

# 智能食堂数据分析系统

指导老师：陈宇杰

答辩学生：丁纪翔、陈鑫、姜宇晨

# 我们的团队



陈鑫  
1925103005

丁纪翔  
1925102007

姜宇晨  
1925102018

骆华钦  
1925103027

梁恺恒  
1925103021

# 人员分工

完成任务	开始日期	结束日期	持续时间	负责人	7. 9	7. 10	7. 11	7. 12	7. 13	7. 14	7. 15	7. 16	7. 17	7. 18	7. 19
项目设计	2022/7/9	2022/7/10	2days	丁纪翔											
数据采集	2022/7/10	2022/7/11	2days	骆华钦											
数据库搭建与处理	2022/7/11	2022/7/13	3days	骆华钦											
分布式数据处理	2022/7/12	2022/7/16	5days	陈鑫											
A R I M A 销售额预测算法	2022/7/13	2022/7/15	3days	陈鑫											
前端开发	2022/7/10	2022/7/18	9days	丁纪翔											
后端开发	2022/7/10	2022/7/18	9days	姜宇晨											
接口测试	2022/7/14	2022/7/18	5days	梁恺恒											
架构评估	2022/7/9	2022/7/10	2days	骆华钦											
服务器部署及运维	2022/7/9	2022/7/19	11days	丁纪翔、姜宇晨											
文档撰写, PPT制作	2022/7/15	2022/7/19	5days	梁恺恒											

# 目 录

*CONTENTS*

01

选题背景与意义

02

实现方法及过程

03

项目成果展示

04

总结分析期望



# 1 选题背景与意义

# 选题的背景



随着高校招生规模的不断扩大，传统的食堂生产方式及经营理念已成为高校发展的瓶颈。由于高校学生饮食需求的多样化，个性化日益复杂，传统食堂已经逐渐失去竞争力。因为缺乏学生满意度反馈，亦或是反馈获取时间长，渠道单一，使得食堂运营管理具有滞后性，无法及时发现和改善问题。在食堂财务方面，需要大量的表单数据来反映食堂销售额，比较繁琐，容易出现错误。

# 选题的意义



## 选题意义一

智能食堂数据分析系统后台能够将对每日食堂菜品销售数据进行清算与统计，按日返回营业额、订单量、人流量、菜品平均评分。



## 选题意义二

智能食堂数据分析系统计算每一餐摄入的食物含有哪些营养成分，学生和食堂都可以通过食堂管理系统平台一目了然地看到今日摄入的营养物质。



## 选题意义三

用户用餐结束后，可以通过评分功能对食堂的菜品进行点评，智慧食堂数据分析系统的后台会收集这些数据，之后会将这些数据发送给食堂管理者。



实现方式及过程



# 业务内容

业务一

智慧食堂系统登录请求业务

业务二

显示当天的订单总数、订单平均评分、总客流量、总营业额

业务三

显示一个月内每日总营业额、总访客量、订单平均评分、总订单数量。

业务四

预测下一天的总营业额、总访客量、订单平均评分、总订单数量

业务五

一个月内三餐的各营养平均摄入量

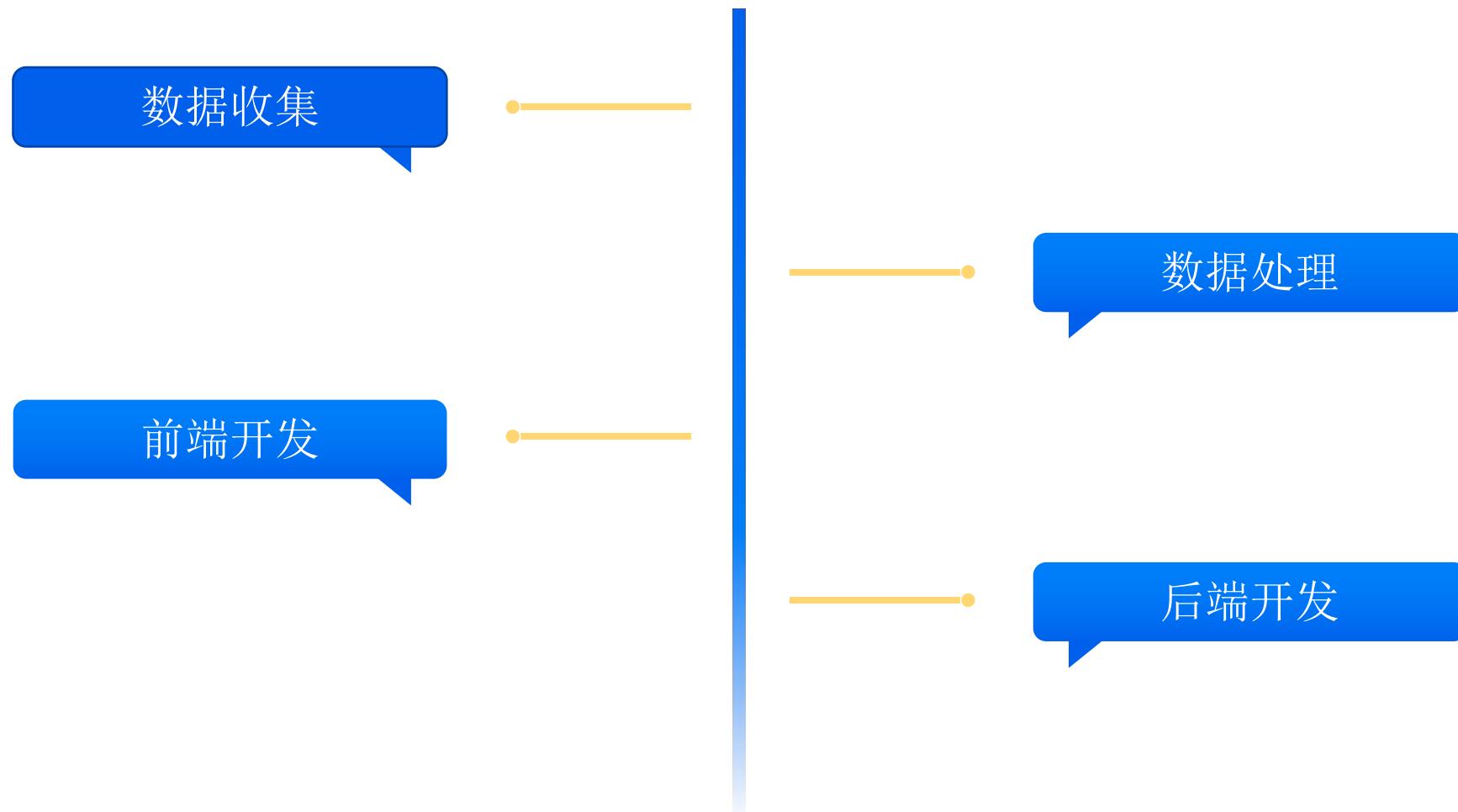
业务六

最受欢迎的10道菜、评分  
最差的10道菜

业务七

显示最近60订单数据（分页显示），可支持条件查询

# 实现过程



# GitHub

sylvanding / AI-Restaurant-Data-Analysis

Public

Notifications

Fork 0

Star 1

Code Pull requests Actions Security Insights

main ▾ 1 branch 0 tags

Go to file

Code ▾

Sylvan Ding deploy page		
apache-tomcat-9.0.64	first commit	25 minutes ago
arima	first commit	25 minutes ago
datasets	first commit	25 minutes ago
docs	deploy page	16 minutes ago
hadoop-hive-sqoop-mysql	first commit	25 minutes ago
img	first commit	25 minutes ago
interface	first commit	25 minutes ago
mysql	first commit	25 minutes ago
LICENSE	Initial commit	8 hours ago
README.md	first commit	25 minutes ago
Tasks- Arranging(in-Gantt).xlsx	first commit	25 minutes ago

## About

AI Restaurant Data Analysis

[github.com/sylvanding/ai-restaurant-dat...](https://github.com/sylvanding/ai-restaurant-dat...)

