

華東理工大學

信息科学与工程学院

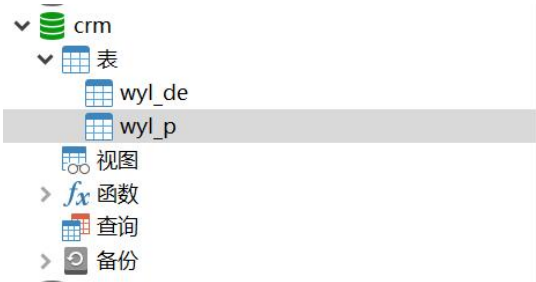
《电子商务》实验报告

系 别	计算机系
专 业	计算机科学与技术
班 级	计 201
学 号	20000370
姓 名	王仪琳
指导教师	翟 洁

2022-2023 学年 第 1 学期

《推荐系统》实 验 报 告

学号： 20000370 姓名： 王仪琳 班级： 计 201 成绩：

实验名称：CRM 系统之推荐算法	实验地点：信息楼 211
所使用的工具软件及环境：环境：Windows10 电脑； 软件：Navicat,IDEA	
一、实验目的： 熟悉关联算法、相邻方法，能应用关联算法或相邻方法设计个性化推荐系统，如推荐相应的优惠折扣或用户可能感兴趣的内容	
二、实验内容： 1、输入客户数据 2、应用关联算法或相邻方法等计算该用户适宜的优惠政策或可能感兴趣的内容 3、输出相应的推荐内容	
三、实验步骤： 1、请简要叙述系统应用技术（如应用的编程语言、数据库或文件技术等）、推荐算法 系统整体主要应用三层架构思想和 SSM（Spring, SpringMVC, MyBatis）开发框架，使用 Java 编程语言开发，数据库使用 MySQL 数据库进行数据的存储。 实现了两种推荐算法：关联算法和相邻算法实现了推荐系统。 2、请叙述数据存储的方式（如果是数据库，请列出表结构；如果是文件，请列出文件中记录的结构），并说明数据的来源（若是自动生成的，介绍一下生成规则） 采用数据库存储数据，总共有两个表，如下图所示：其中 wyl_de 表存储商品的基本信息（如下图 2 所示），wyl_p 表存储商品的购买记录（如下图 3 所示），两个表通过属性 pid 相联系：（如下图 4），两个表的属性含义详见表 1、2。数据库中部分数据见图 5、6 所示 	
图 1.数据库表 1	

字段	索引	外键	触发器	选项	注释	SQL 预览					
名						类型	长度	小数点	不是 null	虚拟	键
pid						int	10	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1
pname						varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
price						decimal	10	2	<input type="checkbox"/>	<input type="checkbox"/>	
fac						varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
detail						varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
dan						varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	

图 2.wyl_de 表

属性名	含义
pid	商品编号
pname	商品名称
price	商品价格
fac	商品生产厂家
detail	商品详细描述
dan	商品售卖单位

表 1.wyl_de 表字段含义

字段	索引	外键	触发器	选项	注释	SQL 预览					
名						类型	长度	小数点	不是 null	虚拟	键
oid						int	10	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
pid						int	10	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
pname						varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
price						decimal	10	2	<input type="checkbox"/>	<input type="checkbox"/>	
num						int	10	0	<input type="checkbox"/>	<input type="checkbox"/>	
fac						varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	
dan						varchar	20	0	<input type="checkbox"/>	<input type="checkbox"/>	
detail						varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>	

