**付费提升模型开发步骤**

1. **数据清洗**
2. 数据源接入
   1. 订购表，一个月，（用户id，产品包名）
   2. 收视表，一周~一个月，（用户id，节目名）

注意两张表的数据日期保持一致

尽量取最新的数据，保证电话回访的时效性

1. 数据过滤
   1. 过滤掉只被一个用户看过的节目，即从收视行为表中剔除包含该节目的行为记录
   2. 过滤掉只看过一个节目的用户，即从收视行为表中剔除包含该用户的行为记录。如果订购率与活跃度正相关，可以考虑进一步过滤，以减少数据量，例如过滤掉只看过两个节目的用户。
2. 数据抽样（可选）

当数据量过大导致一次训练时间过长时（超过5分钟），需要进行数据抽样，以便进行模型调参。

设置抽样条件，例如只保留观看节目数>10的用户，以减少用户数量，生成小数据集，使单次训练时间缩短到可接受的时间范围内。

小数据集只用于模型调参，正式的模型预测需要使用未抽样的数据。

1. 自增id生成

Spark ALS模型只接受长整型的id，所以要将字符串型的id转换成自增id

* 1. 用户自增id生成

将订购表和收视表的用户id合并后去重，建立用户自增id，并保存用户id与用户自增id的映射关系。

* 1. 物品自增id生成

产品包和节目统称为物品。

分别建立产品包自增id和节目自增id，然后在此基础上建立物品自增id，即前半部分为产品包自增id，后半部分为节目自增id+固定数值（产品包数量），并保存产品包名、节目名与物品自增id的映射关系。

1. **模型离线评测与调参**
   1. 数据集分割
      1. 将经过过滤后的产品订购记录（用户id，产品包名）按照8：2的比例随机分割为训练集和验证集。如果验证集的记录数超过2万，则验证集只需保留2万，其余都为训练集。
      2. 收视行为记录（用户id，节目名）无需分割，全部为训练集。
      3. 遍历检查验证集中是否存在训练集中没有的用户id或产品包名，如果存在，则把该条记录从验证集移回至训练集中。
   2. 模型训练
      1. 查询之前保存的映射关系表，将训练集的所有记录转换成ALS模型所需的Rating类型的数据，即Rating（用户自增id，物品自增id，1.0）。
      2. 设定ALS模型的超参数，rank（60~200），lambda（0.01，0.1，1，10），alpha（0.01，0.1，1，10），进行ALS隐式反馈模型训练。
   3. 模型效果评测
      1. 统计出验证集的用户列表，生成并保存每个用户所订购的产品包列表，格式为：

验证集：[(用户56, [产品包3, 产品包6, ...]), (用户873, [产品包9, 产品包13, ...]), ...]

生成并保存验证集用户在训练集中对应的物品（产品包+节目）列表，格式为：

训练集：[(用户56, [物品23, 物品94, ...]), (用户873, [物品31, 物品633, ...]), ...]

* + 1. 模型训练完毕后，对验证集中的所有用户进行物品推荐（recommendProducts），过滤掉所有包含节目自增id的推荐结果，只保留包含产品包自增id的推荐结果，再过滤掉训练集中对应用户的物品自增id的推荐结果，按照预测评分从大到小排序，生成验证集用户的Top10推荐列表，格式为：

预测集：[(用户56, [产品包7, 产品包6, ...]), (用户873, [产品包21, 产品包87, ...]), ...]

* + 1. 对验证集和预测集中的每一个用户进行比对，计算每个用户各自的f1-score，最后求所有用户的平均f1-score。

f1-score = 2 \* 命中数 / (预测列表长度 + 验证列表长度)

* 1. 模型调参

记录下该超参数组合（rank，lambda，alpha）下的f1-score，返回模型训练阶段进行超参数调整，直到找到使f1-score最大的超参数组合。

* 1. 模型效果对比（可选）

实现热门推荐模型，即对所有验证集中的所有用户都推荐相同的最热门产品包列表（去除训练集中已经有的产品包以后，得出的Top10的产品包列表）

预测集：[(用户56, [产品包1, 产品包2, ...]), (用户873, [产品包1, 产品包2, ...]), ...]

计算出热门推荐模型的f1-score，作为基准，用于检验ALS模型的效果。如果f1-score比热门推荐模型的要低，说明该模型没有实际价值。

1. **模型预测与结果输出**
   1. 使用调参后得到的最佳超参数组合，对完整数据集进行模型训练，并保存训练完成的模型。
   2. 针对指定的产品包进行用户推荐（recommendUsers），按照评分从大到小排序。
   3. 如果订购率与用户活跃度正相关，则可以考虑设定一个活跃度阈值（至少看过x个节目）对用户进行筛选。
   4. 根据实际需求截取前N名用户，输出最终的用户清单。