- 购买意向
  - 定义
    - 给定用户评论和评论中的商品词，判断用户对该商品的购买旅程类别
      - 种草：消费者花在知晓及了解产品的时间和精力
        - 喜欢
          - 体现消费者对于产品的认同和喜爱
        - 被推荐
          - 表达被某人推荐或种草
      - 养草：消费者对产品的认同及进入消费者心智
        - 研究
          - 信息搜索或调研，例如询问产品/品牌/价钱/渠道等信息
        - 想要
          - 直接意愿
      - 拔草：实际进行购买
        - 购买
      - 复购：重复购买
        - 复购
      - 推荐
        - 直接表达想要推荐给别人
      - 粉丝带货：表达了日常粉丝支持Idol的代言买货支持的行为
        - 明星种草
          - 对明星代言或者明星同款的产品表达喜欢和支持
        - 明星带货
          - 购买明星代言或者明星同款
      - 其它
        - 广告言论和不相干商品的评论
    - 社媒10个分类
      - 喜欢
      - 想要
      - 研究
      - 购买
      - 复购
      - 明星种草
      - 明星带货
      - 推荐
      - 被推荐
      - 其它
      - 一条数据示例
        - 熬夜小唐 打了泪沟之后大概是医生打多了或者我个人原因导致很肿很肿！大家都嘲笑我我打了个眼袋（没看直播的人想想一下）预热拍视频都肿着，要拍11.1和双十一了，我还是决定去溶了有幸见到自己眼袋的样子，真是可怕立马去下单了一个雅萌的x眼罩了
        - 雅萌的x眼罩
        - 是购买
      - 电商5分类
        - 购买，复购，明星带货，推荐，其它
    - 问题
      - 化妆品领域数据训练的模型在香氛领域效果不好，应该补充香氛领域数据
    - 多任务模型
    - 数据
      - 处理完成后的数据总数是12121
      - {'购买': 4462, '其它': 4152, '推荐': 776, '喜欢': 744, '研究': 562, '想要': 414, '被推荐': 341, '明星带货': 326, '复购': 202, '明星种草': 142}
        - 数据不均衡问题，部分类别数据过多，其它类别过少
      - 准确率72%
        - 单标签：ACC: 71.135， F1MAC: 63.401 数据效果不如多标签数据
    - 数据不平衡如何通过标注扩充
      - 使用关键字匹配特定标签的数据，然后尝试匹配新数据进行标注
      - 从数据库获取数据，使用模型预测数据，然后过滤掉标签数量足够多的数据，对其它数据进行标注
    - 2次标注的比对
      - 2次标注不一样的条数有1944条
      - 标注不一样的数据和跳过的数据占总样本数比例是:0.4198704103671706
      - 共从json文件中收集标注的数据4630条
      - 训练后的准确率提升到78.5%
  - 拆分
    - 更新模型
      - 标注耗时: 多标签的需要比单标签每个多500条左右的数据
      - 社媒多标签模型： 10个类别，每个类别3000条数据 共30000
      - 社媒单标签模型： 10个类别，每个类别2500条数据 共25000
      - 减去现有标签，需要标注14700条左右
      - 电商多标签模型： 5个类别， 每个类别3000条数据 共15000
      - 电商单标签模型： 5个类别， 每个类别2500条数据 共12500 减去现有标签，需要标注6000条左右
      - 购买意向拆分模型:
        理想数据量如下，可以分3期，每期每个类别1000条，测试模型质量
      - 多标签分类模型
        - 损失函数
          - torch.nn.BCEWithLogitsLoss
            - 先进行了sigmoid，然后进行了二分类交叉熵损失函数
        - 多标签的准确率
          - sklearn.metrics.accuracy_score
            - accuracy_score(np.array([[0, 1], [1, 1]]), np.ones((2, 2)))
            - 输出： 0.5
          - 完全预测正确，多预测错误一个标签或这少预测正确一个标签都是错误的
        - 模型实现方式1
          - CLS+句子1+SEP+商品+SEP
            - 对CLS进行求sigmoid和二分类交叉熵
        - 模型实现方式2
          - CLS+句子1+SEP+商品+每个标签的id+SEP
            - 取出每个标签的向量
              - 对每个标签向量进行二分类交叉熵损失
        - 实现方式1和实现方式2，效果相差不大，可能是多标签的数量太少的原因，样本内的标签个数统计: ({1: 14986, 2: 1190, 3: 137, 4: 2}
          - 对于10个标签的多分类，第一种方式的准确率大概为77.68，第二种方式为77
          - 方式1的Hamming准确率是78.9，方式2是79.0
            - 准确率：80.213， Hamming: 82.383