图 3.wyl_p 表

属性名	含义
oid	顾客编号
pid	商品编号
pname	商品名称
price	商品价格
num	顾客购买商品数量

fac	商品生产厂家
dan	商品售卖单位
detail	商品详情

表 2.wyl_p 表字段含义

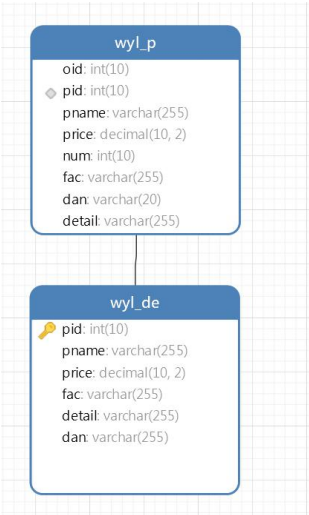


图 4.数据库表联系

对象	wyl_p @crm (local...	* 无标题 - 查询	wyl_de @crm (loca...	wyl_de @crm (loca...	* wyl_p @crm (loca...
开始事务	文本	筛选	排序	导入	导出
oid	pid	pname	price	num	fac dan detail
1	1	每日坚果	15.00	3	三只松箱 4种坚果+4种果干, 经典搭配, 有巴旦木仁、榛子仁、核桃仁、腰果仁、蔓越莓干等
1	2	手撕面包	49.90	3	三只松箱 精选优质原料, 大容量便携带, 隧道式烘烤工艺
1	3	7days坚果	15.00	5	百草味箱 健康的坚果和果干, 有巴旦木仁、榛子仁、核桃仁、腰果仁、蔓越莓干等
1	4	伊利优酸乳	5.00	1	伊利 盒 真实果汁, 一口入魂
2	1	每日坚果	15.00	3	三只松箱 4种坚果+4种果干, 经典搭配, 有巴旦木仁、榛子仁、核桃仁、腰果仁、蔓越莓干等
2	2	手撕面包	49.90	10	三只松箱 精选优质原料, 大容量便携带, 隧道式烘烤工艺
2	5	蒙牛纯牛奶	5.00	1	蒙牛 盒 来自优质牧场奶牛, 带给你自然风味
3	1	每日坚果	15.00	4	三只松箱 4种坚果+4种果干, 经典搭配, 有巴旦木仁、榛子仁、核桃仁、腰果仁、蔓越莓干等
4	5	蒙牛纯牛奶	5.00	1	蒙牛 盒 来自优质牧场奶牛, 带给你自然风味
5	6	AD钙奶	2.18	3	哇哈哈瓶 童年的味道, 少糖更健康
5	1	每日坚果	15.00	9	三只松箱 4种坚果+4种果干, 经典搭配, 有巴旦木仁、榛子仁、核桃仁、腰果仁、蔓越莓干等
5	10	乳酸菌小伴	20.90	34	三只松箱 早餐健康营养面包, 助消化
5	9	厚岩烧乳酪	19.90	21	三只松箱 很好吃, 甜而不腻, 早餐面包
6	1	每日坚果	15.00	35	三只松箱 4种坚果+4种果干, 经典搭配, 有巴旦木仁、榛子仁、核桃仁、腰果仁、蔓越莓干等
6	6	AD钙奶	2.18	32	哇哈哈瓶 童年的味道, 少糖更健康
7	1	每日坚果	15.00	23	三只松箱 4种坚果+4种果干, 经典搭配, 有巴旦木仁、榛子仁、核桃仁、腰果仁、蔓越莓干等
7	6	AD钙奶	2.18	2	哇哈哈瓶 童年的味道, 少糖更健康
7	2	手撕面包	49.90	1	三只松箱 精选优质原料, 大容量便携带, 隧道式烘烤工艺
7	7	光明白雪糕	7.20	2	光明 瓶 早餐家用分享装, 纯正好味道
8	1	每日坚果	15.00	2	三只松箱 4种坚果+4种果干, 经典搭配, 有巴旦木仁、榛子仁、核桃仁、腰果仁、蔓越莓干等
8	2	手撕面包	49.90	1	三只松箱 精选优质原料, 大容量便携带, 隧道式烘烤工艺
8	6	AD钙奶	2.18	2	哇哈哈瓶 童年的味道, 少糖更健康
8	8	550g卤味1	19.90	23	三只松袋 卤味鸭脖鸭翅熟食零食小吃
9	11	桂格原味麦	13.90	23	桂格 袋 早餐速食, 含FIBROQ益生元

