association Rule)：**

**A⇒B,A∩B=∅A \Rightarrow B, \quad A \cap B = \varnothingA⇒B,A∩B=∅**

**Antecedent = 条件商品集，Consequent = 结果商品集。**

**(b) 评价指标**

1. **支持度 (Support)：**

**Supp(A⇒B)=P(A∪B)\text{Supp}(A\Rightarrow B) = P(A\cup B)Supp(A⇒B)=P(A∪B)**

**→ 规则覆盖了多少交易。**

1. **置信度 (Confidence)：**

**Conf(A⇒B)=P(B∣A)\text{Conf}(A\Rightarrow B) = P(B\mid A)Conf(A⇒B)=P(B∣A)**

**→ 在买了 A 的情况下，买 B 的概率。**

1. **期望置信度 (Expected Confidence)：就是 P(B)P(B)P(B)，即 Consequent 的基准概率。**
2. **提升度 (Lift)：**

**Lift(A⇒B)=Conf(A⇒B)P(B)\text{Lift}(A\Rightarrow B)=\frac{\text{Conf}(A\Rightarrow B)}{P(B)}Lift(A⇒B)=P(B)Conf(A⇒B)​**

* + **Lift = 1：无实际关联；**
  + **Lift > 1：正向关联；**
  + **Lift < 1：负向关联。**

1. **Leverage, Zhang’s Metric：额外的衡量指标，用来增强对规则“可信度”的判断。**

**(c) Apriori 算法**

* **核心思想：“频繁项集的子集也必然是频繁的”。**
* **步骤：**
  1. **找出频繁 1-项集；**
  2. **扩展成 2-项集、3-项集……逐步检查支持度；**
  3. **生成并筛选出满足置信度要求的规则。**

**3. Lesson 3: What Can Association Rules Tell Me?**

**(a) 参数设置**

* **min\_support：过低 → 规则过多；过高 → 规则过少。**
* **min\_confidence：根据风险偏好选择。**

**(b) 实例：Imaginary Store 数据**

* **数据：995 个顾客，7037 行交易记录。**
* **结果：**
  + **最大单品支持度：Chips，0.702。**
  + **设最小支持度 = 10/995 ≈ 0.01，最小置信度 = 0.5 → 共发现 53,516 条规则。**
  + **最大 Lift ≈ 3.95（理论上最高可达 7.16）。**

**(c) 规则选择**

* **高支持 + 高置信度：最常见 & 可靠规则。**
* **高置信度 + 低支持：可信但罕见。**
* **低 Lift 或随机模式：无意义的“假规则”。**
* **可视化：绘制 Support vs. Confidence 散点图（点大小可表示 Lift），帮助挑选有效规则。**

**(d) 特殊规则分析**

* **固定 Consequent（例：推广 Cereal → 看哪些 Antecedents 会导致购买 Cereal）。**
* **固定 Antecedent（例：买了 Oranges → 看还能预测哪些商品）。**

**✅ 总结**

1. **市场篮分析的目标：识别购买行为模式，推动交叉销售、追加销售、捆绑销售。**
2. **关键指标：支持度（覆盖率）、置信度（条件概率）、提升度（相对独立性度量）。**
3. **算法：Apriori 通过逐步扩展频繁项集，大幅减少搜索空间。**
4. **实践经验：**
   * **参数阈值的选择决定结果多少与可信度。**
   * **可视化和 PPC（Posterior Predictive Check 类似思想）帮助筛选“有用规则”。**
5. **案例：Imaginary Store 数据展示了如何用 Python (mlxtend) 实际挖掘规则，并通过 Lift/支持度/置信度判断哪些规则有商业价值。**

# Module 4 总结：Partitioning, Segmenting, and Clustering

**Lesson 1: Partition Observations for Training Models**

**🎯 目标**

* **理解为什么要做数据分区（train/test）。**
* **掌握简单随机抽样 (SRS) 与分层抽样 (Stratified SRS)。**

**✅ 核心概念**

* **客观性 (Objectivity)：模型应在不同数据/环境/未来仍能复现效果。**
* **复现性 (Reproducibility)：同一数据同一方法是否可再现结果。**
* **可重复性 (Replicability)：新数据下是否得出相同结论。**

