

技术说明书

KPI 数据解析及通报机器人

角色	姓名	日期
设计人	马风海	2018 年 8 月 14 日
审核人		
批准人		
共 22 页		

目录

一、 概述.....	1
(一) 目的.....	1
(二) 背景.....	1
二、 架构说明.....	2
三、 核心技术.....	3
(一) 数据采集.....	3
(二) 数据处理.....	9
1. MDT 数据解析.....	9
2. 数据清洗.....	10
3. 数据存储.....	10
4. 数据处理步骤.....	10
(三) 模型搭建.....	13
1. 模型定义.....	13
2. 数据建模编程.....	13
3. 结果展示.....	14
(四) 通报机器人.....	15
1. 核心代码.....	15
2. 推送结果.....	17
四、 使用操作.....	18
(一) 产品运行.....	18
(二) 使用和操作说明.....	19
1. 前端操作.....	19
2. 后端操作.....	21
五、 总结.....	22

一、概述

（一）目的

正值公司混改之际，项目积极响应公司“一切为了客户，一切为了一线，一切为了市场”经营管理理念。多元化与智能化的移动蜂窝网络覆盖模型的建立与基站分布模型优化有利于合理利用资源，提升客户满意度。繁杂数据的展示要求明了且全面的呈现方法。数据可视化可实现精简而高效地传递数据信息，从而提高工作效率和工作质量。

（二）背景

获得住宅小区覆盖率、重复覆盖率及价值估算有助于优化基站分布模型。了解住宅小区总户数和租户数有利于精准营销。例如对于租户数较多的小区可以拓展宽带业务等。单价高的小区中的居民为高价值用户潜在群体，可针对性地进行业务拓展。但是数据信息分布不集中，以传统方式整合信息难度大、数据分析不易且信息呈现不直观。

技术的发展使得可通过结合 MDT 数据分析与互联网数据，实现零散和复杂数据的分析。同时随着数据可视化被越来越多的人接受，项目可以采用数据可视化技术，借助图形化手段，清晰有效地传达信息。

此项目——《KPI 数据解析及通报机器人》便是根据这样的实际应用需求立项的。

二、 架构说明

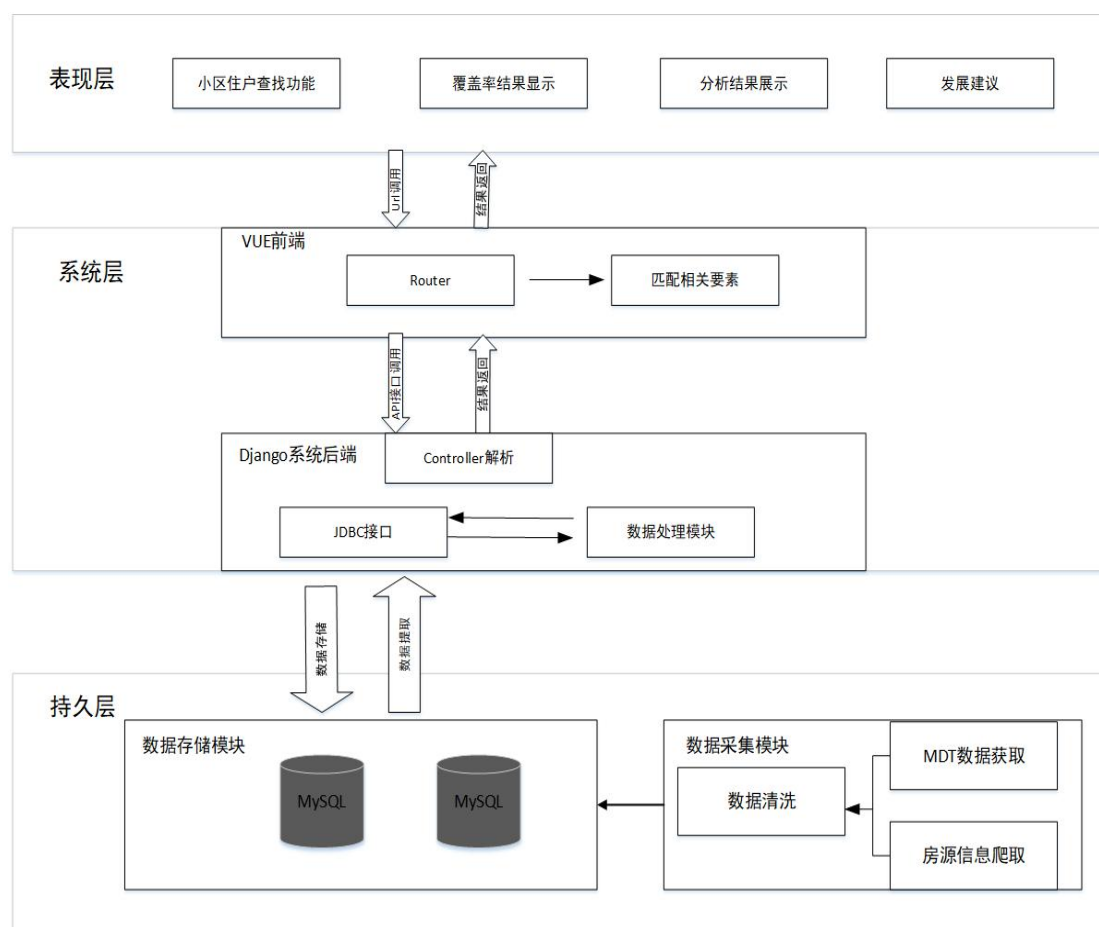


图 2.1 项目架构图

项目架构主要分为三层：持久层、系统层、表现层。持久层主要是将获取到的 MDT 数据和爬取到的房源信息进行数据清洗并存储到 MySQL 数据库中，便于实时提供数据和存储数据；系统层分为系统前端和系统后端，系统后端调用 JDBC 接口连接数据库，数据处理模块通过调用 JDBC 接口进行所需数据的获取和将处理后的数据存储到数据库中。系统层分为系统前端和系统后端，表现层通过 url 命令访问系统层，系统层 flask 系统后端接收 url 命令，通过路由将命令进行解析，从后端调用前端界面返回表现层，前端通过 ajax 请求，向后端发送数据，后端经过处理将结果返回。表现层为用户提供界面操作和结果可视化。