图 5.数据库表 wyl_p 部分数据

对象	wyl_p @crm (localhost) - 表	* 无标题 - 查询	wyl_de @crm (localhost) - 表		
开始事务	文本	筛选	排序		
	导入	导出			
pid	pname	price	fac	detail	dan
1	每日坚果	15.00	三只松	4种坚果+4种果干, 经典搭配, 有巴旦木仁、榛子仁、核桃仁、腰果仁、蔓越莓干等	箱
2	手撕面包	49.00	三只松	精选优质原料, 大容量便携, 隧道式烘烤工艺	箱
3	7days坚果	15.00	百草味	健康的坚果和果干, 有巴旦木仁、榛子仁、核桃仁、腰果仁、蔓越莓干等	箱
4	伊利优酸乳	5.00	伊利	真实果汁, 一口入魂	盒
5	蒙牛纯牛奶	4.90	蒙牛	来自优质牧场奶牛, 带给你自然风味	盒
6	AD钙奶	2.18	娃哈哈	童年的味道, 少糖更健康	瓶
7	光明白雪酸牛奶	7.20	光明	早餐家用分享装, 纯正好味道	瓶
8	550g卤味零食礼包	19.90	三只松	卤味鸭脖鸭翅熟食零食小吃	袋
9	厚岩烧乳酪	19.90	三只松	很好吃, 甜而不腻, 早餐面包	箱
10	乳酸菌小伴侣	20.90	三只松	早餐健康营养面包, 助消化	箱
11	桂格原味麦片	13.90	桂格	早餐速食, 含FIBRQ益生元	袋
12	欧扎克酥脆麦片	49.90	欧扎克	营养早餐代餐饱腹, 内含酸奶快、坚果、水果干, 营养好滋味	袋
13	卡乐比燕麦	49.90	卡乐比	日本进口, 水果麦片, 即冲即饮	袋

图 6.数据库表 wyl_de 部分数据

3、请叙述系统的功能模块和具体的实现步骤（请画出所应用算法的流程图）

系统后端从功能来看分为两大模块：关联算法和相邻算法模块

3.1 关联算法模块：

Apriori 算法。算法开始前先通过 dao 层通过数据库 SQL 语句找到顾客总数和所有商品信息，创建 4 个 ArrayList，分别用来存放频繁项目集中商品的所有信息、频繁项目集中商品对应支持度、频繁项目集中商品对应名称、最终相关联的商品信息列表，为之后做准备。

首先要建立频繁项目集，我采用的方法是对数据库中商品进行遍历，分别计算出其相应的支持度，在这里，我规定支持度大于等于 0.3 的才可以进入 1 项目频繁项目集，通过 if 语句将符合条件的商品信息存入到上面前 3 个 ArrayList 中。（因为只用考虑两个商品之间的关联规则，所以这里只建立 1 项目频繁项目集）

然后，判断关联规则是否成立。通过前端 view 层传入商品名称，可以通过 service,dao 层的 findBuyNum(pname)方法计算出输入商品的支持度，之后遍历频繁项目集中的商品，通过 findTwo(pname, pname1)方法可以计算出输入商品和频繁项目集中商品的支持度，利用公式：可信度= $\frac{P(\text{condition and result})}{P(\text{result})}$ ，可以计算出每对规则的可信度，设可信度大于 0.5 时有意义，将符合条件的商品信息传入前端。

3.2 相邻算法模块：

主要思想是商品的价格相邻。算法开始前创建 ArrayList<wylProDetail>用来存放满足条件的商品信息列表，通过.allPrices()方法找到所有商品的价格，与输入商品的价格进行遍历结算，利用归一化公式： $|\text{输入价格}-\text{商品价格}|/\text{最大价格查}$ ，再利用欧几里得集合几何距离即可求出结果。规定欧几里得集合距离小于 0.5 时，商品相邻较近，较近的商品存入动态数组。

