

## 引子

前面探讨了Zara快消的能力是基于算法支撑的产业链的优势。今天推演以下Shein对**个性化推荐算法**的要求，按业态要求甚至更高，毕竟online模式更需要对信息的智能化算法的指导。而要保持保持快时尚行业的技术优势,必须建立在对人才、市场与变化的高度重视之上。Shein需要用数据与算法不断深化对用户与产品的理解,需要让数字化手段成为推动企业持续进化的内在动力。

需要在算法开发与**业务运营组织和责任人**之间持续学习与提高并**落实主体责任**,需要培养敏锐的数据洞察力与变革思维,需要让数字化研发与业务之间的协作成为常态。

该**个性化推荐算法**通过矩阵分解获得用户商品潜在特征,并基于特征匹配度计算个性化TOP推荐商品,最后返回推荐商品详情,为用户实现个性化商品推荐。

但请记住，个性化推荐算法只是工具,洞察与执行力才是目的

只有企业坚持数字化平台能力和算法平台能力的战略投入的定力，持续推动学习型创新组织,企业才有可能真正掌握并运用新技术带来的变革机会,让数据与算法发挥其应有的巨大价值,成为推动企业获得持续竞争优势的重要动能。

## 前言

基于数据的算法平台有很多，对比Zara的产业链算法优势，Shein在个性化商品推荐算法是牵引商品，供应链，数字化部门不断重塑的牛鼻子，推动基于消费者导向的洞察持续带动后续组织和流程的持续优化并执行。下面针对个性化算法可能在类似Shein类型的场景下如何实现进行一些推演，不代表Shein事实上按照这些算法结构和代码进行实现。

以下case均以Heout为案例名称进行推演，来看看算法如何为每个用户生成个性化的商品推荐列表,从而达到提高用户体验与转化率的目标。

第一部分：Heout的个性化推荐算法主要包含以下功能及流程:

1. 收集用户浏览及行为数据,构建用户-商品交互矩阵。通过获取用户浏览商品、收藏、购买等行为,构建用户-商品交互矩阵,反映用户对商品的兴趣程度。
2. 矩阵分解获得用户latent特征与商品latent特征。使用SVD等矩阵分解算法对用户-商品矩阵进行分解,得到用户与商品的潜在特征,用于计算兴趣匹配度。
3. 根据特征匹配度计算个性化TOP推荐列表。基于用户latent特征与所有商品latent特征的余弦相似度,选取与用户特征最为匹配的Top N商品,构成推荐列表。
4. 查询推荐商品详情,构造个性化推荐结果。根据推荐列表中的商品ID,查询商品图片、名称、价格等信息,组合为个性化推荐商品详情。
5. 返回个性化推荐结果。将TOP推荐商品列表及相关商品详情返回给用户,实现个性化商品推荐。

Heout可以采用以下算法实现个性化商品推荐:

```
python
```

```
# 1收集用户浏览记录与行为数据
```

```
user_logs = collect_user_logs()
```

```
# 构建用户-商品交互矩阵
```

```
user_item_matrix = []
```

```
for user_id, item_id, behavior in user_logs:
```

```
user_item_matrix.append([user_id, item_id, behavior])
```

```
# 2计算用户-商品矩阵的协同过滤预测
```

```
from surprise import Reader, Dataset, SVD, accuracy
```

```
reader = Reader(rating_scale=(1, 5))
```

```
data = Dataset.load_from_df(user_item_matrix, reader)
```

```
svd = SVD(n_factors=50, random_state=0)
```

```
svd.fit(data)
```

```
predictions = svd.test(data.build_testset())
```

```
accuracy.rmse(predictions)
```

```
# 3获取用户的TOP推荐列表
```

```
user_id = 'U1234'
```

```
items = svd.recommend(user_id)
```

```
top_items = items[0:10]
```

```
# 4构造个性化推荐详情
```

```
product_infos = get_product_informations()
```

```
recommends = []
```

```
for item in top_items:
```

```
    product_info = product_infos[item[1]]
```

```
    recommends.append({
```

```
'product_id': item[1],  
'image': product_info['image_url'],  
'name': product_info['name'],  
'price': product_info['price']  
})
```

```
# 5返回个性化推荐结果
```

```
return recommends
```