Readme

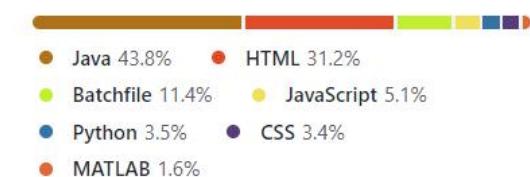
MIT license

1 star

1 watching

0 forks

## Languages



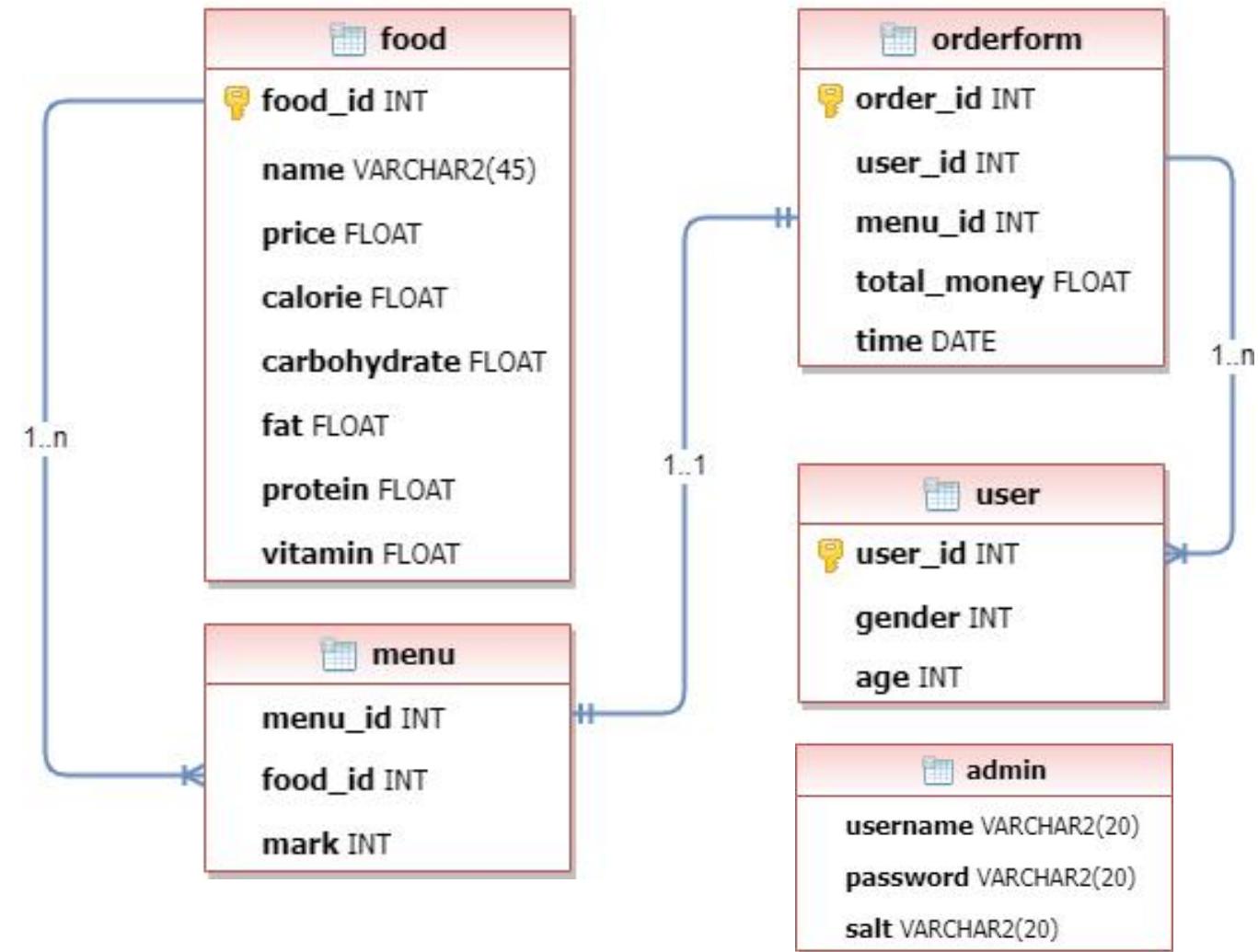
# 数据收集

数据来源：<https://www.boohee.com/food>

	菜名	价格	卡路里(大卡)	碳水化合物(克)	脂肪(克)	蛋白质(克)	维生素(克)
0	稀饭	0.3、0.5	46.00	9.90	0.30	1.10	0.10
1	青菜瘦肉粥	2.0、3.0	68.79	9.01	1.40	5.12	0.07
2	水煮蛋	1	143.00	0.10	10.50	12.10	0.00
3	肉包	2.5	205.00	27.11	5.08	13.40	0.08
4	奶黄包	2	230.73	42.65	6.91	8.68	0.11
5	煮黄豆	1	390.00	34.20	16.00	35.00	15.50
6	包菜	1	139.92	25.89	1.79	6.32	1.12
7	素炒花菜	1.5	82.93	4.41	4.92	7.29	1.31
8	炒南瓜	1	68.07	11.07	2.88	1.02	1.85
9	炸鸡腿	5	243.63	18.78	9.47	16.07	0.10
10	家常豆腐	1.5	85.91	4.25	5.78	5.27	0.79
11	香煎豆腐	2	92.00	3.35	6.96	6.32	0.03
12	贡丸汤	2	22.14	0.64	0.69	3.38	0.03
13	鱼丸汤	2	102.27	7.64	3.76	9.40	0.02
14	西芹炒木耳	3	59.93	10.14	2.06	1.29	1.89
15	美味土豆条	2.5	81.00	17.80	0.20	2.60	1.10
16	红枣银耳汤	2.0、3.0	20.45	5.09	0.08	0.68	1.18
17	扬州炒饭	4.0、5.0	151.17	12.87	7.27	6.65	0.28
18	炒米粉	4.0、5.0	151.63	28.94	3.15	2.06	1.27
19	炒白果	4.0、5.0	75.13	10.81	3.08	2.24	1.90
20	炒面	4.0、5.0	410.00	80.10	4.80	12.40	1.50
21	毛豆炒腊肠	5	104.90	9.50	4.50	8.11	3.18
22	豆浆	0.3、0.5	31.00	1.20	1.60	3.00	0.00
23	白馒头	0.45	223.00	47.00	1.10	7.00	1.30
24	煎蛋	1.5	164.00	0.20	11.70	12.30	-
25	烧麦	1.5	211.16	39.02	2.94	7.03	0.74
26	咸菜	1	33.00	6.50	0.30	2.20	2.10
27	煮花生	1	264.37	12.18	21.11	10.11	6.89
28	白菜	1	20.00	3.40	0.20	1.60	0.90
29	素炒西兰花	2	80.69	6.90	5.25	3.90	3.99
30	素炒豆芽	1	34.21	2.84	2.04	1.68	1.27

60	鸭皮卷	5	327.20	3.40	30.20	11.40	0.00
61	瓦罐煨汤	5	190.00	5.20	9.50	20.90	0.00
62	干煎秋刀鱼	5	198.82	7.70	12.22	14.58	0.03
63	黄焖鸡	6	121.26	3.02	7.40	10.79	0.27
64	五香条	6	197.48	2.95	13.71	15.94	0.74
65	清蒸鱼片	6	135.52	0.39	6.99	17.80	0.04
66	八宝粥	1.5、2.0	76.00	8.80	4.00	2.10	0.80
67	韭菜饼	2.5	25.00	4.50	0.40	2.40	2.40
68	茶叶蛋	1.2	148.65	9.03	6.55	11.60	2.65
69	紫薯包	2	261.83	50.10	2.10	9.80	0.00
70	酱瓜	1.5	90.46	5.15	5.72	5.45	0.60
71	西红柿炒蛋	1.5	71.11	3.58	4.46	4.77	0.66
72	炒上海青	1	18.00	3.20	0.20	1.70	1.60
73	素白萝卜	1	16.00	4.00	0.10	0.70	1.10
74	素炒青瓜	1	30.88	3.00	1.86	1.49	1.09
75	沙茶豆腐角	1.5	84.00	3.40	5.30	6.60	0.00
76	黑椒猪大排	7	264.00	1.70	20.40	18.30	0.00
77	红烧肉	7	286.99	4.11	52.79	8.61	0.55
78	香炸鸡柳	6	190.77	14.55	6.52	19.63	0.02
79	水煮肉片	6	124.30	5.18	8.31	10.29	0.75
80	糖醋肉	7	268.25	14.97	17.88	12.14	0.50
81	酱香鸭	6	229.75	1.87	18.18	14.62	0.10
82	糖醋锅包肉	6	268.25	14.97	17.88	12.14	0.50
83	姜母大鸭腿	8	98.26	1.45	15.60	16.42	0.02
84	口水鸡	6	173.91	3.63	9.18	17.62	0.77
85	椒盐鸭翅	6	293.36	0.30	26.36	13.77	0.03
86	鱼香肉丝	7	109.41	4.74	5.33	11.62	0.76
87	水煮柠檬鱼	7	1.76	0.30	0.06	0.05	0.06
88	花卷	0.7	214.00	45.60	1.00	6.40	1.50

# 数据库介绍



# 数据库介绍

food表

数据数量：100条

	food_id	name	price	calorie	carbohydrate	fat	protein	vitamin
0		稀饭	0.5	46	9.9	0.3	1.1	0.1
1		青菜瘦肉粥	3	68.79	9.01	1.4	5.12	0.07
2		水煮蛋	1	143	0.1	10.5	12.1	0
3		肉包	2.5	205	27.11	5.08	13.4	0.08
4		奶黄包	2	230.73	42.65	6.91	8.68	0.11
5		煮黄豆	1	390	34.2	16	35	15.5
6		包菜	1	139.92	25.89	1.79	6.32	1.12
7		素炒花菜	1.5	82.93	4.41	4.92	7.29	1.31
8		炒南瓜	1	68.07	11.07	2.88	1.02	1.85
9		炸鸡腿	5	243.63	18.78	9.47	16.07	0.1
10		家常豆腐	1.5	85.91	4.25	5.78	5.27	0.79
11		香煎豆腐	2	92	3.35	6.96	6.32	0.03
12		贡丸汤	2	22.14	0.64	0.69	3.38	0.03
13		鱼丸汤	2	102.27	7.64	3.76	9.4	0.02
14		西芹炒木耳	3	59.93	10.14	2.06	1.29	1.89
15		美味土豆条	2.5	81	17.8	0.2	2.6	1.1

menu表

数据数量：106370条

	menu_id	food_id	mark
▶	485003	94	4
	485003	77	8
	485003	41	7
	763461	17	8
	763461	96	9
	424064	97	3
	424064	43	3
	424064	3	8
	424064	35	9
	90817	80	4
	90817	13	9
	90817	29	4
	870329	53	3
	870329	89	8
	870329	70	7