3.3 算法流程图

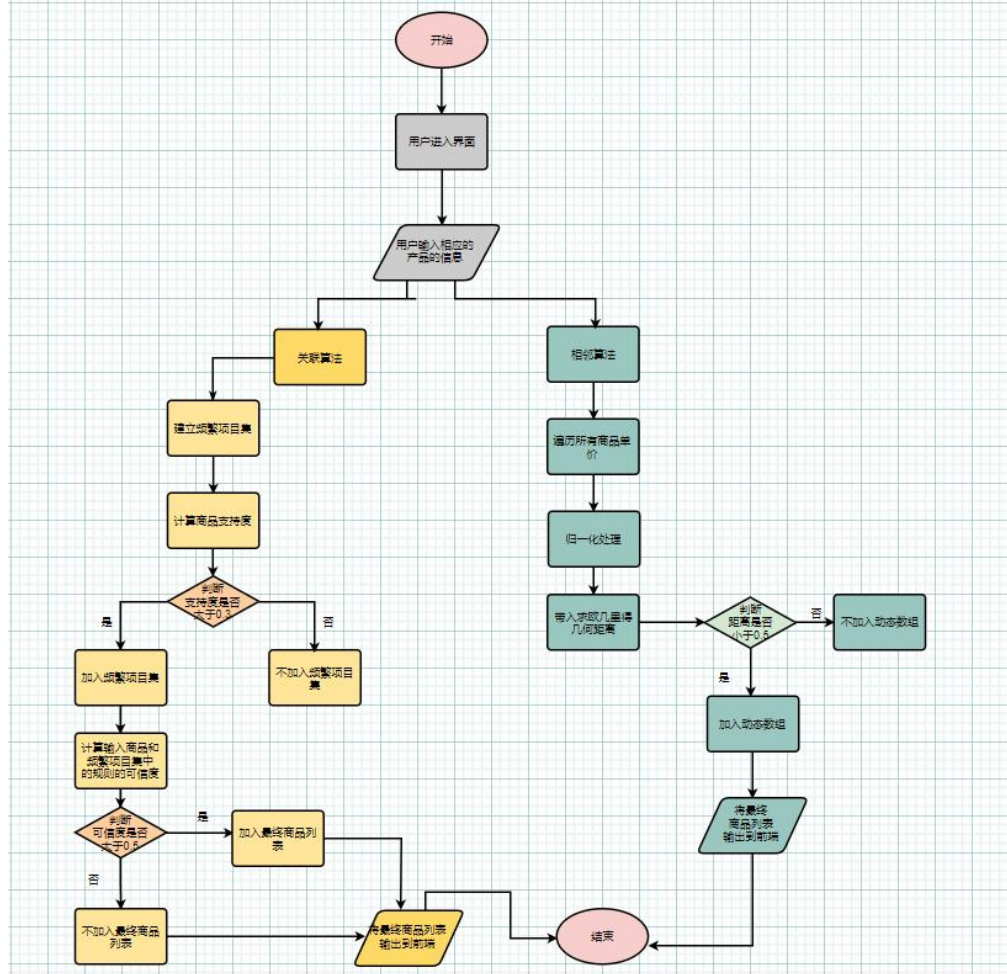


图 7.算法流程图

四、系统运行情况界面



图 8.前端 index 界面



图 9.查询界面（这里输入每日坚果）

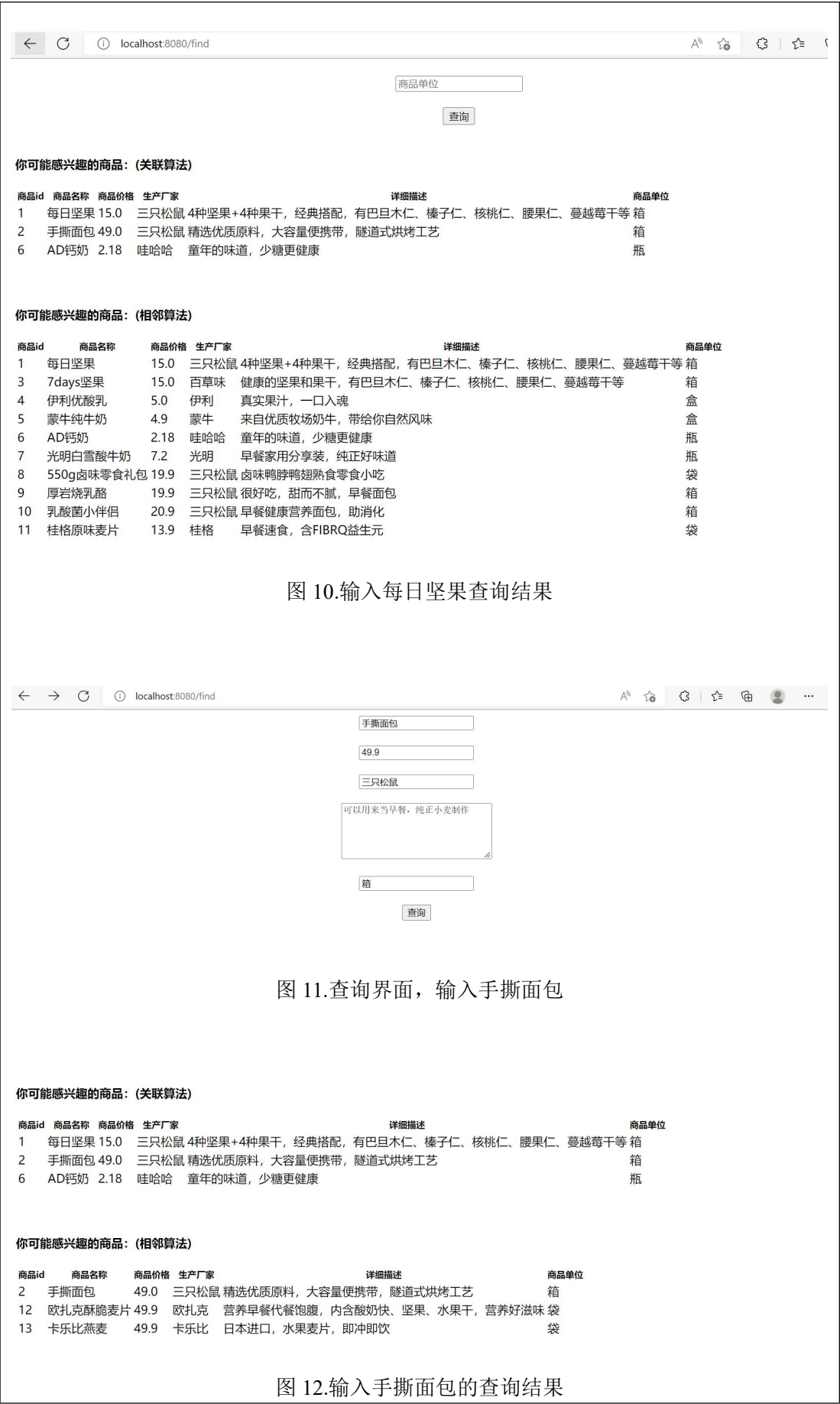


图 10.输入每日坚果查询结果

图 11.查询界面，输入手撕面包

图 12.输入手撕面包的查询结果

五、程序设计代码

1.系统目录结构

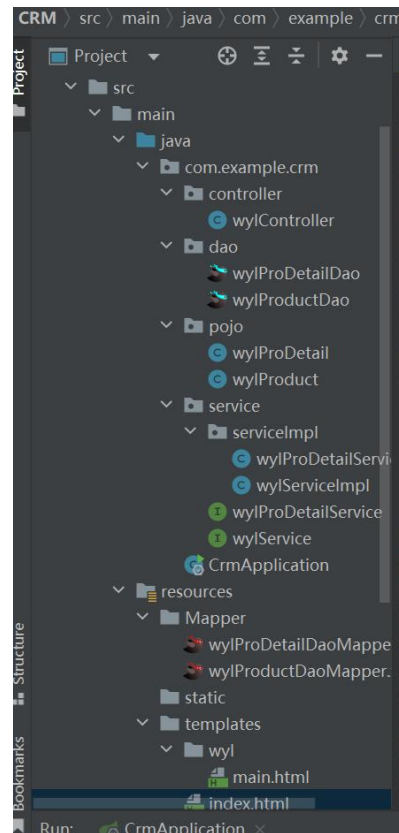


图 13.系统目录结构

2.dao 层

2.1wylProDetailDao

```
@Mapper
public interface wylProDetailDao {
    wylProDetail findByName(String pname);
    //找到最大和最小差值
    Double findDis();
    //所有价格
    List<Double> pList();
    //id
    wylProDetail findById(Integer pid);
}
```

wylProDetailMapper:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--mapper 的 xml 的头文件-->
<!DOCTYPE mapper
    PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
    "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.example.crm.dao.wylProDetailDao">
```

```

<select id="findByName" resultType="com.example.crm.pojo.wylProDetail">
    select DISTINCT* from wyl_de where pname=#{pname}
</select>
<select id="findDis" resultType="java.lang.Double">
    select max(price)-min(price) from wyl_de
</select>
<select id="pList" resultType="java.lang.Double">
    select price from wyl_de
</select>
<select id="findById" resultType="com.example.crm.pojo.wylProDetail">
    SELECT *from wyl_de where pid=#{pid}
</select>
</mapper>

```

2.2wylProductDao

```

@Mapper
public interface wylProductDao {
    Integer findBuyNum(String pname);
    //找到顾客总数
    Integer findAllCus();
    //找到所有商品名称
    List<String> proLists();
    //相关联的两件商品
    Integer findTwo(String pname,String pname1);
}

```

wylProductDaoMapper