Heout可根据该算法为每个用户生成个性化的商品推荐列表,大大提高用户体验与转化率。但Heout在实施个性化推荐时也需考虑以下因素：

1. 算法对新用户或冷启动用户的适用性,避免推荐准确度过低;
2. 推荐商品的多样性与新颖性,避免过于保守或则相同;
3. 用户反馈与转化数据的持续优化,动态调整推荐算法;
4. 不同算法之间的比较与选择,确保最优推荐效果;
5. 研发团队与业务团队的密切协作,让推荐结果真正发挥作用。

## 第二部分：五个阶段的算法实现进行模拟推演

```
# 1收集用户浏览记录与行为数据
```

```
# 2计算用户-商品矩阵的协同过滤预测
```

```
# 3获取用户的TOP推荐列表
```

# 4构造个性化推荐详情

# 5返回个性化推荐结果

### 第一步 收集用户浏览记录与行为数据

用户浏览记录与行为数据是构建个性化推荐系统的重要基础,它包含用户在Heout平台的各种操作与交互信息,为算法模型提供重要的输入与监督。

Heout可以收集如下用户浏览记录与行为数据:

用户ID	时间	页面	行为
U1001	2020-01-01 12:34:56	商品详情页	浏览
U1001	2020-01-01 12:40:23	购物车页面	加入购物车
U1001	2020-01-02 09:25:17	订单确认页	下单
U1001	2020-01-03 15:38:09	商品列表页	收藏商品
U1002	2020-02-05 10:20:13	个人中心页	浏览而已购订单
U1002	2020-02-07 19:35:42	APP推送	点击推送消息
U1003	2020-03-12 16:44:23	品牌页面	浏览品牌商品
...	...	...	...

Heout收集的用户数据主要包含:

1. 用户ID:Heout的注册用户ID。

2. 时间:用户进行操作或交互的精确时间点。
3. 页面:用户浏览或停留的页面类型,如商品详情页、购物车页面等。
4. 行为:用户在页面上的操作行为,如浏览、收藏、购买、下单等。

这些数据为Heout提供了重要的算法模型输入,Heout可以基于这些数据:

1. 分析用户的兴趣与偏好,定制个性化推荐;
2. 发现用户的购买路径与购买留存,优化用户体验;
3. 分析不同推广活动的曝光与转化效果,优化营销策略;
4. 发现用户在不同页面的停留时间与转化率,优化网页或APP接口;
5. 持续优化算法模型,不断提高推荐质量和转化效率。

但在数据收集与应用过程中,Heout也需考虑:

1. 用户数据的准确性、完整性与代表性;
2. 用户隐私与数据安全问题,避免数据泄露或滥用;
3. 数据采集与算法模型优化的协同频率,保证推荐实效性;
4. 研发团队与业务团队的密切配合,让数据发挥最大价值;
5. 数据收集与应用策略的持续优化,避免过度依赖任何单一数据源。

## **第二步：**计算用户-商品矩阵的协同过滤预测

协同过滤是一种基于用户的兴趣相似性进行推荐的方法。Heout可以采用协同过滤算法对用户-商品矩阵进行预测,获得用户的个性化推荐列表。

该算法首先获取用户-商品交互矩阵,然后采用train-test split进行评估,SVD模型在不同参数下进行训练与测试,选择最佳参数。最后使用最佳SVD模型对新用户进行协同过滤推荐,并返回推荐结果。

协同过滤是主流的推荐算法,通过分解用户-商品矩阵预测未知交互来实现个性化推荐。选取合适的矩阵分解算法与最佳参数对结果有很大影响,需要训练验证取得最佳模型。而对新用户或商品的推荐则需要平衡模型预测与默认配置,这也是协同过滤的常见难点。

算法的主要流程为:

1. 构建用户-商品交互矩阵,包含用户ID、商品ID和交互行为;
2. 使用SVD算法对用户-商品矩阵进行分解,获得用户和商品的latent特征;
3. 基于分解结果对用户未交互的商品进行推荐预测;
4. 根据预测结果为用户生成TOP推荐列表。