三、核心技术

（一）数据采集

数据采集中主要用到的是 requests 和 re 模块。

首先定义 open_url 方法：通过 requests.get(url) 向 url 网页发送 get 请求，并增添 headers 配置请求头，解决了网站的反爬虫措施，方法返回值 res 为 url 页面信息。爬虫程序都是从 res 中提取需要的信息，及小区情况。代码如下：

```
def open_url(url):
    headers = {
        "User-Agent":
            "Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 "
            "(KHTML, like Gecko) Chrome/74.0.3729.108 Safari/537.36"
    }
    res = requests.get(url=url, headers=headers)

    return res
```

我们此次爬取的信息全部来自链家网，输入网址后网页呈现画面如下：



很清楚的看到小区名、单价以及在售数量显示在网页上。点进去之后的画面如下：



我们需要的详细信息集中在点进去的网页，如：建筑年代、楼栋总数等所以一条完整的小区数据其实是在不同的两个页面上爬取。

代码如下：

```
def search_stepone(res):
    name = []
    page = []
    user = []
    stepone = []

    # 名称
    name_page = re.findall(r"<div class=\"title\">(.*?)</div>", res.text, re.S)
    for each in name_page:
        target = each.strip("\n").strip()
        if target[:2] == "<a":
            page_info = re.search(r"href=\"(.*?)\"", target).group()[6:-1]
            name_info = re.search(r">(.*?)<", target).group()[1:-1]
            name.append(name_info)
            page.append(page_info)

    # 租户
    user_info = re.findall(r"<span>(\d+)</span>套</a>", res.text)
    for each in user_info:
        user.append(each)

    for i in range(len(name)):
        stepone.append((name[i], page[i], user[i]))

    return stepone
```

在第一个页面上通过 re 正则表达式模块匹配需要的信息，及小区名，小区对应的第二个网页和租户数量三个信息；第二网页爬取代码如下：

```
def search_steptwo(page_res):
    # 详细信息字典
    info = {}
    steptwo = []
    # 价格
    price = re.findall("<span class='\"xiaoquUnitPrice\"'>(\d+)</span>", page_res.text)
    # 详细信息标题
    tabel = re.findall("<span class='\"xiaoquInfoLabel\"'>(.*?)</span>", page_res.text)
    # 详细信息内容
    content = re.findall("<span class='\"xiaoquInfoContent\"'>(.*?)</span>", page_res.text)

    for i in range(len(tabel)):
        if None is not tabel[i]:
            if None is not content[i]:
                info[tabel[i]] = content[i]
            else:
                info[tabel[i]] = ""
        else:
            info[""] = ""

    if price is not None:
        price.append("1")
        steptwo.append((price[0], info))
    else:
        steptwo.append(("1", info))

    return steptwo
```

将单价和详细信息组成一个元祖放进 steptwo 列表中返回,其中网页上将详细信息分成 Label 和 Content,所以在程序中 table 表示存入所有 Label 的列表, content 则为所有的 content。



```
<div class='\"xiaoquOverview\"'>
  ::before
  <div class='\"xiaoquImg fl\"'>...</div>
  <div class='\"xiaoquDescribe fr\"'>
    <div class='\"xiaoquPrice clear\"'>...</div>
    <div class='\"xiaoquInfo\"'>
      <div class='\"xiaoquInfoItem\"'>
        <span class='\"xiaoquInfoLabel\"'>建筑年代</span> == $0
        <span class='\"xiaoquInfoContent\"'>1992年建成 </span>
      </div>
      <div class='\"xiaoquInfoItem\"'>...</div>
      <div class='\"xiaoquInfoItem\"'>...</div>
      <div class='\"xiaoquInfoItem\"'>...</div>
    </div>
  </div>
</div>
```

再通过一个 for 循环将 Label 和 Content 一一对应保存在 info 字典中。其中可能会存在有些脏数据,如有些页面小区因为某些原因在网页上下架了,但是因为缓存的问题在第一个页面中还能显示,但点进去之后这个小区已经不存在了,以下简称为“脏数据”,例子如下:



中国铁建国际汇

90天成交1套 | 12套正在出租

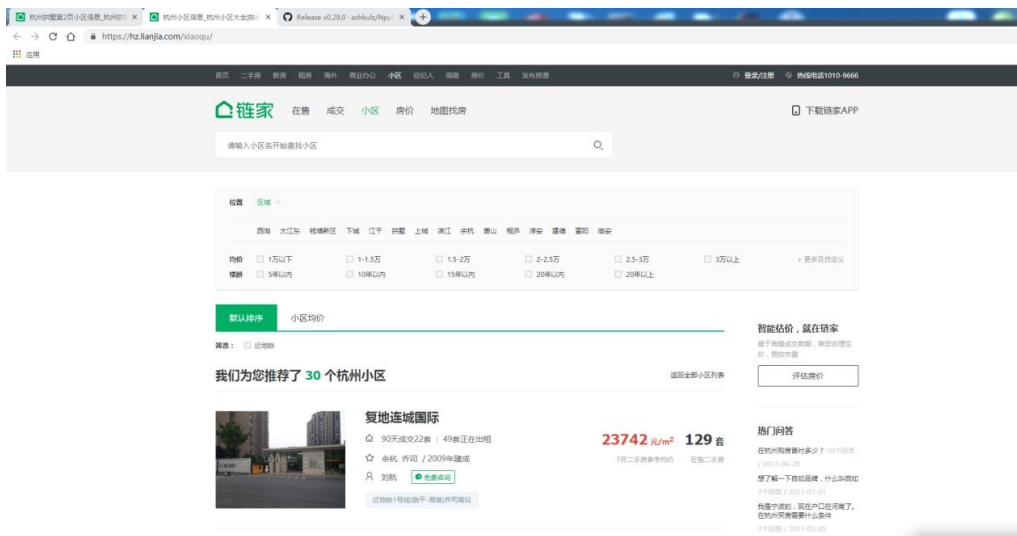
35887 元/m² 49 套

拱墅 拱宸桥 / 2003年建成

7月二手房参考均价 在售二手房

杨俊 免费咨询

点进去之后直接重定向到 <https://hz.lianjia.com/xiaoqu/> 页面：



所以做一个判断，将这些小区的 info 信息设置为空。代码如下：

```
for i in range(len(tabel)):
    if None is not tabel[i]:
        if None is not content[i]:
            info[tabel[i]] = content[i]
        else:
            info[tabel[i]] = ""
    else:
        info[""] = ""
```