```

<?xml version="1.0" encoding="UTF-8"?>
<!--mapper 的 xml 的头文件-->
<!DOCTYPE mapper
    PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
    "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.example.crm.dao.wylProductDao">

    <select id="findBuyNum" resultType="java.lang.Integer">
        select count(DISTINCT oid) from wyl_p where pname=#{pname}
    </select>
    <select id="findAllCus" resultType="java.lang.Integer">
        select count(DISTINCT oid) from wyl_p
    </select>
    <select id="proLists" resultType="java.lang.String">
        select DISTINCT pname from wyl_p
    </select>
    <select id="findTwo" resultType="java.lang.Integer">

```

```

        select count(DISTINCT oid) from wyl_p where oid in( select DISTINCT oid from
wyl_p where pname =#{pname})and pname=#{pname1}
    </select>
</mapper>

```

3.service 层:

3.1wylProDetailsService

```

public interface wylProDetailsService {
    wylProDetail findByName(String pname);
    Double findDis();
    List<Double> allPrices();
    wylProDetail findById(Integer pid);
}

```

3.2wylService

```

public interface wylService {
    Integer findBuyNum(String pname);
    Integer findAllCus();
    List<String> proList();
    Integer findTwo(String pname,String pname1);
}

```

4.pojo

4.1 商品售货单

```

package com.example.crm.pojo;

public class wylProduct {
    private Integer oid;
    private Integer pid;
    private Integer num;
    public Integer getOid() {
        return oid;
    }
    public void setOid(Integer oid) {
        this.oid = oid;
    }
    public Integer getPid() {
        return pid;
    }
    public void setPid(Integer pid) {
        this.pid = pid;
    }
    public Integer getNum() {
        return num;
    }
    public void setNum(Integer num) {

```

```

        this.num = num;
    }
    private String pname;
    private Double price;
    private String fac;
    private String dan;
    private String detail;
    public String getPname() {
        return pname;
    }
    public void setName(String pname) {
        this.pname = pname;
    }
    public Double getPrice() {
        return price;
    }
    public void setPrice(Double price) {
        this.price = price;
    }
    public String getFac() {
        return fac;
    }
    public void setFac(String fac) {
        this.fac = fac;
    }
    public String getDan() {
        return dan;
    }
    public void setDan(String dan) {
        this.dan = dan;
    }
    public String getDetail() {
        return detail;
    }
    public void setDetail(String detail) {
        this.detail = detail;
    }
}

```

4.2 商品详情单

```

package com.example.crm.pojo;

public class WylProDetail {

```

```
private Integer pid;
private String pname;
private String fac;
private String dan;
private String detail;
private Double price;
public Integer getPid() {
    return pid;
}
public void setPid(Integer pid) {
    this.pid = pid;
}
public String getPname() {
    return pname;
}
public void setPname(String pname) {
    this.pname = pname;
}
public String getFac() {
    return fac;
}
public void setFac(String fac) {
    this.fac = fac;
}
public String getDan() {
    return dan;
}
public void setDan(String dan) {
    this.dan = dan;
}
public String getDetail() {
    return detail;
}
public void setDetail(String detail) {
    this.detail = detail;
}
public Double getPrice() {
    return price;
}
public void setPrice(Double price) {
    this.price = price;
}
}
```

5.controller

关联算法:

```
@Controller
public class WylController {

    @Autowired
    WylService wylService;

    @Autowired
    WylProDetailService wylProDetailService;

    @GetMapping("/toWyl")
    public String toWyl(){
        return "wyl/main";
    }

    @PostMapping("/find")
    public String find(String pname, Double price, Model model){
//关联算法
        Integer allCus = wylService.findAllCus();
        List<String> wylProducts = wylService.proList();
        //存放频繁项目集
        ArrayList<WylProDetail> pros=new ArrayList<>();
        //存放频繁项目集中支持度
        ArrayList<Double> zhichi=new ArrayList<>();
        //存放频繁项目集中商品名称
        ArrayList<String> pinName=new ArrayList<>();
        //最终的关联商品列表
        ArrayList<WylProDetail> fin=new ArrayList<>();
        //设最小支持度为 0.3
        //1 项目频繁项目集极其支持度
        for (int i = 0; i < wylProducts.size(); i++) {
            String name = wylProducts.get(i);
            Double zhi=(double)(wylService.findBuyNum(name))/allCus;
            if(zhi>=0.3){
                WylProDetail p1= wylProDetailService.findByName(name);
                pros.add(p1);
                zhichi.add(zhi);
                pinName.add(name);
            }
        }
//关联规则
        for (int j = 0; j < pinName.size(); j++) {
            String pname1=pinName.get(j);
            Integer tNum = wylService.findTwo(pname, pname1);
            //System.out.println(tNum);
            double both =(double) tNum / (double)allCus;
            //System.out.println(both);
        }
    }
}
```