# 数据库介绍

orderform表

数据数量： 35496条

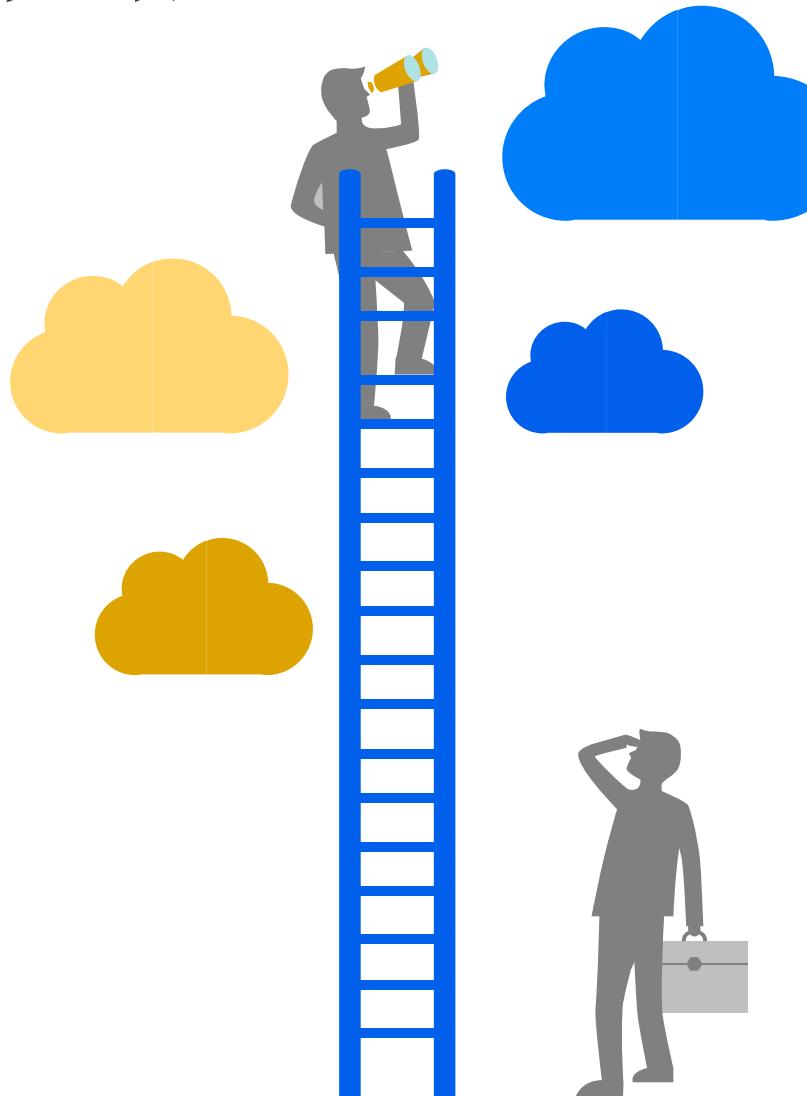
	order_id	user_id	menu_id	total_money	time
▶	66486	354699	310627	7.2	2022-06-01 06:00:00
	818489	16530	778428	17	2022-06-01 06:01:00
	228513	889312	216828	4	2022-06-01 06:02:00
	668604	191427	94548	5.7	2022-06-01 06:04:00
	870424	973175	914992	6	2022-06-01 06:04:00
	842910	928243	509193	2.65	2022-06-01 06:05:00
	697382	266156	137393	18	2022-06-01 06:05:00
	567992	940179	774019	14.5	2022-06-01 06:05:00
	460127	207718	28591	18.5	2022-06-01 06:05:00
	808803	612802	167119	8	2022-06-01 06:05:00
	23027	360379	32099	4	2022-06-01 06:05:00
	754040	581623	745086	18	2022-06-01 06:06:00
	341430	965460	821810	2.5	2022-06-01 06:06:00
	138600	472763	851930	7	2022-06-01 06:06:00
	127255	703357	500666	12.45	2022-06-01 06:06:00

user表

数据数量： 500条

	user_id	gender	age
▶	292	0	18
	3165	1	17
	4361	0	22
	5610	0	20
	13656	1	25
	13995	0	18
	15062	1	19
	16530	1	22
	20101	1	22
	22124	0	18
	24039	0	18
	29994	1	21
	31557	1	18
	33436	0	20
	37698	0	22