将第一个页面的信息与第二个页面的详细信息合并，用 for 循环将里面的数据一一对应，如果详细信息为空说明是“脏数据”，就 continue 删选，将每组数据保存到 every_dict 中，然后将每页的数据保存在 page_list 中，并将“脏数据”过滤，代码如下：


```

for each in name_and_user:
    every_dict = {"小区名": each[0], "租户": each[2]}
    page_url = each[1]
    page_res = open_url(page_url)

    price_and_others = search_steptwo(page_res)
    for target in price_and_others:
        if target[1] == {}:
            continue

        every_dict["价钱"] = target[0]
        every_dict["建筑年代"] = target[1]["建筑年代"]
        every_dict["建筑类型"] = target[1]["建筑类型"]
        every_dict["物业费用"] = target[1]["物业费用"]
        every_dict["物业公司"] = target[1]["物业公司"]
        every_dict["开发商"] = target[1]["开发商"]
        every_dict["楼栋总数"] = target[1]["楼栋总数"]
        every_dict["房屋总数"] = target[1]["房屋总数"]
        every_dict["附近门店"] = target[1]["附近门店"]

    if len(every_dict) == 11:
        page_list.append(every_dict)
    print(every_dict)

```

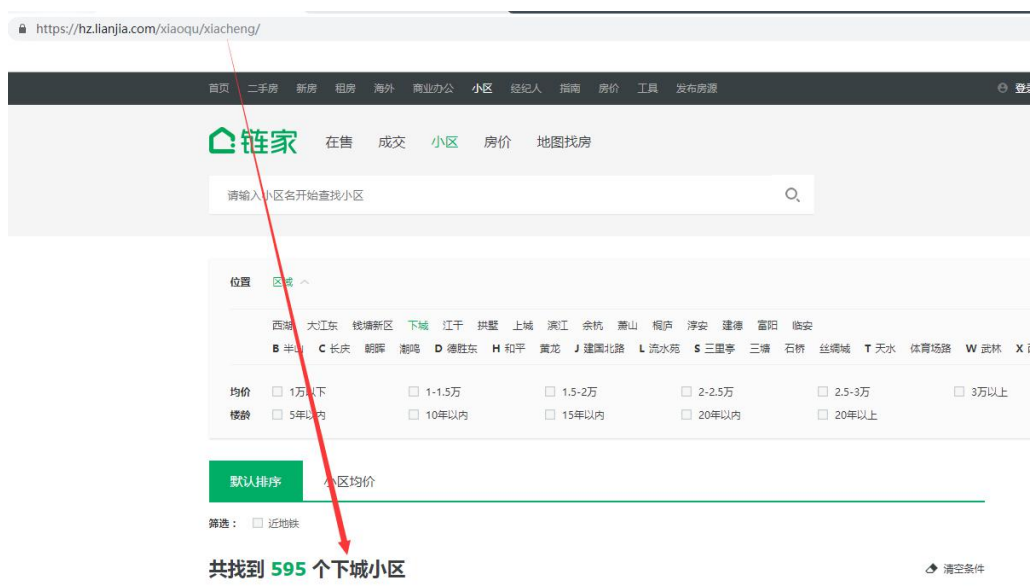
将 page_list 里的信息保存在要保存的文件中等待下一步操作：

```

with open("text.txt", "a", encoding="utf-8") as file:
    for i in range(len(page_list)):
        file.write(str(page_list[i])+"\n")

```

最后通过观察地区名在 url 中显示，所以可以将地区名作为参数传入 url。



因为地区并不多，所以将每个地区有多少页面保存在字典中，并且人性化程序，并设置睡眠，降低服务器压力。程序如下：

```
def main():
    page_list = []
    count = 0
    print(
        "\n=====n区域分别为: xihu(西湖)、qiantangxinqu(钱塘新区)、xiacheng(下城)、jianggan(江干)、"
        "gongshu(拱墅)、shangcheng(上城)、binjiang(滨江)、yuhang(余杭)、xiaoshan(萧山)、jiande(建德)、fuyang(富阳)、linan("
        "临安)\n=====")
    # 要爬取的地区
    diqu = str(input("请输入要爬取的地区拼音(如xiacheng): "))
    # 各个区域有多少页数
    city_num = {"xihu": 31, "qiantangxinqu": 8, "xiacheng": 21, "jianggan": 20, "gongshu": 19, "shangcheng": 22,
                "binjiang": 11, "yuhang": 31, "xiaoshan": 27,
                "jiande": 1, "fuyang": 8, "linan": 11}

    for page_num in range(city_num[diqu]):
        print("{}区第{}页在3秒后开始爬取...".format(diqu, page_num+1))
        sleep(3)
        print("开始爬取...\n")
        url = "https://hz.lianjia.com/xiaoqu/{0}/pg/{1}/?from=rec&tsourcetag=s_pctim_aiomsg".format(diqu, page_num)
        res = open_url(url)
        name_and_user = search_stepone(res)

        for each in name_and_user:
            every_dict = {"小区名": each[0], "租户": each[2]}
            page_url = each[1]
            page_res = open_url(page_url)

    count += len(page_list)
    if page_num != city_num[diqu]-1:
        print("{}区第{}页保存完毕，休眠10秒中...".format(diqu, page_num+1))
        sleep(7)

        page_list = []
    else:
        print("全部保存完毕!")
    print("\n爬虫任务结束，此次共爬取{}条数据".format(count))
```