```

//知道到输入的商品的支持度，从表里面找? double a=zhichi.get(j);
double a=Double.valueOf(wylService.findBuyNum(pname))/(double)allCus;
//System.out.println(a);
//System.out.println(both/a);
//可信度大于 0.5 时关联规则有意义
if((both/a)>0.5){
    //System.out.println(both/a);
    wylProDetail p= pros.get(j);
    fin.add(p);
}
}
model.addAttribute("plist",fin);

```

相邻算法：

```

//最近邻算法
//商品价格归一化
ArrayList<wylProDetail> arrayList=new ArrayList<>();
Double dis = wylProDetailService.findDis();
List<Double> doubles = wylProDetailService.allPrices();
for(int k=0;k<doubles.size();k++){
    Double aDouble = doubles.get(k);
    double v = Math.abs((price - aDouble)) / dis;
    double v1 = Math.sqrt(Math.pow(v, 2));
    if(v1<0.5){
        wylProDetail byId = wylProDetailService.findById(k+1);
        arrayList.add(byId);
    }
}
model.addAttribute("plist1",arrayList);
return "wyl/main";
}
}

```


六、实验体会

1.算法设计体会：

首先是算法方面，因为是对关联算法和相邻算法的应用，所以加深了对这两种算法的理解。

在设计关联算法的时候，整体的思想是先构建频繁项集，再根据频繁项集计算验证规则的合理性，我这里采用的是支持度大于 0.3 加入频繁项集，可信度大于 0.5 判断为合理的规则，我认为这里的算法实现思想并不是难点，但是数据存储结构确实一个要考虑的问题，我在构建频繁项集的时候其实创建了 3 个动态数组，所以从数据结构上来看并不是很简洁。这是算法存在的一个问题。在设计相邻算法的时候按照课堂上讲的方法先归一化处理商品价格，再通过欧几里得几何距离来判断即可，对于欧几里得距离小于 0.5 的判断为满足条件的商品，在这里使用两个动态数组存储相应的归一化后的数值和满足条件的商品信息。

2.系统框架体会：

系统设计方面，开始是打算使用 `JavaWeb:servlet,jsp,JDBC`,但是最后这次使用了 SSM 来设计系统，采用这个框架的原因是实现前后端的交互较为简单，SSM 框架较为成熟，通过本次实验使我对 MVC 的设计思想加深了理解，对 SSM 的使用规则更熟悉。

3.选择设计早餐零食、饮品推荐系统的原因及体会：

我们小组这次设计的主要内容是基于社区团购的电商推荐系统，我选择可用于早餐的零食、饮品推荐，主要是考虑到一个如今学生或者家长的上学、上班时间都较早，如果时间来不及就没有时间自己做饭或者去社区早餐店吃饭，但是一些零售店或者超市并没有专门的售卖早餐类零食、饮品的分区，导致大多数有次需求的顾客很多时候想到早餐零食时只想到牛奶和面包。通过这个早餐零食、推荐系统可以使顾客在输入一些早餐时看到推荐的其他商品，既解决了单独购买时商品形式单一的问题，又可以促进其他商品的销售。

4.不足之处：

本项目的一些不足之处：在最初建表的时候，没有考虑充分，实际上数据库的表是有冗余的情况存在的，商品售卖表实际上不需要存储商品的详情介绍，这点是需要改进的地方。

七、教师评语

该学生完成了 CRM 系统之推荐算法的实验，（很好/较好/基本）完成了实验任务，算法设计（很好/较好/尚可），实验结果（很好/较好/尚可），实验体会（深刻/较深刻/尚可）。

因此总体评价为_____。

教师签字：

2022 年 12 月 1 日