推荐算法的思想就是找到和你相似的人，也就是口味相同的人，把他喜欢的物品或者电影歌曲推荐给你；或者是将你买过的或者喜欢的物品，电影或者歌曲推荐给你；还有就是可能将你搜索过的物品、电影或者歌曲推荐给你；还有可能就是综合上述的算法做推荐。

协同过滤算法的流程：

1. 手机用户的喜好

大数据的收集处理，这些数据其实反映了用户的某些偏好，通过分析用户的行为评分，基于用户所有的数据来做出推荐。

* 将不同的行为分组：
* 商城商品
* 用户购买产品
* 用户重复浏览某类性的产品
* 用户收藏的产品
* 将用户的行为进行加权操作
* 商城商品
* 用户购买产品+高权重
* 用户重复浏览某类性的产品+高权重
* 用户收藏的产品+低权重

对数据进行一些预处理，主要的操作就是减噪和归一化。减噪的目的就是让数据更加纯净，在后续的数据分析更加精确；因为获取的数据可能来源于不同的系统，所以数据的样式参差不齐，加权操作就没办法进行，需要把数据统一标准，将数据归一化。

1. 查找相似的用户或者商品

* 协同过滤(collaborative filtering)算法主要有两个分支，
  + - 基于用户的协同过滤(user-based collaborative filtering)
    - 基于商品的协同过滤(item-based collaborative filtering)

1. 欧式距离：欧几里得距离（Euclidean Distance）
2. 余弦相似：一般在文本处理中，可以将语料分词向量化之后进行两个向量的余弦计算，来判断两个文本的相似程度，当两个向量正交的时候，余弦值为0，表示两个文本的相似度低，余弦值越接近1表示两个文本的相似度越高，反之余弦值为越接近-1，表示两个文本越负相关。
3. 皮尔逊相关系数（Pearson Correlation Coefficient）：皮尔逊相关系数是在机器学习中很常用的一个相关性指标。皮尔逊相关系数其实就是协方差除以标准差得到的，取值范围是[-1,1]，同理，皮尔逊相关系数越接近1就表示相关度越高，越接近-1，表示越负相关，越接近0，就表示相关度越低。
4. 计算推荐

协同过滤分成两种过滤方法，基于用户的和基于商品。

* 基于用户的协同过滤推荐方法

于用户的推荐大体意思就是找到一些相似性很高的用户，那么别人喜欢的商品也很有可能是你喜欢的商品，以此来作为依据推荐。

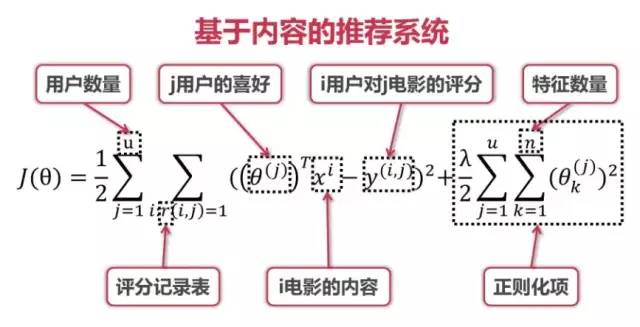
* 基于商品的推荐

类似于基于用户的协同过滤，这里就是寻找商品之间的关系，商品之间的相似度，关联度，以此来对用户做推荐

1. 该推荐系统的说明

数据集中，根据用户对电影的评分，来判断用户对电影的喜好程度，然后根据电影信息的.csv文件获取电影信息。

1. 将用户评分表和电影信息表合并，留下userId，movied，rating，timestamp，MovieRow，title字段信息
2. 创建电影评分矩阵rating和评分记录矩阵record
3. 添加一个标签位flag，，然后获取合并之后列表的列数，然后遍历rating矩阵每一个位置，将电影的评分填入矩阵中
4. 构建模型



# 函数解释：# reduce\_sum() 就是求和reduce\_sum( input\_tensor, axis=None,  keep\_dims=False, name=None, reduction\_indices=None)

# reduce\_sum() 参数解释：

# 1) input\_tensor：输入的张量。

# 2) axis：沿着哪个维度求和。对于二维的input\_tensor张量，0表示按列求和，1表示按行求和，[0, 1]表示先按列求和再按行求和。

# 3) keep\_dims：默认值为Flase，表示默认要降维。若设为True，则不降维。

# 4) name：名字。

# 5) reduction\_indices：默认值是None，即把input\_tensor降到 0维，也就是一个数。对于2维input\_tensor，reduction\_indices=0时，按列；reduction\_indices=1时，按行。

# 6) 注意，reduction\_indices与axis不能同时设置。

# tf.matmul（a,b）,将矩阵 a 乘以矩阵 b，生成a \* b

# tf.matmul（a,b）参数解释：

# 1) a：类型为 float16，float32，float64，int32，complex64，complex128 和 rank > 1的张量。

# 2) b：与 a 具有相同类型和 rank。

# 3) transpose\_a：如果 True，a 在乘法之前转置。

# 4) transpose\_b：如果 True，b 在乘法之前转置。

# 5) adjoint\_a：如果 True，a 在乘法之前共轭和转置。

# 6) adjoint\_b：如果 True，b 在乘法之前共轭和转置。

# 7) a\_is\_sparse：如果 True，a 被视为稀疏矩阵。

# 8) b\_is\_sparse：如果 True，b 被视为稀疏矩阵。

# 9) name：操作名称（可选）