执行效果展示：

```
C:\Users\lll\env4\Scripts\python.exe "D:/python 3.6.4/Hello Python1/爬虫/杭州房产信息/catch.py"

=====
区域分别为: xihu(西湖)、qiantangxinqu(钱塘新区)、xiacheng(下城)、jianggan(江干)、gongshu(拱墅)、shangcheng(上城)、binjiang(滨江)、yuhang(余杭)、xiaoshan(萧山)、jiande(建德)、fuyang(富阳)、linan(临安)
=====
请输入要爬取的地区拼音(如xiacheng): jiande
(jiande)区第1页在3秒后开始爬取....
开始爬取...

{'小区名': '御品江南', '租户': '0', '价钱': '10610', '建筑年代': '2018年建成', '建筑类型': '板楼', '物业费用': '暂无信息', '物业公司': '暂无信息', '开发商': '建德市嘉凯置业有限公司', '楼栋总数': '7栋', '房屋总数': '178户', '附近门店': '暂无门店信息'}
{'小区名': '万国源澜府', '租户': '0', '价钱': '1', '建筑年代': '暂无信息', '建筑类型': '板楼', '物业费用': '3元/平米/月', '物业公司': '杭州滨江物业管理有限公司', '开发商': '建德万国房地产开发有限公司', '楼栋总数': '7栋', '房屋总数': '722户', '附近门店': '暂无门店信息'}
{'小区名': '华宇悦府', '租户': '0', '价钱': '10074', '建筑年代': '2018年建成', '建筑类型': '板楼', '物业费用': '暂无信息', '物业公司': '暂无信息', '开发商': '建德华宇置业有限公司', '楼栋总数': '7栋', '房屋总数': '446户', '附近门店': '暂无门店信息'}

{'小区名': '御品江南', '租户': '0', '价钱': '10610', '建筑年代': '2018年建成', '建筑类型': '板楼', '物业费用': '暂无信息', '物业公司': '暂无信息', '开发商': '建德市嘉凯置业有限公司', '楼栋总数': '7栋', '房屋总数': '178户', '附近门店': '暂无门店信息'}
{'小区名': '万国源澜府', '租户': '0', '价钱': '1', '建筑年代': '暂无信息', '建筑类型': '板楼', '物业费用': '1.8元/平米/月', '物业公司': '新安江广安物业管理有限公司', '开发商': '杭州金鼎房地产开发有限公司', '楼栋总数': '9栋', '房屋总数': '117户', '附近门店': '暂无门店信息'}
{'小区名': '印象新城', '租户': '0', '价钱': '1', '建筑年代': '2015年建成', '建筑类型': '未知类型', '物业费用': '暂无信息', '物业公司': '暂无信息', '开发商': '中登房地产开发有限公司', '楼栋总数': '12栋', '房屋总数': '12户', '附近门店': '暂无门店信息'}
{'小区名': '仁和城', '租户': '0', '价钱': '11', '建筑年代': '2014年建成', '建筑类型': '板楼', '物业费用': '2元/平米/月', '物业公司': '金管家', '开发商': '杭州和谐置业有限公司', '楼栋总数': '4栋', '房屋总数': '224户', '附近门店': '暂无门店信息'}
{'小区名': '金色佳园', '租户': '0'}
{'小区名': '和谐江和城', '租户': '0', '价钱': '1', '建筑年代': '暂无信息', '建筑类型': '未知类型', '物业费用': '2.8元/平米/月', '物业公司': '暂无信息', '开发商': '杭州和谐置业有限公司', '楼栋总数': '3栋', '房屋总数': '26户', '附近门店': '暂无门店信息'}
(jiande)区第1页爬取完毕....
全部保存完毕

爬虫任务结束，此次共爬取12条数据

进程已结束，退出代码0
```

文档内容：

```
{
  "小区名": "御品江南",
  "租户": "0",
  "价钱": "10610",
  "建筑年代": "2018年建成",
  "建筑类型": "板楼",
  "物业费用": "暂无信息",
  "物业公司": "暂无信息",
  "开发商": "建德市嘉凯置业有限公司",
  "楼栋总数": "7栋",
  "房屋总数": "178户",
  "附近门店": "暂无门店信息"
},
{
  "小区名": "万国源澜府",
  "租户": "0",
  "价钱": "1",
  "建筑年代": "暂无信息",
  "建筑类型": "板楼",
  "物业费用": "3元/平米/月",
  "物业公司": "杭州滨江物业管理有限公司",
  "开发商": "建德万国房地产开发有限公司",
  "楼栋总数": "7栋",
  "房屋总数": "722户",
  "附近门店": "暂无门店信息"
},
{
  "小区名": "华宇悦府",
  "租户": "0",
  "价钱": "10074",
  "建筑年代": "2018年建成",
  "建筑类型": "板楼",
  "物业费用": "暂无信息",
  "物业公司": "暂无信息",
  "开发商": "建德华宇置业有限公司",
  "楼栋总数": "7栋",
  "房屋总数": "446户",
  "附近门店": "暂无门店信息"
},
{
  "小区名": "绿城春江明月",
  "租户": "0",
  "价钱": "1",
  "建筑年代": "2016年建成",
  "建筑类型": "未知类型",
  "物业费用": "2.8元/平米/月",
  "物业公司": "暂无信息",
  "开发商": "绿城房产",
  "楼栋总数": "12栋",
  "房屋总数": "12户",
  "附近门店": "暂无门店信息"
},
{
  "小区名": "幸福康桥水岸",
  "租户": "0",
  "价钱": "1",
  "建筑年代": "2012年建成",
  "建筑类型": "板楼/板楼",
  "物业费用": "0.8元/平米/月",
  "物业公司": "建德市新鹏物业",
  "楼栋总数": "12栋",
  "房屋总数": "12户",
  "附近门店": "暂无门店信息"
},
{
  "小区名": "水岸绿城",
  "租户": "0",
  "价钱": "1",
  "建筑年代": "2015年建成",
  "建筑类型": "板楼",
  "物业费用": "0.45元/平米/月",
  "物业公司": "暂无信息",
  "开发商": "建德市绿城房产",
  "楼栋总数": "4栋",
  "房屋总数": "224户",
  "附近门店": "暂无门店信息"
},
{
  "小区名": "国大阳光星展",
  "租户": "0",
  "价钱": "1",
  "建筑年代": "暂无信息",
  "建筑类型": "板楼",
  "物业费用": "暂无信息",
  "物业公司": "暂无信息",
  "开发商": "暂无",
  "楼栋总数": "4栋",
  "房屋总数": "224户",
  "附近门店": "暂无门店信息"
},
{
  "小区名": "香醍御苑",
  "租户": "0",
  "价钱": "1",
  "建筑年代": "暂无信息",
  "建筑类型": "板楼",
  "物业费用": "1.8元/平米/月",
  "物业公司": "新安江广安物业管理有限公司",
  "楼栋总数": "9栋",
  "房屋总数": "117户",
  "附近门店": "暂无门店信息"
},
{
  "小区名": "印象新城",
  "租户": "0",
  "价钱": "1",
  "建筑年代": "2015年建成",
  "建筑类型": "未知类型",
  "物业费用": "暂无信息",
  "物业公司": "暂无信息",
  "开发商": "中登房地产开发有限公司",
  "楼栋总数": "12栋",
  "房屋总数": "12户",
  "附近门店": "暂无门店信息"
},
{
  "小区名": "仁和城",
  "租户": "0",
  "价钱": "11",
  "建筑年代": "2014年建成",
  "建筑类型": "板楼",
  "物业费用": "2元/平米/月",
  "物业公司": "金管家",
  "开发商": "杭州和谐置业有限公司",
  "楼栋总数": "4栋",
  "房屋总数": "224户",
  "附近门店": "暂无门店信息"
},
{
  "小区名": "和谐江和城",
  "租户": "0",
  "价钱": "1",
  "建筑年代": "暂无信息",
  "建筑类型": "未知类型",
  "物业费用": "2.8元/平米/月",
  "物业公司": "暂无信息",
  "开发商": "杭州和谐置业有限公司",
  "楼栋总数": "3栋",
  "房屋总数": "26户",
  "附近门店": "暂无门店信息"
},
{
  "小区名": "御品江南",
  "租户": "0",
  "价钱": "10610",
  "建筑年代": "2018年建成",
  "建筑类型": "板楼",
  "物业费用": "暂无信息",
  "物业公司": "暂无信息",
  "开发商": "建德市嘉凯置业有限公司",
  "楼栋总数": "7栋",
  "房屋总数": "178户",
  "附近门店": "暂无门店信息"
},
{
  "小区名": "万国源澜府",
  "租户": "0",
  "价钱": "1",
  "建筑年代": "暂无信息",
  "建筑类型": "板楼",
  "物业费用": "3元/平米/月",
  "物业公司": "杭州滨江物业管理有限公司",
  "开发商": "建德万国房地产开发有限公司",
  "楼栋总数": "7栋",
  "房屋总数": "722户",
  "附近门店": "暂无门店信息"
}
```