**📊 数据分区**

* **常见分法：训练集 70% + 测试集 30%。必要时加 Hold-Out 验证集。**
* **按时间/地理划分：如用 2020–2022 训练 → 测 2023 数据。**
* **Python 工具：sklearn.model\_selection.train\_test\_split。**

**🔑 抽样方法**

* **SRS：完全随机。可能造成标签分布偏差。**
* **Stratified SRS：按标签类别分层抽样，保持比例一致。适合分类问题。**

**Lesson 2: Create Segments of Observations for Business Reasons (RFM Analysis)**

**🎯 目标**

* **识别“最有价值客户”。**
* **用 RFM 分析 分层：Recency, Frequency, Monetary。**

**✅ RFM 定义**

* **Recency (R)：最近一次消费距参考日期的时间。**
* **Frequency (F)：消费次数。**
* **Monetary (M)：消费金额总和。**

**📊 计算流程**

1. **选参考日期：通常为交易历史最早日期的前一天。**
2. **算 Recency/Frequency/Monetary → 五分位 → 打分 1–5。**
3. **RFM Score：三位数 (如 555 = 最忠诚客户, 111 = 最低价值客户)。**

**💡 应用**

* **识别忠诚客户、潜在流失客户。**
* **针对不同 RFM 分组定制营销策略。**

**Lesson 3: Put Observations with Similar Feature Values in Clusters**

**🎯 目标**

* **学习聚类（Clustering）的思想与 K-Means/K-Modes。**
* **用数值特征对客户/商品分组。**

**✅ 聚类基础**

* **Cluster ≠ Clutter：聚类应形成有意义的组，而不是随意堆叠。**
* **聚类标签：只是标识符，不代表顺序/强弱。**

**📏 相似度度量**

* **常见距离：欧几里得 (连续特征)，其他类型需特殊定义 (如 K-Modes 适合分类数据)。**

**🔑 K-Means 算法**

1. **随机选初始质心。**
2. **分配点到最近质心。**
3. **更新质心。**
4. **重复直到收敛。**

**❓ 多少个聚类 (k)？**

* **Elbow Method：观察误差下降曲线拐点。**
* **Silhouette Index：衡量聚类紧密度与分离度。**
* **经验法则：先看 2–10 个，结合业务逻辑判断。**

**✅ 模块总览**

* **Lesson 1 (Partitioning)：确保模型评估客观 → SRS / Stratified SRS。**
* **Lesson 2 (Segmenting / RFM)：业务角度客户分层，优先维护高 RFM 客户。**
* **Lesson 3 (Clustering)：数据驱动客户分组，需决定聚类数 → Elbow & Silhouette。**

# Module 5 总结：Linear Regression Analysis

**Lesson 1: Linear Regression Model**

**🎯 目标**

* 理解多元线性回归模型、最小二乘法 (OLS)、BLUE 性质。
* 掌握 **Aliasing（别名/共线性问题）** 与 **SWEEP 算子**。

**✅ 核心要点**

* **回归模型**

y=Xβ+ε,ε∼N(0,σ2I)y = X\beta + \varepsilon, \quad \varepsilon \sim N(0,\sigma^2 I)y=Xβ+ε,ε∼N(0,σ2I)

* **OLS 估计**：  
  β^=(X⊤X)−1X⊤y\hat\beta = (X^\top X)^{-1}X^\top yβ^​=(X⊤X)−1X⊤y。
* **BLUE**：Best Linear Unbiased Estimator → 最优线性无偏估计量。
* **Aliasing（别名）**：当 X⊤XX^\top XX⊤X 不满秩（列线性相关），参数估计不唯一。
  + 解决办法：SWEEP 算子可检测和处理共线性。
  + 尽管参数估计不唯一，但**预测值唯一**。
* **案例（Chicago Taxi 20 次行程）**
  + 初步模型：

Trip Payment^=6.6456+0.0834⋅Minutes+2.1768⋅Miles\widehat{\text{Trip Payment}} = 6.6456 + 0.0834 \cdot \text{Minutes} + 2.1768 \cdot \text{Miles}Trip Payment​=6.6456+0.0834⋅Minutes+2.1768⋅Miles