Heout可以使用Surprise库中的SVD算法执行如下操作:

```
python

# 构建用户-商品交互矩阵

user_item_matrix = []

for user_id, item_id, behavior in user_logs:

    user_item_matrix.append([user_id, item_id, behavior])

# 使用SVD算法进行矩阵分解

from surprise import Reader, Dataset, SVD, accuracy

reader = Reader(rating_scale=(1, 5))
```

```
data = Dataset.load_from_df(user_item_matrix, reader)

svd = SVD(n_factors=50, random_state=0)

svd.fit(data)

# 计算推荐预测

predictions = svd.test(data.build_testset())

accuracy.rmse(predictions)

# 获取用户的TOP推荐列表

user_id = 'U1234'

items = svd.recommend(user_id)

top_items = items[0:10] # TOP 10推荐
```

该算法可以为Heout带来以下好处:

1. 个性化推荐商品,提高用户转化率;
2. 发现用户潜在兴趣,为营销场景提供依据;
3. 发现联合推荐机会,实现商品长尾推荐;
4. 持续优化推荐效果,不断学习用户兴趣变化。

但Heout在使用协同过滤预测时也需考虑:

1. 新用户或冷启动商品的推荐问题,避免推荐准确率过低;
2. 推荐结果的多样性与创新性,避免内容重复或保守;

3. 用户数据的代表性,考虑推荐的全面性与准确性;
4. 推荐效果的持续监测与算法优化;
5. 研发团队与业务团队的密切协作,推动协同过滤产生真正影响。

### 第三步 获取用户的TOP推荐列表

该算法先获得用户与商品的潜在特征,然后基于特征余弦相似度计算每个商品的兴趣匹配度,选取匹配度最高的Top N商品作为个性化推荐,最后返回每个推荐商品的详细信息,为用户实现个性化推荐。

获取用户的TOP推荐列表详细信息主要包含以下步骤:

1. 根据用户ID获取用户的latent特征向量`user_latent_vec`,代表用户的兴趣特征;
2. 获取所有商品的latent特征向量`item_latent_vec`,代表每个商品的特征信息;
3. 计算`user_latent_vec`与`item_latent_vec`的余弦相似度,得到用户对每个商品的兴趣匹配度;
4. 根据匹配度由高到低排序所商品,选取前N个商品构成用户的TOP推荐列表;
5. 根据TOP推荐列表中的商品ID,查询每个商品的详细信息,如图片、名称、价格等;
6. 将TOP推荐商品及相关详情返回,实现个性化推荐。

具体代码实现如下:

```
python  
  
# 用户ID  
  
user_id = 'U1234'  
  
# 根据用户ID获取用户latent特征
```

```
user_latent_vec = user_latent_dict[user_id]

# 获取所有商品latent特征

item_latent_vec_list = list(item_latent_dict.values())

# 计算用户对每个商品的兴趣匹配度

interest_list = []

for item_latent_vec in item_latent_vec_list:

interest = cos_sim(user_latent_vec, item_latent_vec)

interest_list.append(interest)

# 根据兴趣匹配度获取TOP N推荐

sorted_interest = sorted(interest_list, reverse=True)

top_n = sorted_interest[0:10]

recommend_items = []

for interest in top_n:

item_index = interest_list.index(interest)

recommend_items.append(item_index)

# 查询TOP N商品详细信息

product_infos = get_product_informations()

recommends = []
```

```
for item in recommend_items:
    product_info = product_infos[item]
    recommends.append({
        'product_id': item,
        'image': product_info['image_url'],
        'name': product_info['name'],
        'price': product_info['price']
    })

# 返回个性化推荐结果

return recommends
```

#### 第四步 构造个性化推荐详情

个性化推荐详情是根据用户的TOP推荐列表生成的推荐商品相关信息,它提供推荐商品的名称、价格、图片等详细内容,为用户提供购买决策的重要参考。

该算法先获取TOP N推荐列表中的商品ID,然后根据ID分别查询每个商品的详细信息,并对信息进行整合,构造完整的个性化推荐商品详情。如果某商品信息不全,需要设置默认值以保证推荐详情的完整性。最后返回构造好的个性化推荐商品详情实现个性化推荐。