（二）数据处理

1. MDT 数据解析

MR(Measurement Report,测量报告)是指信息在业务信道上每480ms(信令信道上470ms)发送一次数据,这些数据可用于网络评估和优化。基于传统的网络优化方法,只能通过路测、定点测试来获得用户感受信息,如网络覆盖情况、通话质量情况等,而路测和定点测试往往只能对一些主干道、重点场所进行测试,所获得的采样点数据相对于MR的用户信息要少得多,因此分析的结果存在片面性。

MR 弱覆盖: 城镇 MR 弱覆盖: RSRP<-110dbm 的采样点比例>15%的占比,定义为弱覆盖小区占比; 乡村 MR 弱覆盖: RSRP<-110dbm 的采样点比例>20%的占比,定义为弱覆盖小区占比。其中采样点比例为 MR 弱覆盖点个数除以 MR 采样总个数。

RSRP(Reference Signal Receiving Power, 参考信号接收功率): 信号接收功率,单位是: dBm。RSRP 的衡量标准如下表所示:

表 1. RSQRP 衡量标准

RSRP (dBm)	覆盖强度级别	备注
$R_x \leq -110$	覆盖强度等级 6	覆盖较差, 业务基本无法起呼
$-110 < R_x \leq -95$	覆盖强度等级 5	覆盖差, 室外语音业务能够起呼, 但呼叫成功率低, 掉话率高。室内业务基本无法发起业务。
$-95 < R_x \leq -85$	覆盖强度等级 4	覆盖一般, 室外能够发起各种业务, 可获得低速率的数据业务。但室内呼叫成功率低, 掉话率高。
$-85 < R_x \leq -75$	覆盖强度等级 3	覆盖较好, 室外能够发起各种业务, 可获得中等速率的数据业务。室内能发起各种业务, 可获得低速率数据业务。
$-75 < R_x \leq -65$	覆盖强度等级 2	覆盖好, 室外能够发起各种业务可获得高速率的数据业务。室内能发起各种业务, 可获得中等速率数据业务
$R_x > -65$	覆盖强度等级 1	覆盖非常好

RSRQ (Reference Signal Receiving Quality): 表示 LTE 参考信号接收质量, 这种度量主要是根据信号质量来对不同 LTE 候选小区进行排序。这种测量用作切换和小区重选决定的输入。

RSRQ 被定义为 $N \times \text{RSRP} / (\text{LTE 载波 RSSI})$ 之比, 其中 N 是 LTE 载波 RSSI 测量带宽的资源块 (RB) 个数。RSRQ 实现了一种有效的方式报告信号强度和干扰相结合的效果。

取值范围: $-3 \sim -19.5$, 值越大越好。

2. 数据清洗

数据清洗 (Data cleaning) - 对数据进行重新审查和校验的过程, 目的在于删除重复信息、纠正存在的错误, 并提供数据一致性。一般包含一致性检查和无效值和缺失值的处理。

3. 数据存储

数据存储对象包括数据流在加工过程中产生的临时文件或加工过程中需要查找的信息。数据以某种格式记录在计算机内部或外部存储介质上。数据存储要命名, 这种命名要反映信息特征的组成含义。数据流反映了系统中流动的数据, 表现出动态数据的特征; 数据存储反映系统中静止的数据, 表现出静态数据的特征。

4. 数据处理步骤

1) 分析得到的数据。首先对得到的 MDT 数据进行处理, 原始数据包含栅格 ID、栅格经度、栅格纬度、栅格高度、栅格 MR 总数、栅格室内 MR 数、栅格室外 MR 数、栅格弱覆盖 MR 数、栅格室内弱覆盖 MR 数 栅格室外弱覆盖 MR 数、栅格室内 MR 占比、栅格室外 MR 占比 栅格 RSRP 均值、栅格室内 RSRP 均值 栅格室外 RSRP 均值、栅格弱覆盖占比 栅格室内弱覆盖占比、栅格室外弱覆盖占比、栅格 RSRQ 均值、栅格室内 RSRQ 均值、栅格室外 RSRQ 均值、栅格室内置信度、栅格室外置信度等字段信息。如下图所示:

X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16		X16	
-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--

经过前期建模小组的建模，考虑到实际的建模需求理解，经过优化，我们将 MDT 数据进行清洗，删除与建模无关数据，留下建模有效信息。清洗后数据如下图所示：

id	栅格ID	栅格经度	栅格纬度	栅格MR总数	栅格弱覆盖MR数	栅格RSRP均值	栅格弱覆盖占比	栅格RSRQ均值
1	9	119	30	1	0	-102 0.00%		-6
2	14	119	30	1	0	-88 0.00%		-8
3	6	119	30	2	0	-99 0.00%		-6
4	23	119	30	10	0	-84 0.00%		-7
5	20	119	30	13	0	-85 0.00%		-7
6	11	119	30	1	0	-89 0.00%		-3
7	21	119	30	11	0	-85 0.00%		-7
8	15	119	30	1	0	-90 0.00%		-7
9	22	119	30	7	0	-87 0.00%		-7
10	17	119	30	8	0	-79 0.00%		-7
11	16	119	30	8	0	-84 0.00%		-6
12	18	119	30	2	0	-90 0.00%		-7
13	13	119	30	1	1	-119 100.00%		-7
14	19	119	30	4	0	-87 0.00%		-7
15	1	119	30	1	0	-88 0.00%		-3
16	10	119	30	1	0	-95 0.00%		-4
17	8	119	30	1	0	-95 0.00%		-9
18	4	119	30	1	0	-90 0.00%		-3
19	12	119	30	1	0	-106 0.00%		-4
20	7	119	30	1	0	-95 0.00%		-4
21	2	119	30	2	0	-95 0.00%		-9
22	3	119	30	1	0	-98 0.00%		-3
23	5	119	30	1	0	-98 0.00%		-3