* + 截距代表起步价（Flag Pull）。
  + Miles 影响更大，Minutes 在统计上可能不显著。

**Lesson 2: Forward Selection**

**🎯 目标**

* 理解嵌套模型 (Nested Models) 与 F 检验。
* 学习逐步选择 (Forward Selection) 构建模型。

**✅ 核心要点**

* **嵌套模型**：通过增加变量形成的一系列模型。
* **F 检验**：比较两个嵌套模型，看新增变量是否显著提升拟合。
* **Forward Selection 步骤**：
  1. 从空模型开始；
  2. 每次尝试加入一个变量，用 F 检验评估；
  3. 显著则保留，否则舍弃。
* **分类变量处理**：
  1. 例子：Payment\_Method (Cash / Credit Card) → 转换成两个 Dummy 变量。
  2. Dummy 变量需 **成组进入或移除**。
* **结果**：
  1. 模型解释：Miles 每英里 ≈ $2.18；
  2. Payment\_Method: 信用卡支付平均多 $7.91（可能解释为小费）。

**Lesson 3: Feature Importance**

**🎯 目标**

* 用 **Shapley Value（夏普利值）** 解释各特征的重要性。
* 连接 **博弈论 (Game Theory)** 与回归特征贡献。

**✅ 核心要点**

* **R² (Coefficient of Determination)**：表示模型解释的变异比例。
* **夏普利值**：
  + 将特征看作玩家，模型解释的变异 (R²) 是总“收益”。
  + 每个特征的贡献 = 在所有进入顺序下的平均边际贡献。
  + 即便特征因共线性未入模，Shapley Value 也能度量其潜在重要性。
* **案例（Taxi Trip）**
  + R² = 0.9794，模型解释了 ~98% 的支付变异。
  + 分解贡献：
    - Trip\_Miles → 64.15%
    - Payment\_Method → 29.02%
    - Trip\_Minutes → 4.77%

👉 Shapley Value 提供了**绝对特征重要性**，比 Forward Selection 更全面。

**✅ 模块要点总结**

1. **OLS & BLUE**：回归系数估计最优无偏。
2. **Aliasing**：多重共线性问题 → 用 SWEEP 算子检测和修正。
3. **Forward Selection**：嵌套模型 + F 检验 → 自动选变量。
4. **Shapley Value**：公平分配 R²，给出每个特征的解释贡献。
5. **Taxi 案例结论**：Miles 最关键，Payment\_Method 次之，Minutes 影响最小。

# Module 6 总结：Binary Logistic Regression

**Lesson 1: Logistic Regression 基本概念**

**🎯 目标**

* 理解 **Odds（优势比）** 与 **Probability（概率）** 的关系。
* 学习 Logistic 回归公式、参数估计方法。
* 了解 **完全分离 (Complete Separation)** 问题。

**✅ 核心要点**

A white background with black text

AI-generated content may be incorrect.

→ Logistic 回归

建模的是 **log-odds 与自变量的线性关系**。

* **估计方法**:
  + 使用 **极大似然估计 (MLE)**；
  + 常用 **Newton-Raphson 迭代法**，收敛速度约每次提升两位有效数字。
* **分离现象 (Separation)**:
  + **完全分离**：自变量完美预测响应变量 (p=0 或 p=1)；
  + **准完全分离**：大多数观测被完美预测；
  + 后果：参数估计无穷大、标准误巨大；预测概率看似“完美”但不可信。
  + **解决方案**：
    - 合并类别（减少稀疏）；
    - 连续变量预分箱；
    - 降维减少特征。

**Lesson 2: Forward Selection in Logistic Regression**

**🎯 目标**

* 使用 **嵌套模型 (Nested Models)** 和 **Deviance Test** 进行变量选择。

**✅ 核心要点**

* **嵌套模型**：小模型 ⊂ 大模型（增加一个或多个自变量）。
* **Deviance Test**:

A black text on a white background

AI-generated content may be incorrect.