# 数据处理



## 数据上传本地和 Hadoop

将user表、food表、orderform表、menu表存入



## 上传到 Hive 仓库

创建 catten\_data 数据库，然后依次创建四张表用来存储



## Hive 中进行数据操作

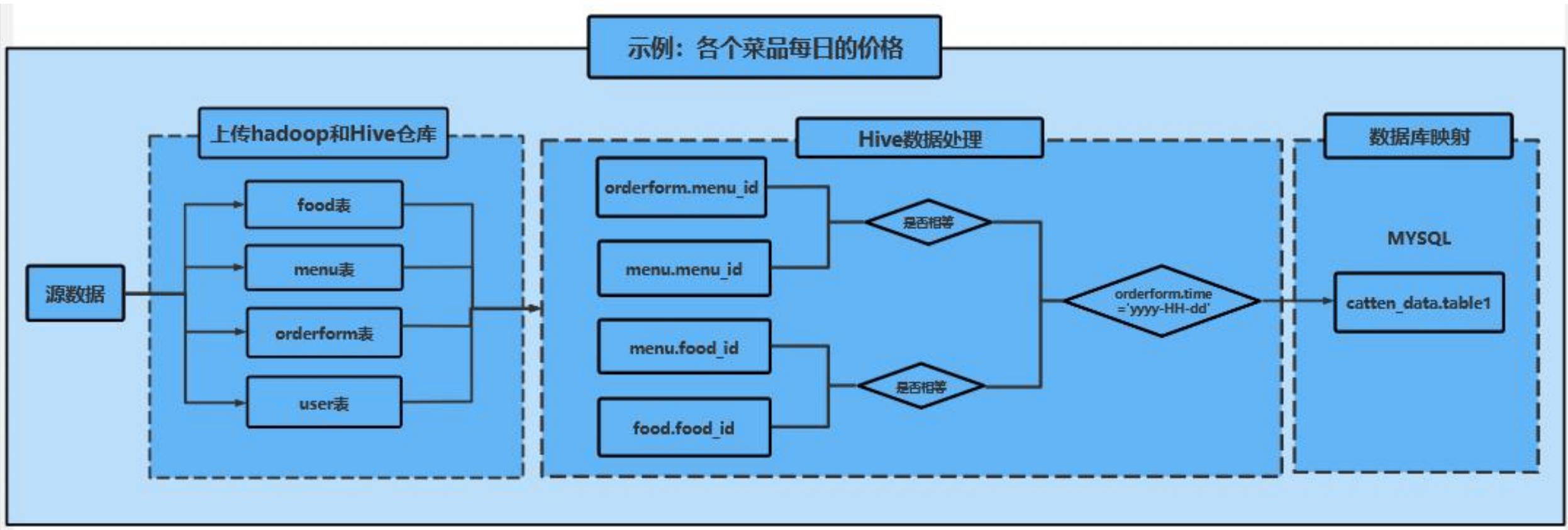
创建临时表来储存结果



## 使用 Sqoop 将处理结果映射到Mysql

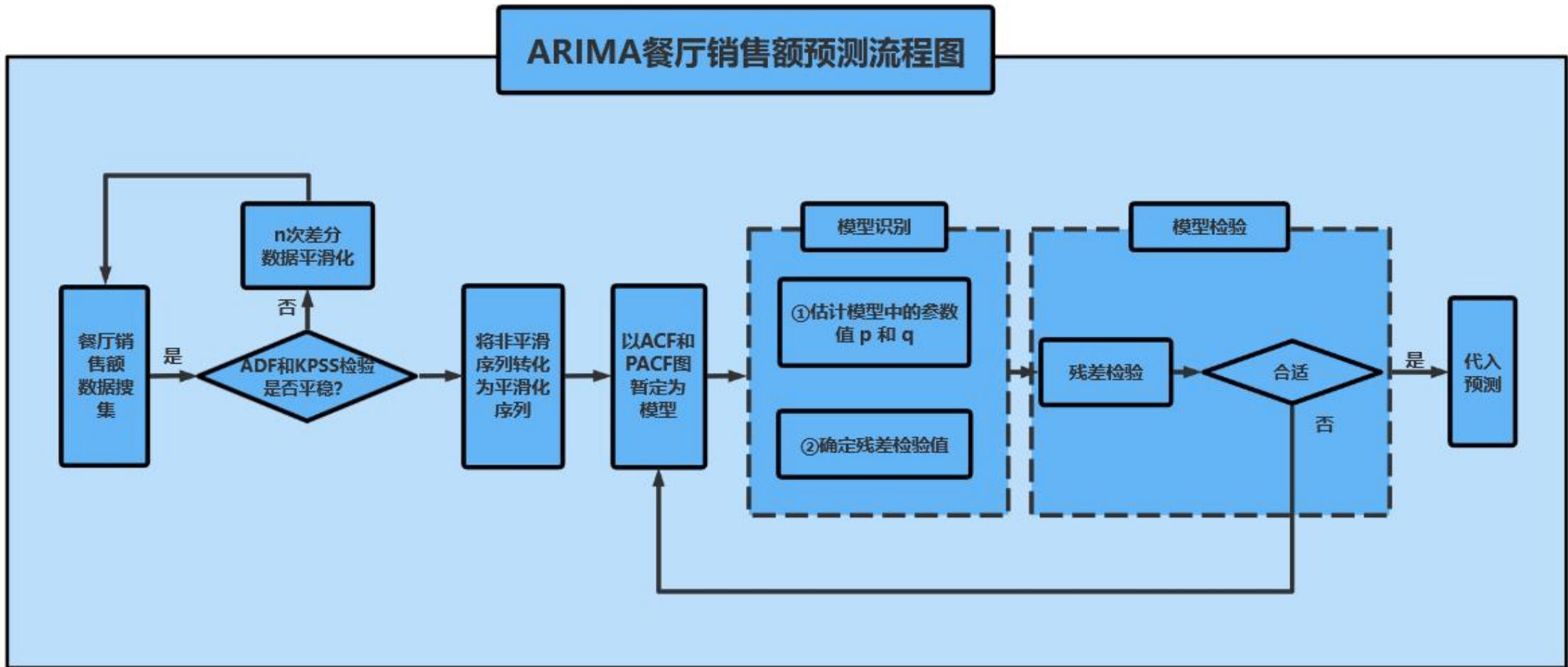
在Mysql创建 catten\_data 数据库, 对应处理好后的相应表

# 分布式数据处理示例



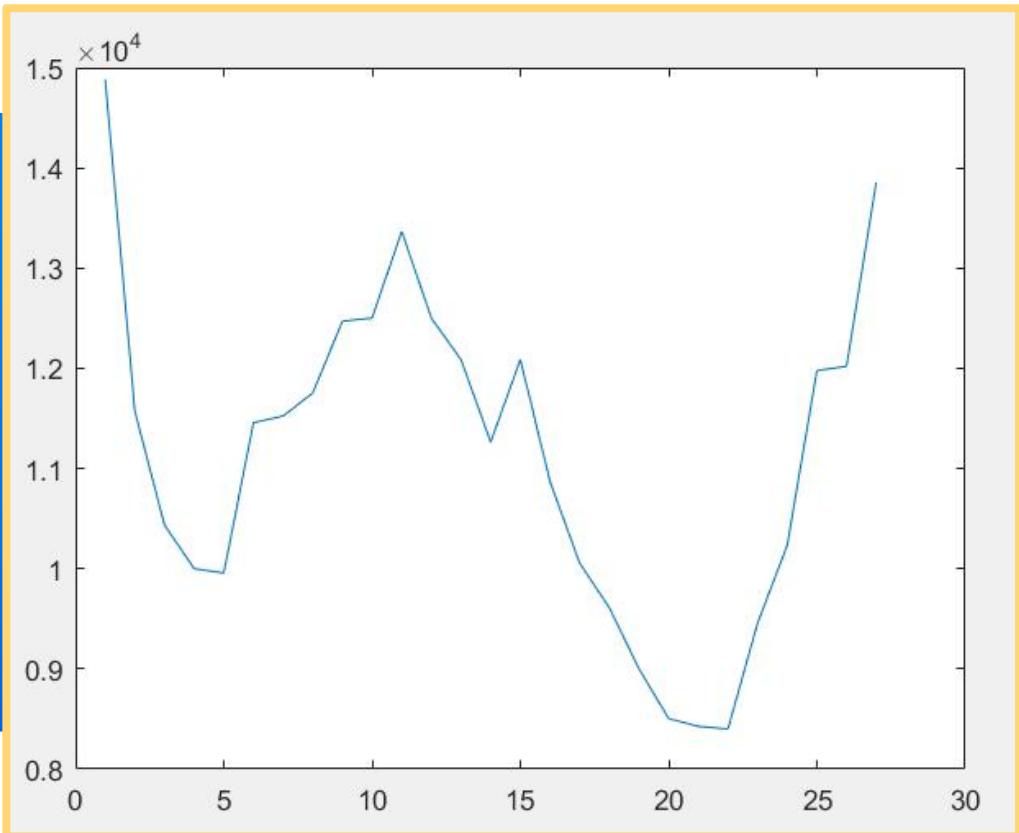
# ARIMA时间序列预测销售额算法

流程图



# ARIMA时间序列预测销售额算法

## 1. 时间序列的获取



为了预测销售额，第一步就是需要进行数据筛选和获取，需要的数据是餐厅的每日总销售额和相应的时间节点，并将此做为时间序列数据。左图为源数据的销售额随时间变化图。

# ARIMA时间序列预测销售额算法

## 2.时间序列平稳化

使用ARIMA时间序列模型要求时间序列必须是平稳的，所以第一步是对原始数据进行平稳性检验。检验方法这里采用ADF检验和KPSS检验。

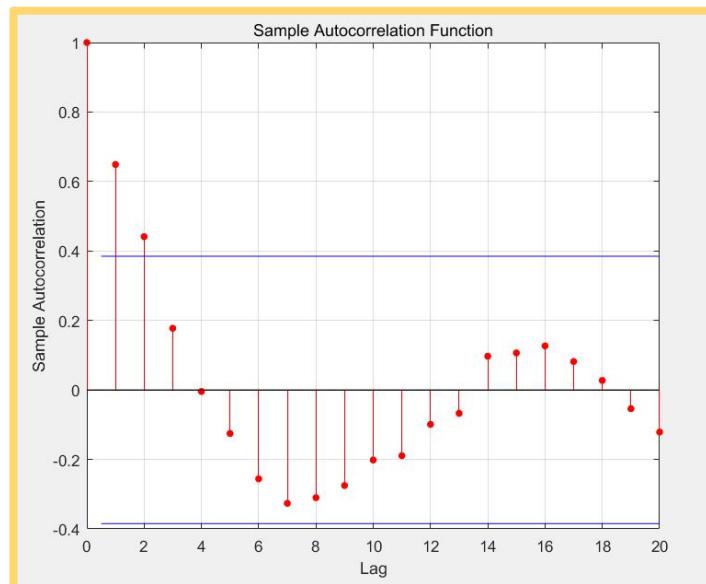
```
%平滑性检验, yd1_h_adf =1, yd1_h_kpss =0, 通过检验  
y_h_adf = adftest(Y);  
y_h_kpss = kpsstest(Y);
```

如果没有通过检验则需要对Y进行去差分，一阶差分后如果平稳则确定数据，如果不平稳则再取二阶差分，如此反复直至数据达到平稳。该步骤确定了模型ARIMA（p ,q , i）中的i一阶则对应i=1。由此将非平稳的时间序列源数据转化成了平稳的时间序列源数据。

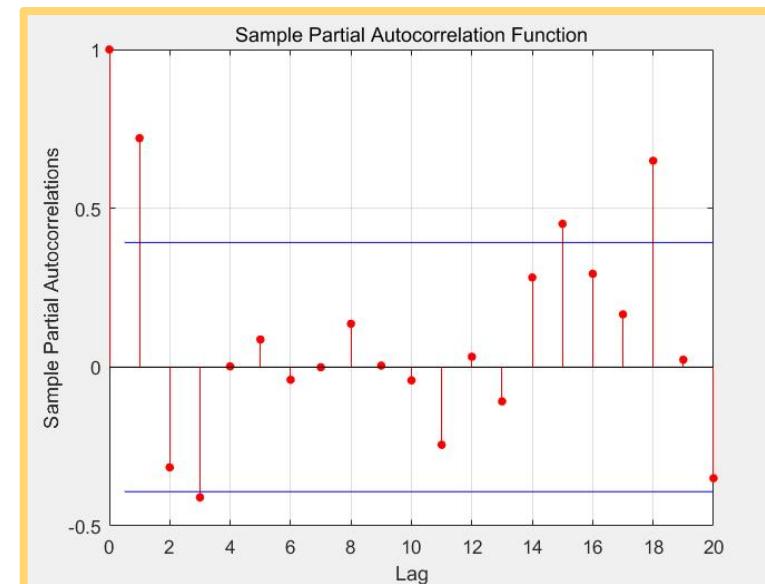
# ARIMA时间序列预测销售额算法

## 3.模型识别

模型识别主要是为了确定ARIMA模型的阶数，对应ARIMA ( $p, q, i$ ) 中的 $p$ 和 $q$ 。这里采用的模型阶数获取方法是ACF和PACF法，即自相关函数和偏自相关函数。根据数据画出的ACF和PACF图如下所示：