2) 清洗爬取到的杭州小区信息。因为网站的反扒工具以及爬取过程中的不可控因素，最终获取到的小区信息存在信息不全、信息错误等可能影响项目的因素。

240	双荡弄56号	0	1	暂无信息	板楼	暂无信息	暂无信息	暂无信息	1	1	莫干山店
241	蔡马人家	0	1	暂无信息	板楼	暂无信息	暂无信息	暂无信息	3	344	永庆店
242	小河直街9	0	1	暂无信息	未知类型	暂无信息	暂无信息	暂无信息	1	1	景上公寓店
243	登云路198	0	1	暂无信息	板楼	暂无信息	暂无信息	暂无信息	1	70	和睦店
244	银润星座	0	1	暂无信息	板楼	暂无信息	暂无信息	暂无信息	1	84	风景店
245	大兜路198	0	1	暂无信息	板楼	暂无信息	暂无信息	暂无信息	1	9	湖墅嘉园店
246	湖墅北路9	0	1	暂无信息	板楼	暂无信息	暂无信息	暂无信息	1	12	湖墅店
247	新创弄10号	0	1	暂无信息	未知类型	暂无信息	暂无信息	暂无信息	1	1	湖墅店
248	乐富智汇	0	1	暂无信息	未知类型	暂无信息	暂无信息	暂无信息	2	2	万达广场店
249	湖墅南路4	0	1	暂无信息	板楼	暂无信息	暂无信息	暂无信息	1	1	湖墅店
250	德胜路146	0	1	暂无信息	板楼	暂无信息	暂无信息	暂无信息	1	1	香积寺店
251	锦文雅苑	0	1	暂无信息	板楼	暂无信息	暂无信息	暂无信息	17	1297	德信臻园店
252	星桥锦绣	0	1	暂无信息	塔楼	0.8	暂无信息	暂无信息	5	592	万达广场店
253	田園春晓	0	1	暂无信息	未知类型	暂无信息	暂无信息	暂无信息	8	1317	暂无门店信息
254	金典家园	0	1	暂无信息	未知类型	暂无信息	暂无信息	暂无信息	3	514	暂无门店信息
255	浙租大楼	0	1	暂无信息	板楼	暂无信息	暂无信息	暂无信息	1	525	米市巷店
256	湖墅南路2	0	1	暂无信息	板楼	暂无信息	暂无信息	暂无信息	1	2	米市巷店
257	招商远洋	0	1	2019	塔楼/板楼	4.1	招商局物业管理	杭州北晨	11	754	暂无门店信息
258	桃源新区	0	1	暂无信息	未知类型	暂无信息	暂无信息	暂无信息	1	5	杭钢南苑店
259	泮水人家	0	1	暂无信息	未知类型	暂无信息	暂无信息	暂无信息	5	1492	芳满庭店
260	唐尚433	0	1	暂无信息	未知类型	暂无信息	暂无信息	暂无信息	1	1	华海园店
261	登云路213	0	1	暂无信息	未知类型	暂无信息	暂无信息	暂无信息	1	1	嘉泰馨庭店
262	建华文创	0	1	暂无信息	板楼	暂无信息	暂无信息	暂无信息	1	21	运河广场新店
263	德胜巷58号	0	1	暂无信息	板楼	暂无信息	暂无信息	暂无信息	1	1	清水公寓店
264	美好国际	0	1	暂无信息	塔板结合	暂无信息	暂无信息	暂无信息	1	3	左岸花园店
265	运河ONE	0	1	暂无信息	板楼	融创物业服务集团	杭州融创尚	1	1	暂无门店信息	
266	大关苑路4	0	1	暂无信息	未知类型	暂无信息	暂无信息	暂无信息	1	1	暂无门店信息
267	大家绿城	0	1	2019	塔楼/板楼	2.8	浙江绿城物业管理	杭州绿城	11	660	永庆店

例如像上图中出现的信息缺失、房价均价为1元每平方米的极端情况。为了保证项目的顺利进行，我们首先需要将类似上图中的信息清洗删除，留下项目需要的有效信息。

3) 数据存储。首先创建名为 kpi 的数据库，数据库采用 mysql 关系型数据库，在虚拟机 linux 环境下创建，该数据库开放了远程登录，方便共同使用数据库。本次项目中，使用

navicat 图形化连接查看数据库。创建数据库图形如下：

```
CREATE DATABASE `kpi` CHARACTER SET 'utf8' COLLATE 'utf8_danish_ci';
```

在数据库中根据前期的数据清洗结果，我们主要留下了 MDT 数据表和杭州小区数据表。

创建表的结构如下图所示：

MDT 数据表字段设计：

名	类型	长度	小数点	不是 null	虚拟	键	注释
id	int	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	
栅格ID	int	255	0	<input type="checkbox"/>	<input type="checkbox"/>		
栅格经度	float	255	6	<input type="checkbox"/>	<input type="checkbox"/>		
栅格纬度	float	255	0	<input type="checkbox"/>	<input type="checkbox"/>		
栅格MR总数	int	255	0	<input type="checkbox"/>	<input type="checkbox"/>		
栅格弱覆盖MR数	int	255	0	<input type="checkbox"/>	<input type="checkbox"/>		
栅格RSRP均值	float	255	0	<input type="checkbox"/>	<input type="checkbox"/>		
栅格弱覆盖占比	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		
栅格RSRQ均值	float	255	0	<input type="checkbox"/>	<input type="checkbox"/>		