* **Forward Selection 步骤**：
  1. 从截距项模型开始；
  2. 每次尝试加入一个变量，用 Deviance Test 检验；
  3. p<0.05 则保留，否则丢弃；
  4. 直到没有变量能再进入。
* **优缺点**：
  1. 优点：快，模型简洁；
  2. 缺点：可能错过“非嵌套的最佳模型”，且阈值选择具有主观性。
* **案例：College Student Retention**
  1. 响应变量：是否 Dropout (Yes/No)。
  2. 特征：Attendance, Debtor, Scholarship holder, Age at enrollment。
  3. Entry threshold = 0.05。
  4. 结论：部分特征（如 Debtor\_Yes, Scholarship holder\_Yes）因 **别名 (Aliasing)** 被省略。

**Lesson 3: Interpret Model & Assess Performance**

**🎯 目标**

* 正确解释 Logistic 回归参数。
* 使用不同的 **Pseudo R² 指标** 评估模型。

**✅ 核心要点**

* **解释参数 (Odds Ratio)**
  + Logistic 回归系数的指数形式代表 **Odds Ratio**。
  + 例如：
    - 若 OR=1.28 → 在控制其他条件下，某场景比另一场景 dropout 的 odds 高 28%。
    - OR>1 → 增加风险；OR<1 → 降低风险。
* **别名参数 (Aliasing)**
  + 当变量完全共线时，部分参数无法唯一估计 → 设置为 0。
  + 不影响预测概率。
* **Pseudo R² 指标**（因 Logistic 没有传统 R²）：
  + **McFadden R²**
  + **Cox & Snell R²**
  + **Nagelkerke R²**
  + **Tjur R²**（Discrimination Coefficient, 解释度较直观）。
* **模型评估总结**
  + 模型有较好的区分能力；
  + 但还有改进空间 → 可考虑更多特征或交互项。

**✅ 模块总结**

1. **Lesson 1**：Logistic 回归通过建模 log-odds 解决概率范围 (0–1) 限制；MLE 估计；需警惕分离问题。
2. **Lesson 2**：Forward Selection 使用 Deviance Test 构建简洁模型，但可能错过最佳非嵌套解。
3. **Lesson 3**：参数解释依靠 **Odds Ratio**；模型优度用 Pseudo R²；别名参数不影响预测结果。

# Module 7 总结：Decision Trees (CART & Cluster Profiling)

**Lesson 1 – Motivation of Decision Trees**

1. **Incremental Model of Decision-Making**
   * 决策树把复杂决策拆分为一系列简单步骤。
   * 每一步仅基于一个特征进行二元选择。
   * 目标是最小化每一步的代价（errors, misclassification, squared error）。
2. **例子：Spoiled Milk Decision**
   * 判断牛奶是否还能用：通过连续规则（如颜色、气味、时间）最终决定“丢弃”或“烘焙使用”。
3. **Data-driven & Replicable**
   * 不依赖直觉或试错，而是基于数据分布，寻找能最好区分类别的切分点。
4. **Recursive Partitioning**
   * 通过不断寻找最佳切分点（降低方差或不纯度），把数据划分为纯净的子集。
   * 当子节点方差=0 或 不纯度=0 时，分裂终止。

**Lesson 2 – The CART Algorithm**

1. **Terminology**
   * **Node 节点**: 树中的一个方框。
   * **Branch / Split 分支**: 分割条件。
   * **Root Node 根节点**: 顶部起点，无输入分支。
   * **Leaf Node 叶节点**: 无输出分支，给出预测结果。
   * **Depth 深度**: 层数 - 1。
2. **CART 定义**
   * **Classification Tree** → 目标变量是类别型。
   * **Regression Tree** → 目标变量是连续型。
   * CART 对缺失值、离群点鲁棒；对单调变换（对数、平方、平方根）不敏感。
3. **CART 算法步骤**
   * 在父节点内循环所有特征；
   * 枚举所有可能二分切割；
   * 计算不纯度（Impurity Metric，如 Gini, Entropy, SSE）；
   * 选择最优切割点（最小不纯度）；
   * 递归生成子节点，直到停止条件。
4. **Impurity Metrics**