需要注意的是,推荐商品信息的获取与构造是个性化推荐的重要组成部分。信息详尽与匹配度高的推荐商品详情能显著提高用户体验,需要平台投入大量资源进行商品信息的收集与建设。

构造个性化推荐详情主要包含以下步骤:

1. 获取TOP N推荐列表,包含商品ID;
2. 根据商品ID查询商品详细信息,包括:
  - 商品图片:用于展示推荐商品;
  - 商品名称:用于推荐商品说明;
  - 商品价格:用于推荐商品价格参考;
  - 商品描述:用于更详细了解推荐商品;
  - 其他规格参数:不同商品有不同规格信息;
3. 将获取的商品详细信息组合为推荐商品详情;
4. 如果某商品信息不存在,需要采取备选方案,如设置默认图片等,以免推荐详情不全;
5. 返回构造好的个性化推荐商品详情。

具体代码实现如下:

```
python  
  
# TOP N推荐列表  
  
top_n = [1234, 2345, 3456]  
  
# 查询商品1234详细信息  
  
product_info_1234 = get_product_info(1234)  
  
image_1234 = product_info_1234['image_url']
```

```
name_1234 = product_info_1234['name']  
price_1234 = product_info_1234['price']  
desc_1234 = product_info_1234['desc']  
params_1234 = product_info_1234['params']
```

```
# 查询商品2345详细信息
```

```
# ...
```

```
# 构造推荐详情
```

```
recommends = []  
recommends.append({  
    'product_id': 1234,  
    'image': image_1234,  
    'name': name_1234,  
    'price': price_1234,  
    'desc': desc_1234,  
    'params': params_1234  
})
```

```
# 商品3456信息缺失,设置默认图片
```

```
recommends.append({
```

```
'product_id': 3456,  
'image': 'default.jpg',  
'name': '商品3456',  
})
```

# 返回构造好的个性化推荐详情

```
return recommends
```

Heout可以为每个TOP推荐商品构造如下个性化推荐详情:

```
{  
  
  "product_id": "P0001",  
  
  "name": "中国风印花连衣裙",  
  
  "price": "¥199",  
  
  "image_url": "[图片链接]",  
  
  "detail": "95% viscose, 5% spandex; 洗衣机可洗; 中国风立领卫衣式连衣裙;  
方片印花加饰蕾丝边; 修身版型显显腰身比例"  
  
}  
  
{  
  
  "product_id": "P0002",  
  
  "name": "松紧带休闲运动鞋",  
  
  "price": "¥299",  
  
}
```

```
"image_url": "[图片链接]",  
  
"detail": "EVA护踝设计,支撑关节稳定性;网面透气舒适,防臭设计;橡胶鞋底高耐磨防滑;鞋带可调,穿脱方便"  
  
}  
  
{  
  
"product_id": "P0003",  
  
"name": "修身牛仔西装外套",  
  
"price": "¥399",  
  
"image_url": "[图片链接]",  
  
"detail": "98%棉2%聚脂纤维,洗涤可洗; 时尚修身牛仔西装外套;  
前襟扣子设计,内袋方便随身物品; 两侧口袋,装饰皮扣设计; 质感优质牛仔面料"  
  
}  
  
...
```

个性化推荐详情为用户提供了丰富的商品信息,使用户可以快速了解推荐商品并作出购买决定。Heout可以根据详情为用户:

1. 展示推荐商品图片及描述,吸引用户关注;
2. 提供商品规格、材质等详细参数,增强用户购买信心;
3. 赠送优惠券或立减信息,刺激用户下单转化;
4. 根据用户购买与详情浏览的数据持续优化推荐算法和推荐内容;
5. A/B测试不同推荐详情样式,选择最优模板提高用户转化率。

但在生成与使用个性化推荐详情时,Heout也需考虑:

1. 推荐商品信息的准确性与实时性;
2. 推荐详情的个性化程度,避免过于模板化;
3. 不同用户的详情浏览偏好,提供个性化体验;
4. 转化效果的持续监测与详情优化;
5. 研发团队与业务团队的密切配合,发挥推荐详情的真正作用。