ACF自相关图

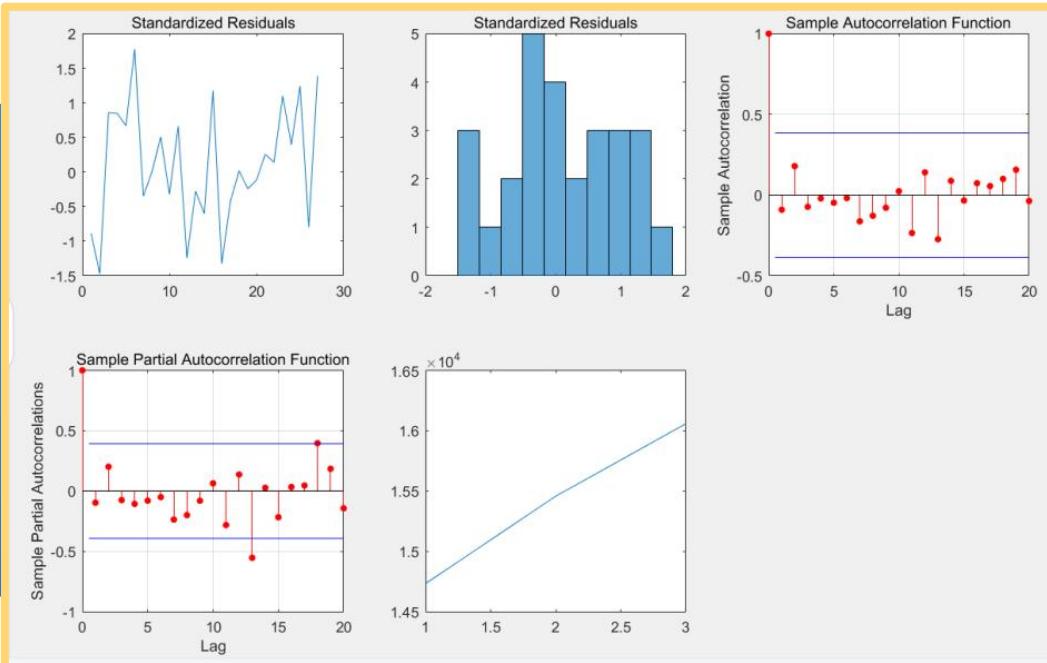


PACF偏自相关图

目测可以发现其中 $q$ 等于2， $p$ 等于18，相对应ARIMA模型中的ARIMA ( $p, q, i$ ) 中的 $p$ 和 $q$ 。

# ARIMA时间序列预测销售额算法

## 4. 残差检验



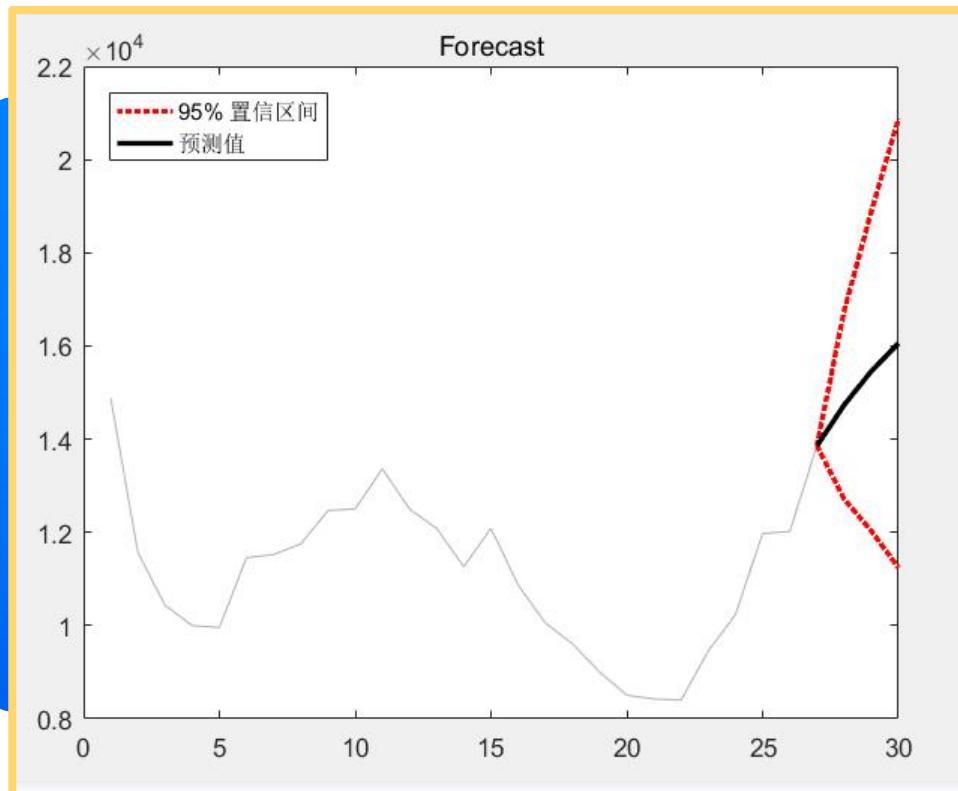
为了确保确定的阶数合适，还需要进行残差检验。残差即原始信号减掉模型拟合出的信号后的残余信号。如果残差是随机正态分布的、不自相关的，这说明残差是一段白噪声信号，也就说明有用的信号已经都被提取到ARMA模型中了。左图为得出的残差检验图。

其中**Standardized Residuals**图是查看残差是否接近正态分布，理想的残差要接近正态分布，除此之外还画出了直方图进行展示；**ACF**和**PACF**检验残差的自相关和偏自相关，理想的结果应该在图中不存在超出蓝线的点；最后一张图是检验残差是否接近正太分布的。

# ARIMA时间序列预测销售额算法

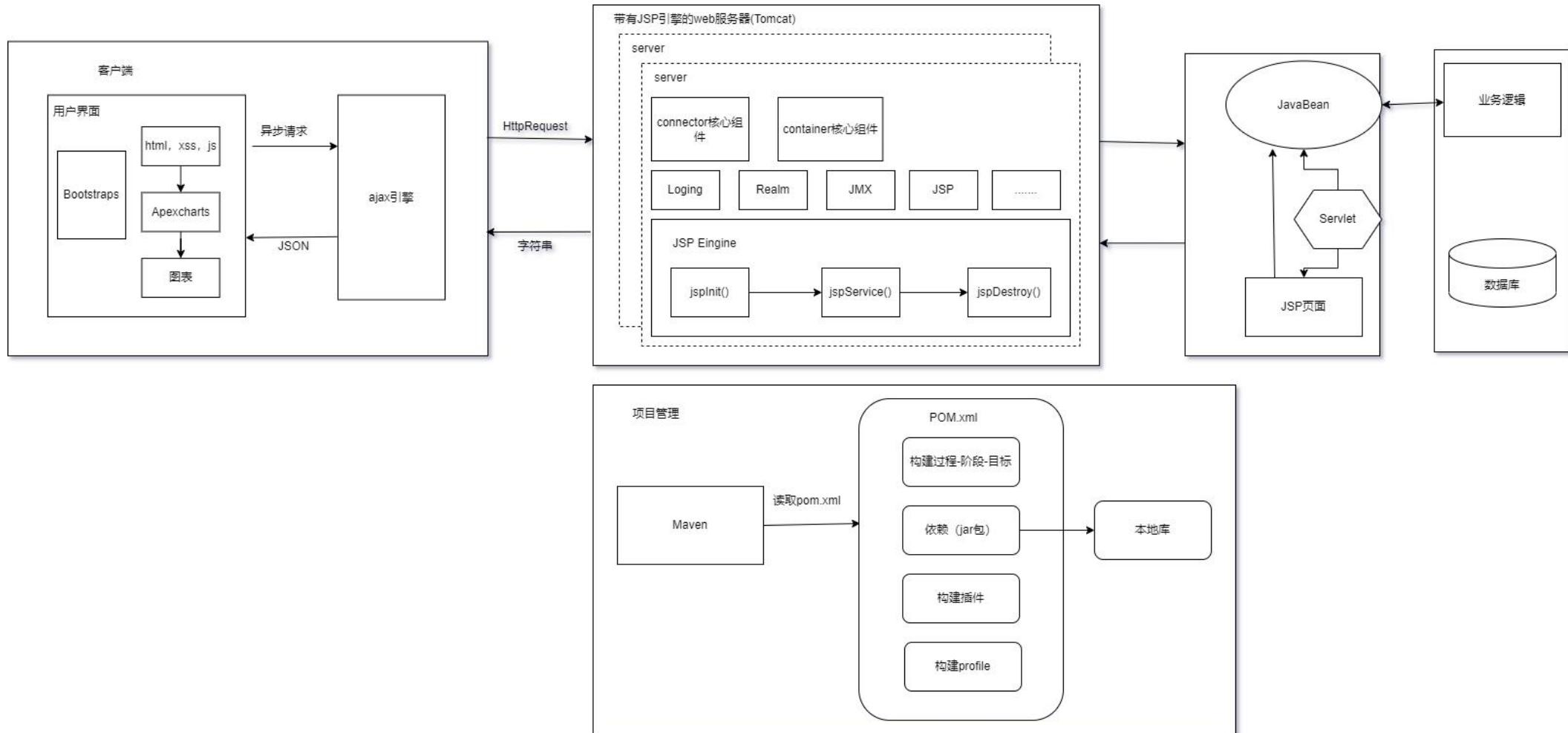
## 5.模型预测

这里是将餐厅销售额的前27天作为源数据，预测餐厅最后3天的销售额，即设置预测步数为3步，最终的结果图如下所示：

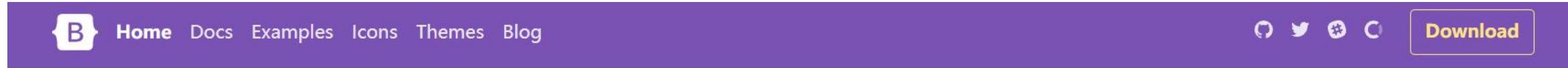


左图中灰线为用来训练的27个数据，黑线为未来值的预测，红线为95%置信区间上下限。也就是说未来真实值有95%的概率落在这个范围内。除此之外，可以看到使用ARIMA方法进行长期预测的结果是趋势性的。由此便完成了基于ARIMA时间序列的餐厅销售额预测，预测餐厅销售额最后三天数据分别是14733、15457和16058，观察可发现所得结果与源数据相差不大，符合价格趋势。

# 前端框架



# 前端概况

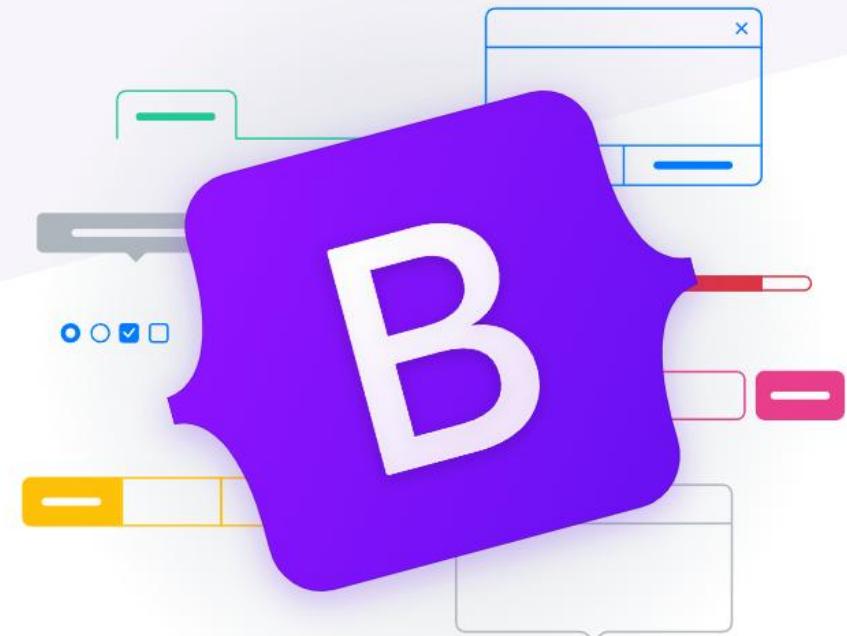


## 使用 Bootstrap 构建快速、响应式布局的网站

Bootstrap 是全球最流行的前端开源工具包，它支持 Sass 变量和 mixins、响应式网格系统、大量的预建组件和强大的 JavaScript 插件，助你快速设计和自定义响应式、移动设备优先的站点。