A group of black text

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

**Lesson 3 – Cluster Profiling**

1. **动机**
   * 先用 **Clustering**（如 k-Means）找到聚类；
   * 再用 **Decision Tree** 描述聚类特征，解释哪些特征最能区分簇。
   * 目标：高分类准确率、纯净叶节点。
2. **Scikit-Learn 工具**
   * DecisionTreeClassifier()：分类树
   * DecisionTreeRegressor()：回归树
   * plot\_tree()：可视化树结构
3. **Bike Sharing 示例**
   * 数据：气温 temp、湿度 humidity、风速 windspeed。
   * k=2 簇：
     + Cluster 0: 湿度 > 63.5 且风速 ≤ 27 → 夏日湿润微风 (Tourist Season)。
     + Cluster 1: 湿度 ≤ 59.5 → 春秋干燥舒适天气 (School/Work Days)。
   * 分类树准确率 = 99.63%。

| **特性** | **分类树 (Classification Tree)** | **回归树 (Regression Tree)** | **聚类剖析 (Cluster Profiling)** |
| --- | --- | --- | --- |
| **目标变量** | 类别型 (Nominal/Ordinal) | 连续型 (Interval) | 聚类ID (Nominal) |
| **切分准则** | 熵 (Entropy), Gini 指数 | 方差减少 (SSE) | 熵 / Gini |
| **输出** | 类别预测 | 数值预测 | 聚类特征描述 |
| **应用场景** | 信用风险、是否购买 | 房价预测、销量预测 | 解释聚类 → 给簇贴标签 |
| **优点** | 可解释性强，非线性 | 不依赖分布假设 | 把“黑箱”聚类结果转为业务可理解的规则 |

# Module 8 总结：Evaluating Model Performance

**Lesson 1 – Prediction Models**

**1. 残差 (Residuals)**

* ei=yi−y^ie\_i = y\_i - \hat{y}\_iei​=yi​−y^​i​
* 正残差：模型 **低估** (underestimates)
* 负残差：模型 **高估** (overestimates)

**2. 评估指标**

1. **RASE (Root Average Squared Error)**
   * 类似 RMSE
   * 衡量预测值与真实值的偏差
   * 值越小越好
2. **Relative Error**
   * 
   * = 1 → 模型等同于无信息模型
   * < 1 → 模型有解释力
   * 1 → 模型比无信息模型还差
   * 在线性回归中：Relative Error = 1 - R^2
3. **MAPE (Mean Absolute Proportion Error)**
   * 平均相对误差比例
   * 衡量预测值与实际值差距的相对大小

**案例**

* 芝加哥出租车支付预测模型：
  + RASE = $1.19
  + Relative Error ≈ 2.5%
  + MAPE ≈ 1.9%  
    → 模型拟合效果好。

**Lesson 2 – Nominal Classification Models**

**1. 评估指标**

1. **Misclassification Rate (错误分类率)**
   * FP+FN/总数v​
2. **RASE for classification**
   * 根据 one-hot 编码的概率预测与实际标签的差异计算
   * 最小=0，最大取决于类别数

**直觉**

* 错误分类率关注是否分对类。
* RASE 关注概率预测的“可信度”。

**Lesson 3 – Binary Classification Models**

**1. Confusion Matrix**

* TP, FP, TN, FN
* 指标：
  + **TPR / Sensitivity / Recall** = TP / (TP+FN)
  + **TNR / Specificity** = TN / (TN+FP)
  + **误分类率** = (FP+FN)/N

**2. AUC (Area Under Curve)**

* 通过 **ROC 曲线** 计算：
  + 横轴：FPR = 1 – Specificity
  + 纵轴：TPR = Sensitivity
* AUC = 随机选择一个 Event 概率 > Non-event 概率的概率
* AUC ∈ [0,1]，0.5=随机，1=完美

**3. KS 统计量 (Kolmogorov-Smirnov)**

* 最大化TPR−FPR
* 找到最佳阈值

**4. Precision & Recall**

* Precision = TP/(TP+FP)
* Recall = TP/(TP+FN)
* **F1 Score** = 调和平均 = 2/(1/P+1/R)

**5. Lift & Gain Charts**

* 将客户按预测概率排序 → 分为 decile (十分位)
* **Gain**: 每个分组实际捕捉的 Event 比例
* **Lift**: Gain% / 总体 Event 比例
* 应用：营销资源分配