### 第五步：返回个性化推荐结果主要包含以下步骤:

该算法首先获取构造好的个性化推荐商品详情,然后进行推荐结果的整理与包装,设置推荐顺序并添加推荐理由,也提供图片版与文本版 2种格式的推荐结果。最后将完整的个性化推荐结果返回给用户,推荐结果包含推荐商品图片、文字介绍、使用理由等信息,方便用户理解与使用。

返回个性化推荐结果是推荐系统的最后一步,直接决定了用户体验的好坏。

1. 获取构造好的个性化推荐商品详情,包含每个推荐商品的图片、名称、价格等信息;
2. 整理推荐商品详情,设置好展示顺序、格式等,便于用户浏览;
3. 添加辅助信息,如推荐理由、使用说明等,帮助用户 better 理解个性化推荐结果;
4. 考虑到用户使用环境不同,推荐结果需要有多种格式,如图片版、文本版等;
5. 返回最终的个性化推荐结果给用户。

具体代码实现如下:

python

```
# 获取构造好的推荐商品详情
```

```
recommends = construct_recommends_details()
```

```
# 设置推荐商品顺序:价格由低到高
```

```
sorted_recommends = sorted(recommends, key=lambda x: x['price'])
```

```
# 添加推荐理由
```

```
reason = '根据您的浏览与购买记录,我们为您推荐以下商品'
```

```
sorted_recommends.insert(0, reason)
```

```
# 图片版推荐结果
```

```
pic_recommends = []
```

```
for recommend in sorted_recommends:
```

```
if 'image' in recommend:
```

```
pic_recommends.append(recommend['image'])
```

```
# 文本版推荐结果
```

```
text_recommends = [f"{recommend['name']} {recommend['price']}" for  
recommend in sorted_recommends]
```

```
# 返回最终推荐结果
```

```
return {
```

```
'reason': reason,
```

```
'pics': pic_recommends,  
'texts': text_recommends  
}
```

Shein 个性化推荐算法推演返回结果示例如下:

推荐详情:

商品1:夏季印花短袖T恤

图片:<https://xxx.com-Summer-Printed-Graphic-Tees-p-1922516-cat-1773.html>

价格:89元

商品2:高腰牛仔阔腿裤

图片:<https://xxx.com/High-Waist-Flare-Leg-Jeans-p-1977027-cat-1934.html>

价格:169元

推荐理由:根据您的浏览和购买记录,我们挑选了适合当前夏季的流行印花T恤和高腰牛仔阔腿裤进行推荐,希望您会喜欢!

使用说明:夏季必备的清爽印花和经典高腰牛仔裤,两款都很容易搭配,推荐您按照您的服装搭配风格选择。

推荐来源: Shein个性化推荐系统基于对您的浏览、收藏、购买记录等多种行为的分析,综合确定您的兴趣与偏好,为您生成每日个性化推荐。

备选推荐:

翻领宽松连衣裙

荷叶边上衣

格纹连衣裙

图片版推荐结果:

[印花T恤.jpg, 高腰牛仔裤.jpg, 连衣裙1.jpg, 上衣.jpg, 连衣裙2.jpg]

文本版推荐结果:

夏季印花短袖T恤 89元

高腰牛仔阔腿裤 169元

翻领宽松连衣裙 129元

荷叶边上衣 79元

格纹连衣裙 109元

该示例给出了两套夏季个性化推荐服装,包含 商品图片、价格、推荐理由以及备选推荐,并提供图片版与文本版两种结果格式,能够满足用户的不同浏览需求。同时,也解释了个性化推荐系统的来源与运作流程,这有助于用户了解推荐策略,建立信任,提高推荐质量。

通过追踪用户的各种行为与互动,分析用户的偏好与兴趣,为用户生成个性化商品推荐。个性化推荐结果不仅提供商品信息,也给出清晰的推荐理由与使用说明,并说明推荐来源,这有助用户理解推荐策略,采纳个性化推荐,是实现高质量个性化推荐的重要手段。

个性化推荐一定也面临诸如冷启动、长尾等问题,构建优质商品库及平衡推荐多样性也是值得探讨的话题。所有以上信息完全是根据个人的思路进行推演,不代表Shein采用该算法和方案。