小区信息字段设计：

名	类型	长度	小数点	不是 null	虚拟	键	注释
id	int	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	
小区名	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		
租出	int	11	0	<input type="checkbox"/>	<input type="checkbox"/>		
价钱	int	11	0	<input type="checkbox"/>	<input type="checkbox"/>		
建筑年代	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		
建筑类型	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		
物业费用	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		
物业公司	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		
开发商	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		
楼总数	int	11	0	<input type="checkbox"/>	<input type="checkbox"/>		
房屋总数	int	11	0	<input type="checkbox"/>	<input type="checkbox"/>		
附近门店	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		
LATQ	float	0	0	<input type="checkbox"/>	<input type="checkbox"/>		
LNGQ	float	0	0	<input type="checkbox"/>	<input type="checkbox"/>		

最终通过数据导入后形成最终的数据库。

4) 配置数据库文件。为了方便项目中访问我们的数据，我们提前将连接数据库进行配置，数据库连接的主要参数如下：

```
USER = 'root'
PASSWORD = '123456'
HOST = 'localhost'
PORT = 3306
DB = 'kpi'
CHARSET = 'utf8'
```

（三）模型搭建

1. 模型定义

对于 KPI (Key Performance Indicator) 这个指标的建模与定义，我们主要考虑下面三个因素：

- 1) 小区住户数 N (户)。
- 2) 小区价格 P (万元)。
- 3) 小区 MR 弱覆盖率 R 。

定义 KPI 的值 K 的计算表达式如下：

$$K = \frac{P \cdot R}{N} \quad (3.1)$$

其中，弱覆盖节点数目用 M_R 表示，采样节点总数目用 M 表示，则 R 可以表达为：

$$R = \frac{M_R}{M} \quad (3.2)$$

本次项目不用考虑室内覆盖和室外覆盖的区别，只需要考虑 RSRP 和 RSRQ 的均值和平均覆盖率之间的关系。

RSRP < -110dbm (表格中 M 列)，且弱覆盖 MR 数/MR 总数 > 15% (表格中 H 列/E 列) 的数据是弱覆盖占比 100% (P 列) 的情况。

2. 数据建模编程

对于该条件的判定与数据的筛选与建模，在 pycharm 环境下利用 python 语言进行了编程，具体过程如下：

- 1) 导入 lib 库与 MDT 数据文件读取

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

df = pd.read_csv('data2.csv')
print(df)
```

Lib 库导入与 MDT 数据读取

- 2) 进行数据筛选，按照弱覆盖率情况进行分类输出结果

```

df_perfect = df[df.weakOver_rate <= 0]
print(df_perfect)
data_perfect = df_perfect.to_csv('data_perfect.csv', encoding='utf-8')

df_well = df[df.weakOver_rate <= 0.05]
print(df_well)
data_well = df_well.to_csv('data_well.csv', encoding='utf-8')

df_good = df[df.weakOver_rate <= 0.15]
print(df_good)
data_good = df_good.to_csv('data_good.csv', encoding='utf-8')

df_bad = df[df.weakOver_rate <= 0.3]
print(df_bad)
data_bad = df_bad.to_csv('data_bad.csv', encoding='utf-8')

df_worse = df[df.weakOver_rate < 1]
print(df_worse)
data_worse = df_worse.to_csv('data_worse.csv', encoding='utf-8')

df_lowest = df[df.weakOver_rate >= 1]
print(df_lowest)
data_lowest = df_lowest.to_csv('data_lowest.csv', encoding='utf-8')

```

弱覆盖率情况分类

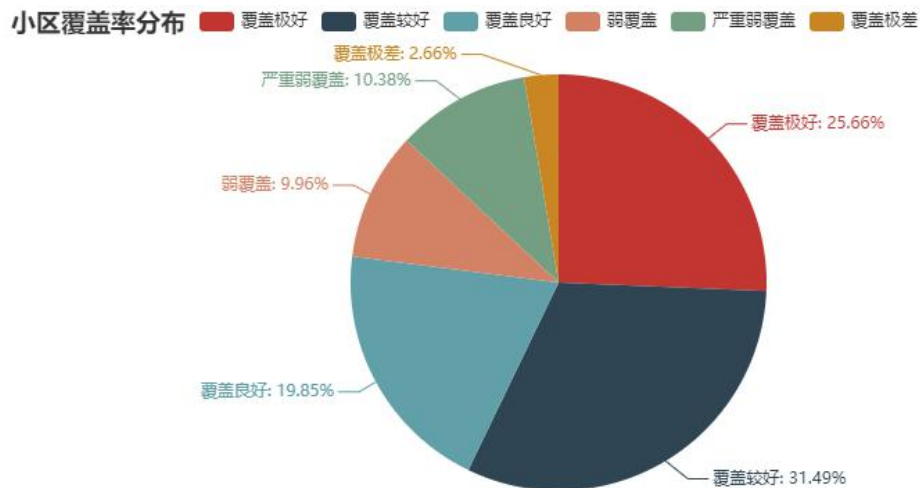
3. 结果展示

弱覆盖率分类依据与统计结果：

表 2. 弱覆盖率分类与统计结果

覆盖情况	覆盖极好	覆盖较好	覆盖良好	弱覆盖	严重弱覆盖	覆盖极差
rate 取值	0	0-0.05	0.05-0.15	0.15-0.3	0.3-1	1
分布个数	187559	230237	145148	72774	75908	19446

并由此得出小区弱覆盖分布情况如图所示：



小区弱覆盖分布情况

对于图表数据进行初步分析可知，小区的信号覆盖率在良好以上的占比超过 75%，整体覆盖情况较好。同时需要进一步结合 GPS 信息对小区覆盖分布情况进行分析，以最终得出小区信号覆盖分布模型。

（四）通报机器人

1. 核心代码

dingding.py

```
import json
import requests

WEBHOOK = 'https://oapi.dingtalk.com/robot/send?access_token=a86dfb27606676d63560637891d0e6443bbbe9f2858d065c2489747959539a7d'
TITLE = '群发消息'

def send_text_msg(message):
    headers = {'Content-Type': 'application/json; charset=utf-8'}
    data = {
        'msgtype': 'text',
        'text': {
            'title': TITLE,
            'content': message,
            'messageUrl': 'http://请求 IP+端口号'
        }
    },
```

```

        'at': {
            'atMobiles': [
                '群消息@的用户 1',
                '群消息@的用户 2'
            ],
            'isAtAll': False
        }
    }
    post_data = json.dumps(data)
    response = requests.post(WEBHOOK, headers=headers, data=post_data)
    return response.text

```

首先在钉钉中申请机器人，复制路径到 WEBHOOK。

创建钉钉调用的后端接口程序，为前端提供调用的 api。核心代码如下：