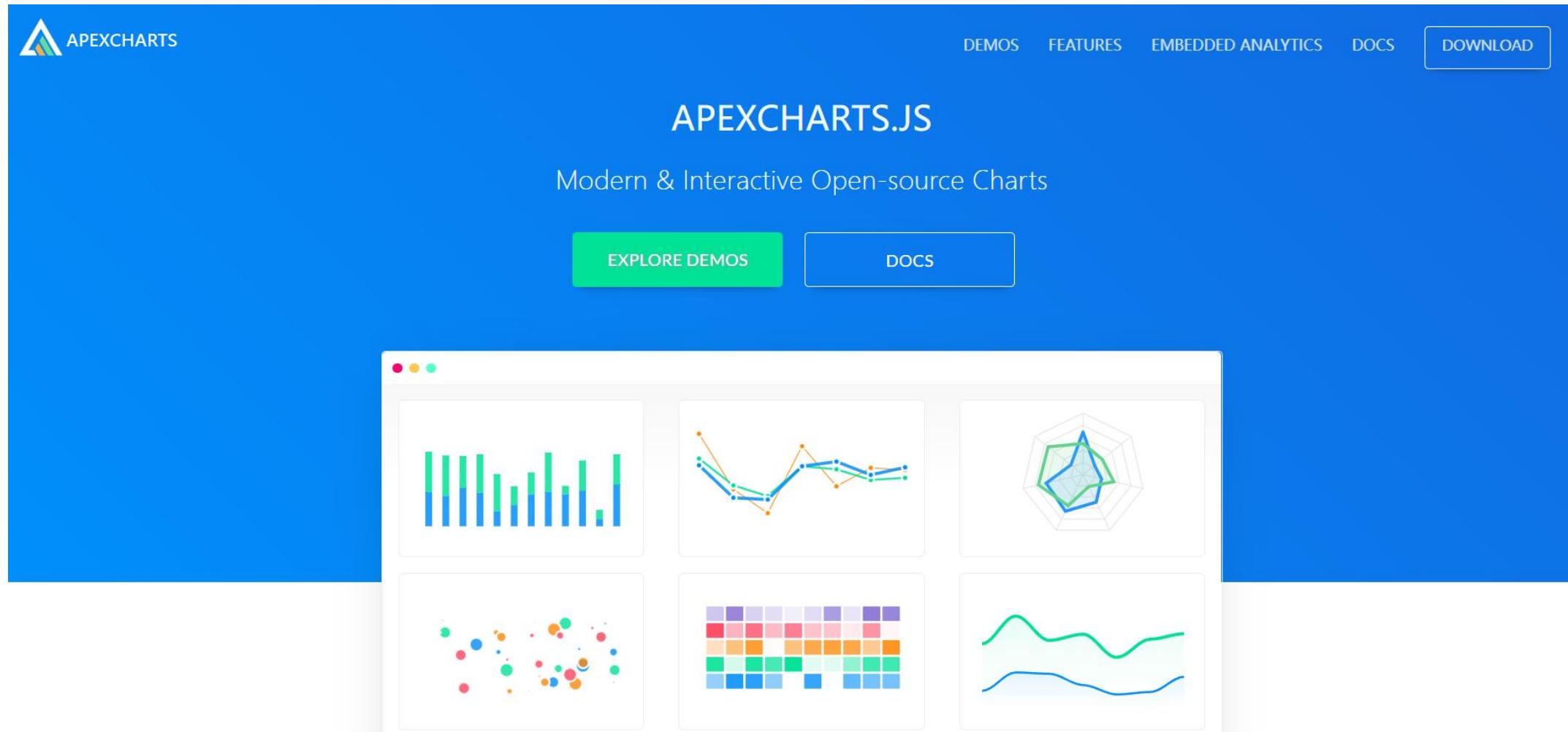
[快速入门](#)[下载 Bootstrap](#)

当前版本: [v5.1.3](#) · [v4 版本](#) · [历史版本列表](#)



- 前端采用*Bootstrap*响应式布局设计，兼容不同分辨率的设备，给用户提供更好的视觉使用体验。

# 前端概况



- 数据可视化使用ApexCharts. 相较于Echarts, ApexCharts更为轻量, 图标简洁美观, 可定制化程度高。但是ApexCharts文档不齐全, 给开发带来了难度。

# 前端概况

## jQuery Ajax Methods

`jQuery.get( url, [data], [callback],  
[type] )`

`jQuery.getJSON( url, [data],  
[callback] )`

`jQuery.getScript( url, [callback] )`

`jQuery.post( url, [data], [callback],  
[type] )`

`load( url, data, callback):`

`serialize()`

`serializeArray(`

`ajaxComplete( callback )`

`ajaxStart( callback )`

`ajaxError( callback )`

`ajaxSend( callback )`

`ajaxStop( callback )`



- 请求远程数据采用jQuery的AJAX实现，进入网站时异步请求所有数据，接着渲染图表，实现数据可视化。

# 前端概况

☰ README.md

## fastjson

Java CI passing codecov 80% maven central 2.0.9.graal release v1.2.83 license Apache 2 Gitpod Ready-to-Code

oss-fuzz fuzzing

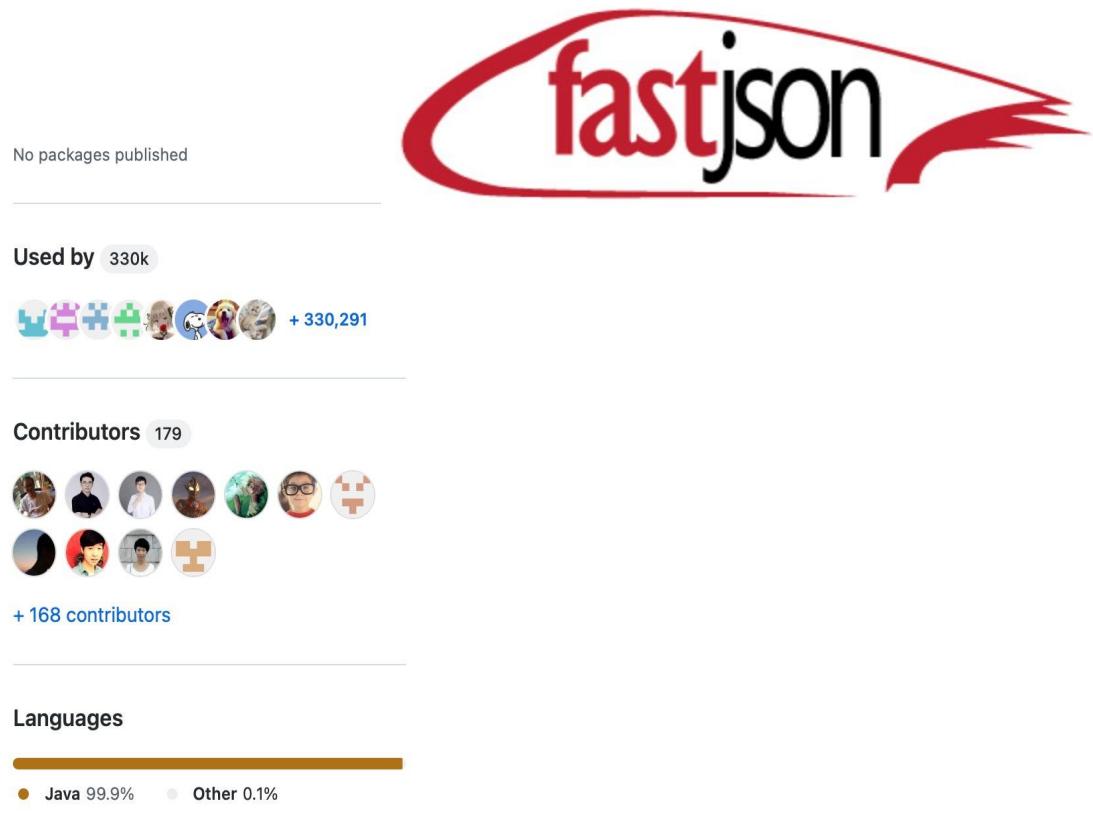
Maintainability 3.35

Fastjson is a Java library that can be used to convert Java Objects into their JSON representation. It can also be used to convert a JSON string to an equivalent Java object. Fastjson can work with arbitrary Java objects including pre-existing objects that you do not have source-code of.

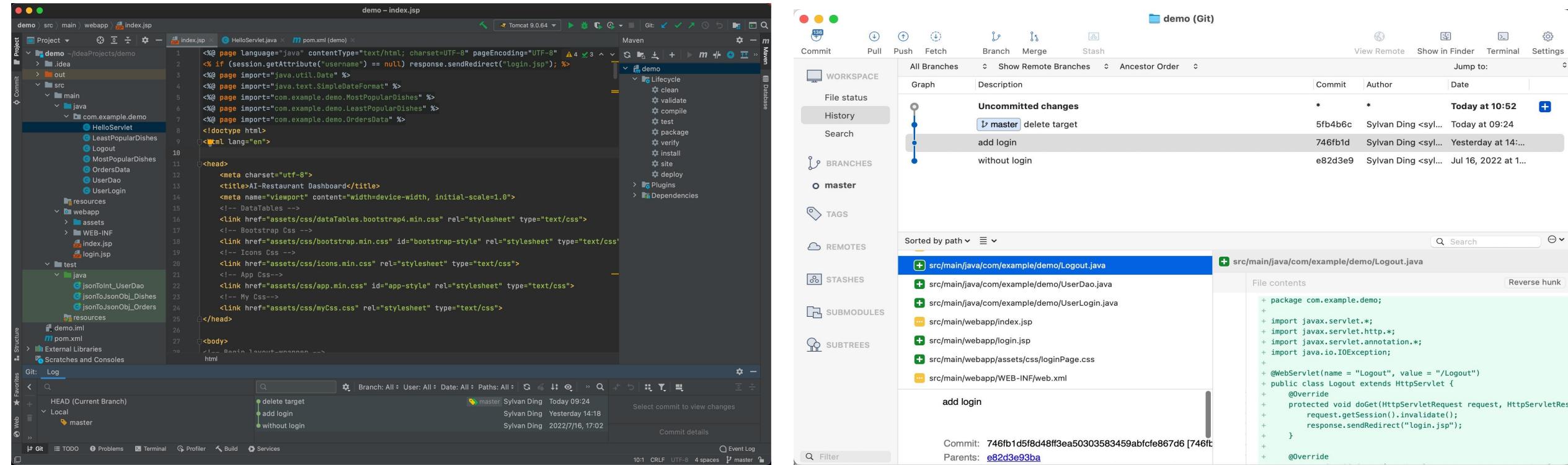
### Fastjson Goals

- Provide the best performance on the server-side and android client
- Provide simple `toJSONString()` and `parseObject()` methods to convert Java objects to JSON and vice-versa
- Allow pre-existing unmodifiable objects to be converted to and from JSON
- Extensive support of Java Generics
- Allow custom representations for objects
- Support arbitrarily complex objects (with deep inheritance hierarchies and extensive use of generic types)

- 基本实现前端静态网页样式以后，创建JSP动态网页，实现登陆和数据请求等业务。
- JSP中的JSON解析使用阿里的Fastjson，实现json对象与JavaBean对象的转换。



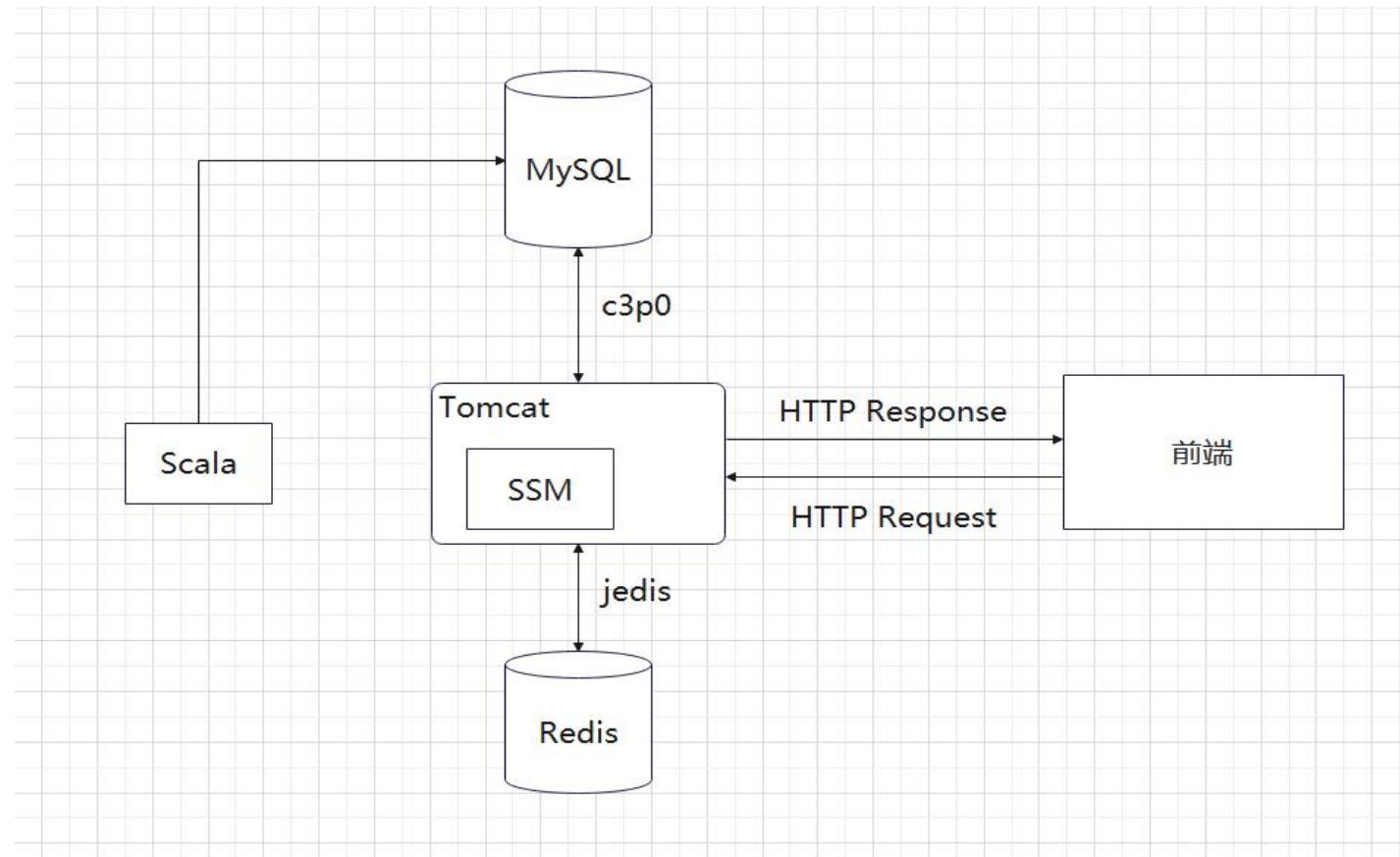
# 前端概况



- 使用Maven管理JAVA项目，使用Git进行项目版本控制。

# 后端框架

采用Spring+SpringMVC+Mybatis+Redis框架



# 后端概况

- 使用SSM框架实现八个模块共十七个接口



**接口2:**  
通过Cookie检验登陆状态  
**接口地址:**  
`http://175.178.233.192:8080/spring/isLogin`  
**请求方式:**  
`get/post`  
**返回格式:**  
`json`  
**请求示例:**  
`http://175.178.233.192:8080/spring/isLogin`  
**响应示例:**  
`{"msg":0}`  
**Body请求参数:**  
无  
**返回参数说明:**

名称	类型	说明
msg	int	响应码, 0代表用户已登录, 1代表未找到此用户, 需跳转登录页

# 后端概况

- 版本迭代：

7.09 项目搭建以及服务器环境搭建

实现了业务1的接口：订单按照时间聚集

7.10 发现业务1的接口*Bug*, 给出了规避方法

实现了业务5的接口：男女消费水平差异

7.11 优化业务1的接口：返回数据按时间段升序

优化业务5的接口：中文参数名改为英文

实现了业务8的接口：按日返回营业额、订单量、人流量、平均评分

7.12 对部分接口增加了参数判断

实现了业务2的接口：返回最热门、冷门、销量最高、最低的菜品列表

找了一天的打包版本*bug*.....

7.13 实现了业务4的接口：每天三个时间段人流量

7.14 应前端要求将业务8接口进行整合

框架整合了redis

实现了管理员登录接口以及添加了相关工具类  
添加跨域请求配置

实现业务11的接口：分页查询

7.15 为业务4添加更多查询数据项

整合了总交易额接口

删除了原总交易额接口

新增三餐平均营养值接口

修复了业务8接口传参为1时会炸的*bug*

7.16 为部分无参接口整合了redis

新增，修改部分数据接口

完善登录接口

7.18 实现Cookie校验接口

# 横切事务

- 对部分后端处理较慢的无参接口使用AOP横切织入的方式整合Redis，对Response进行序列化存储
- 采用c3p0连接池提高复用减轻数据库压力

AOP示例：

```
@Around(value = "execution(* com.jiang.service.DayDataServiceImpl.getEvrDay(..))")
public Object aroundGetMonthEverydayData(ProceedingJoinPoint pjp) throws Throwable {
    Object result = null;
    // @Before
    if(redisUtil.hasKey("getMonthEverydayData")){
        return redisUtil.get("getMonthEverydayData");
    }
    // 执行切点
    result = pjp.proceed();
    // @After
    redisUtil.set(key: "getMonthEverydayData",result.toString(), time: 1000);
    return result;
}
```

使用`@Around`切入，对原代码没有侵入性，对应缓存存在时对切点执行进行拦截，否则在切点执行后进行Redis缓存操作。

# 横切事务

- 对部分后端处理较慢的无参接口使用AOP横切织入的方式整合Redis，对Response进行序列化存储
- 抛弃传统的JDBC，采用c3p0连接池减轻数据库压力