```

@web_server.route('/dingding', methods=['POST', 'GET'])
def dd():
    # message = request.form.get('msg')
    message1 = request.get_json("fugai")
    print(type(message1), message1)
    result = "大数据分析结果推送: \n\t\t 弱覆盖率 = {} \n\t\t 杭州小区价格={} \n\t\t 杭州租房户数 = {}".format(message1['fugai'],
message1['hz_price'], message1['hushu'])
    message = "后台传送数据测试!"
    print(result)
    dingding.send_text_msg(result)

```

前端通过调用 /dingding 接口并通过 ajax 向后端传输 json 数据。通过 request.get_json("fugai") 方法进行数据接收。并将数据进行解析，将值赋给 result，并将结果推送给钉钉机器人。前端调用核心代码如下：

```

function sendMessage() {
    var data = {
        "fugai": "23%",
        "hz_price": 28702,
        "hushu": 44607
    };
    $.ajax({
        method: 'post',
        url: "/dingding",
        dataType: 'json',
        data: JSON.stringify(data),
        success: function(data) {

```

```
}  
})  
}
```

2. 推送结果



钉钉大数据分析结果推送

四、 使用操作

(一) 产品运行



图 4-1 登录界面

运行 webserver 后，用户在浏览器中输入网址即可出现登录首页。如图 4-1 登录界面。图 4-2 为登录后的 KPI 数据解析及通报机器人可视化界面。



图 4-2 KPI 数据解析及通报机器人可视化界面

（二）使用和操作说明

1. 前端操作

1) 注册及登录。点击首页中的注册按钮，跳转到注册界面。在对应的输入框中输入账号及密码之后点击注册按钮实现用户注册。图 4-3 为用户注册界面。点击注册按钮完成注册后，界面会跳转至登录界面。图 4-4 为用户登录界面。在登录界面输入注册成功的账号及密码后，点击登录按钮进入 KPI 数据解析及通报机器人可视化界面。



图 4-3 注册用户界面



4-4 用户登录界面

2) 基础数据展示。KPI 数据解析及通报机器人可视化界面如图 4-5 所示。页面展示了基于住户数量的热力图、信号覆盖率分布图、杭州不同区域房价分布图等数据图表。



图 4-5 KPI 数据解析及通报机器人可视化界面

3) 电子边框显示。用户可以在城市输入框输入城市，在小区输入框输入要显示的区域。之后点击显示按钮，在地图对应的区域上会显示红色电子边框。显示效果如图 4-6 电子边框示意图。



图 4-6 电子边框示意图

4) 发送钉钉消息。点击发布钉钉消息框下的“点击发送”按钮，实现将数据发送至指定的钉钉群。操作框如图 4-7 所示。数据发送效果如图 4-8 所示。

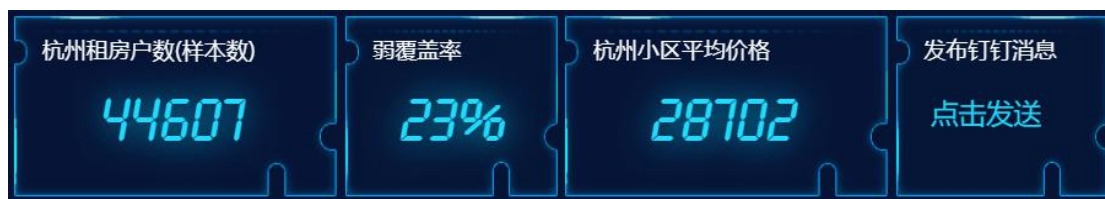


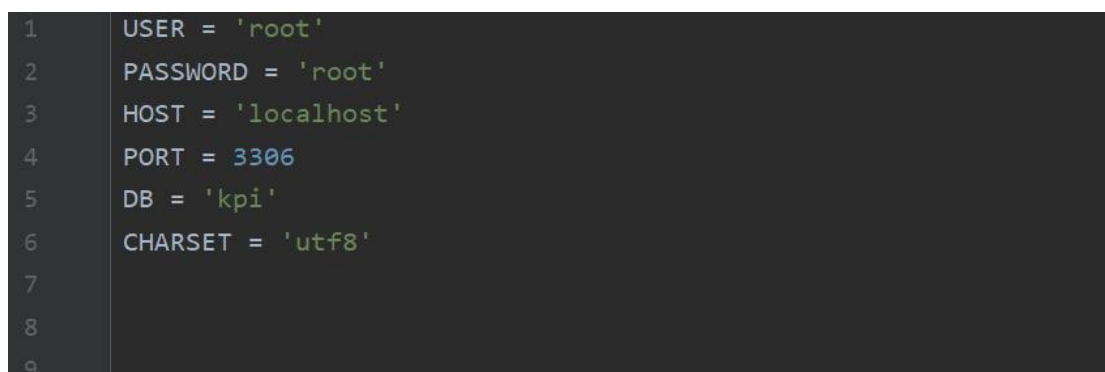
图 4-7 钉钉消息操作框示意图



图 4-8 钉钉消息发布示意图

2. 后端操作

数据库配置，如图 4-9 数据库配置界面。



4-9 数据库配置界面

Webserver 配置，如图 4-10 为 Webserver 部分配置界面。

```

1  from flask import Flask, render_template, request, redirect
2      import db_operate
3      import json
4  import dingding
5
6  web_server = Flask(__name__, static_folder='', template_folder='templates')
7  # 进入登录界面
8  @web_server.route('/', methods=['GET'])
9  def index():
10     return render_template('KPI.html')
11
12  # 进入注册界面
13  @web_server.route('/main', methods=['GET'])
14  def show_register():
15     return render_template('KPImain.html')
16

```

图 4-10 Webserver 配置界面

五、 总结

我们可以通过分析租户情况(数量)和网络覆盖率的情况,来判断用户办理业务的需求,比如说如果某个区域住户数量比较多,那么这一块的业务办理能力比较高,或者说某个地区房价较高,那么这个地区的租户消费能力较高,可以根据此类情况进行一些针对性的业务推送,再对比区域网络的覆盖率,覆盖率高说明此地区的网络信号强度较强也就是 kpi 值较高,通过两个向结合,可以指导我们布网的方向并且提高效率,也可以判断以后的业绩发展方向,我们还分析了新旧小区的租户情况,通过对比对于以后的业务上也可以深入挖掘其中的价值。同时对基站的建设 and 网络覆盖率的提高也有较好的体现。