效果：



原生请求响应时间：5.32秒



织入Redis后响应时间：0.2秒

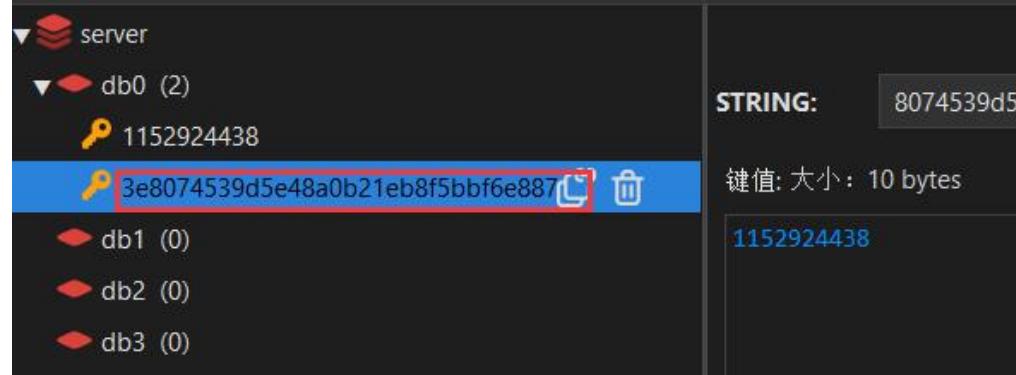
# 单点登录

- 登录业务对密码进行前后端两次MD5加密，通过写入Cookie和Redis实现单点登录

```
@Override  
public int login(String username, String password, HttpServletRequest request,  
  
    //根据用户名获取用户  
    Admin admin = adminLoginMapper.getAdminByUsername(username);  
    if(null == admin){  
        //用户不存在  
        return 0;  
    }  
    //密码校验  
    if(!MD5Util.formPassToDBPass(password,admin.getSalt()).equals(admin.getPassword())){  
        //密码错误  
        return 1;  
    }  
    //生成Cookie  
    String ticket = UUIDUtil.uuid();  
    //Session存入redis  
    if(redisUtil.hasKey(username)){  
        //单点登录需删除用户已有Session  
        redisUtil.del((String) redisUtil.get(username));  
        redisUtil.del(username);  
    }  
    redisUtil.set(ticket,username, time: 2000);  
    redisUtil.set(username,ticket, time: 2000);  
    //写入Cookie  
    CookieUtil.setCookie(request,response, cookieName: "ticket", ticket, cookieMaxAge:  
        //登陆成功  
        return 2;  
    }
```

登陆业务Service层

Content-Length: 9  
Content-Type: text/html;charset=UTF-8  
Date: Mon, 18 Jul 2022 07:36:44 GMT  
Keep-Alive: timeout=20  
Set-Cookie: ticket=3e8074539d5e48a0b21eb8f5bbf6e887; Domain=178.233.19  
XDomainRequestAllowed: 1



The screenshot shows the Redis data browser interface. On the left, there's a tree view with 'server' expanded, showing 'db0 (2)', 'db1 (0)', 'db2 (0)', and 'db3 (0)'. Under 'db0', there are two entries: a key '1152924438' and a value '3e8074539d5e48a0b21eb8f5bbf6e887'. To the right of the tree view, there's a panel titled 'STRING' containing the value '8074539d5e48a0b21eb8f5bbf6e887'. Below this, another panel shows the key '1152924438'. A tooltip at the bottom right of the Redis interface says '键值: 大小: 10 bytes'.

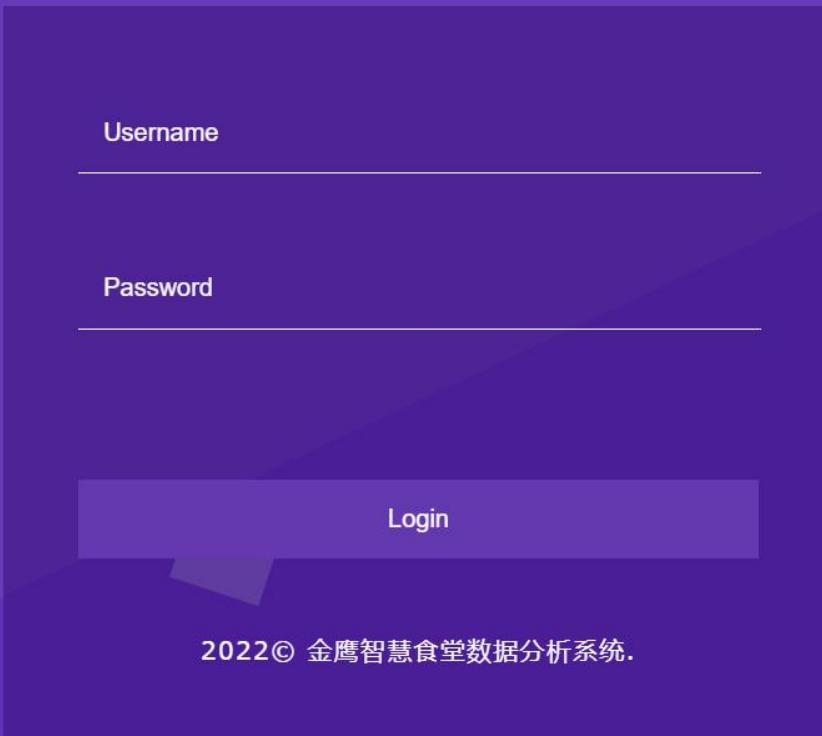
Redis缓存用户名-Session双向键值对



项目成果展示及应用



# 登录界面



# 数据可视化

AI-Restaurant Dashboard

欢迎 admin ! | 丁纪翔 1925102007 | 骆华钦 1925103027 | 陈鑫 1925103005 | 梁恺恒 1925103021 | 姜宇晨 1925102018 | 2022-07-19 09:45:20

Daily Earnings

↓ \$ 12217

-4.82% From Last Day

Daily Orders

↓ 1168

-5.12% From Last Day

Daily Visitors

↑ 499

0.40% From Last 24 Hours

Daily Avg. Rating

↑ 6.53

1.58% From Last 24 Hours

Daily Transaction Events

\$ 12326.30

Total Income

1183.20

Total Orders

495.37

Total Visitors

★ 6.51

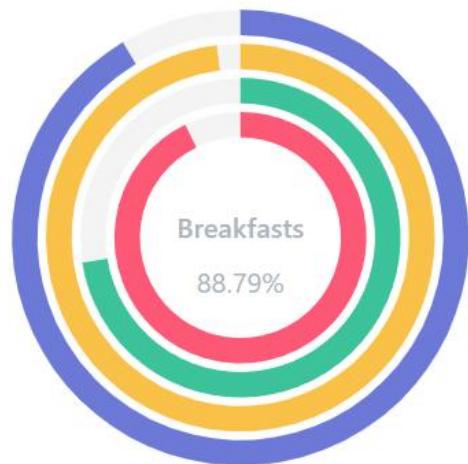
Avg. Rating



# 数据可视化

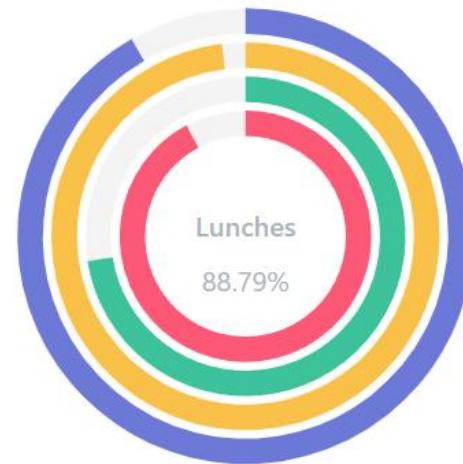
Breakfasts Nutrition Scores

411 Cal    22.54 g    25.40 g    3.25 g  
O Calorie    O Fat    O Protein    O Vitamin



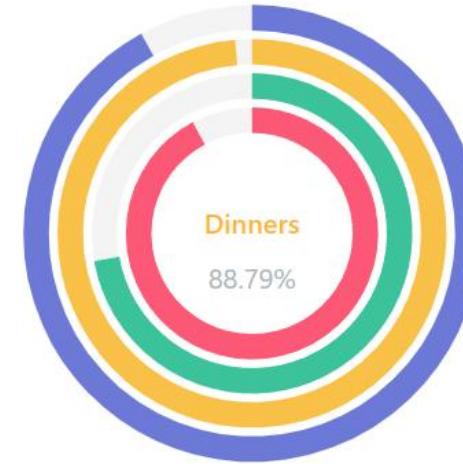
Lunches Nutrition Scores

413 Cal    22.54 g    25.41 g    3.23 g  
O Calorie    O Fat    O Protein    O Vitamin



Dinners Nutrition Scores

414 Cal    22.66 g    25.32 g    3.23 g  
O Calorie    O Fat    O Protein    O Vitamin



# 数据可视化

♥ Dishes with Popularity



💔 Least Popular Dishes



# 数据可视化

Recent Orders

Show 10 entries

Search: 蛋

UID	Order ID	Contents	Time	Consumption
20101	878992	素炒西兰花,卤蛋	2022-06-30 20:22:00	\$3.50
50125	589963	西红柿蛋汤,清蒸金针菇	2022-06-30 20:44:00	\$4.50
50731	906239	咸菜,水煮蛋	2022-06-30 20:56:00	\$2.00
57757	323707	柠檬手撕鸡,西红柿蛋汤,稀饭	2022-06-30 20:44:00	\$8.00
65320	405925	清蒸太阳蛋,煮黄豆,炒南瓜	2022-06-30 20:33:00	\$7.00
109681	192045	香煎豆腐,素菜炒蛋类,美味土豆条,地三鲜	2022-06-30 20:23:00	\$9.50
147059	751234	卤蛋,红烧鸭肉,素炒土豆丝	2022-06-30 20:39:00	\$8.50
207062	973222	素炒花菜,素炒豆芽,卤蛋,素菜炒蛋类	2022-06-30 20:54:00	\$6.00
232833	842735	青菜瘦肉粥,清蒸太阳蛋	2022-06-30 20:53:00	\$8.00
241336	836884	韭菜饼,清蒸太阳蛋,西红柿炒蛋,宫保鸡丁	2022-06-30 20:38:00	\$15.00

Showing 1 to 10 of 18 entries (filtered from 60 total entries)

First Previous 1 2 Next Last

# 总结分析期望

THIS IS JUST A PIECE OF DECORATIVE TEXT



# 总结分析期望



## 符合预期的项目结果

- 实现了菜品评分、健康饮食、订单管理、后台管理等一系列功能。
- 实现了登录功能。
- 完成了对营业额进行预测的ARIMA时间序列预测销售额算法。

## 项目需要改进的地方

- 实现多个食堂的数据分析。
- 实现三餐食物营养达标比例。
- 在网页端实时实现数据的增删改查。



感谢各位老师的指导