背景意义　解决思路　解决方案　具体实现　效果展示　分析总结

# 基于大数据技术的雪球网股票数据的爬取与分析

曲容升 崔书鑫 宋俊炜 杨杰

汇报人：第九组

# CONTENT

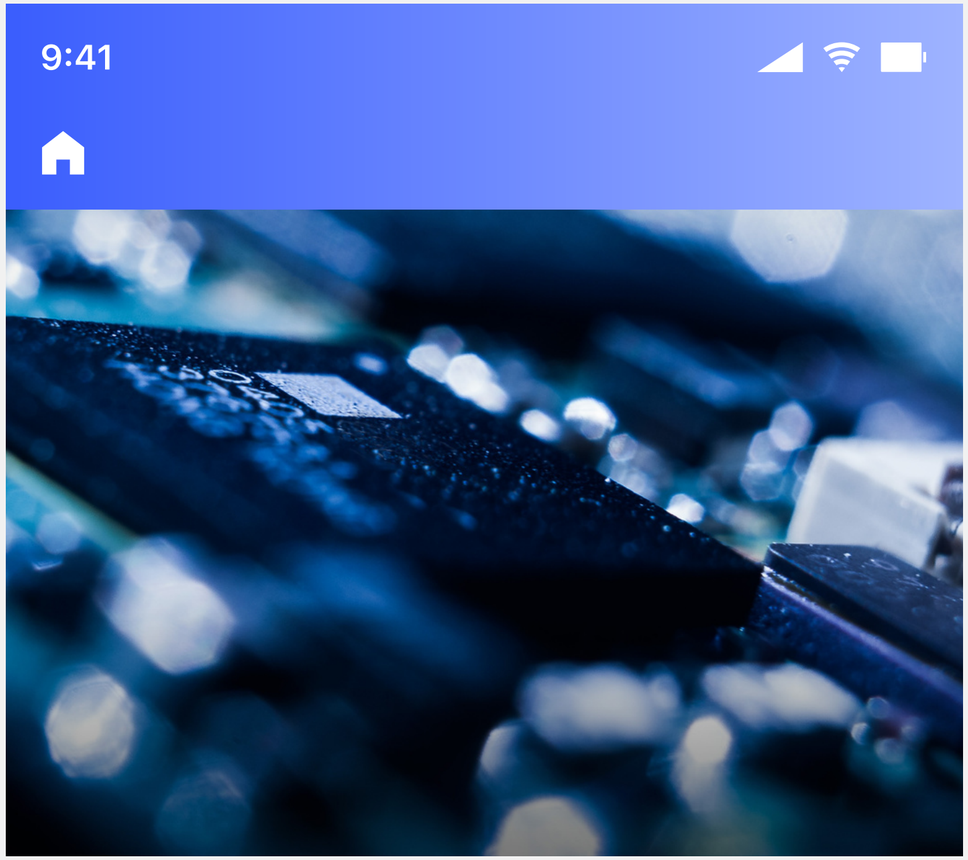
目录

# 01 背景意义

## Project Introduction

本组从获取、处理、雪球网股票信息入手，展示和分析国内电子科技企业的行情。

# 项目介绍




## 基于大数据技术的雪球网股票数据的爬取与分析

电子科技是国民经济中最重要的新兴生产部门之一，电子科技与互联网等科技的发展状况直接影响着国家经济的发展。此外， 市场环境的变化将对行业进行洗牌，只有掌握变化趋势，提前做出应对，才能让企业处于不败之地。

本组从获取、处理、雪球网股票信息入手，展示和分析国内电子科技企业的行情。

# 02 解决思路

## Project Introduction

本组从获取、处理、雪球网股票信息入手，展示和分析国内电子科技企业的行情。

# 解决思路

## 雪球网

**数据源**

项目选择雪球网（https://xueqiu.com/）作为数据源

# 解决思路

## 雪球网

**数据源**

项目选择雪球网（https://xueqiu.com/）作为数据源

## 禾迈股份等

**处理对象**

选择禾迈股份、思瑞浦电子科等公司作为分析对象

# 解决思路

## 雪球网

**数据源**

项目选择雪球网（https://xueqiu.com/）作为数据源

## 禾迈股份等

**处理对象**

选择禾迈股份、思瑞浦电子科等公司作为分析对象

## Python

**爬取工具**

scrapy进行数据爬取

# 解决思路

## 雪球网

**数据源**

项目选择雪球网（https://xueqiu.com/）作为数据源

## 禾迈股份等

**处理对象**

选择禾迈股份、思瑞浦电子科等公司作为分析对象

## Python

**爬取工具**

scrapy进行数据爬取

## jQuery+Hive+Spark+echarts

**前后端处理工具**

使用hive和spark实现对数据的分析处理；利用jQuery向后端查询数据；spark和hive计算得到的数据，进行缓存

# 03 解决方案

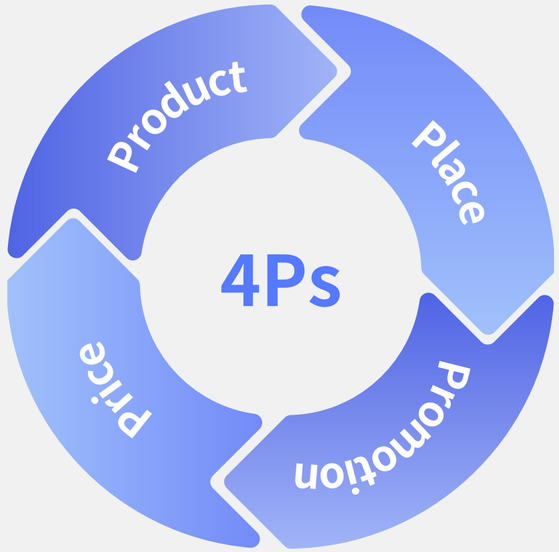
## Project Introduction

本组从获取、处理、雪球网股票信息入手，展示和分析国内电子科技企业的行情。

# 解决方案

## Python数据爬取

使用python的scrapy框架
csv文件作为缓存格式
导入hdfs数据库



# 解决方案

## Python数据爬取

使用python的scrapy框架
csv文件作为缓存格式
导入hdfs数据库


## Spark+Hive数据处理

根据粒度将数据分片
形成K线图需要的数据格式
返回前端

# 解决方案

## Python数据爬取

使用python的scrapy框架
csv文件作为缓存格式
导入hdfs数据库


## Spark+Hive数据处理

根据粒度将数据分片
形成K线图需要的数据格式
返回前端

## hdfs存储

搭建hadoop集群
hdfs存储
利用sqlite存用户信息

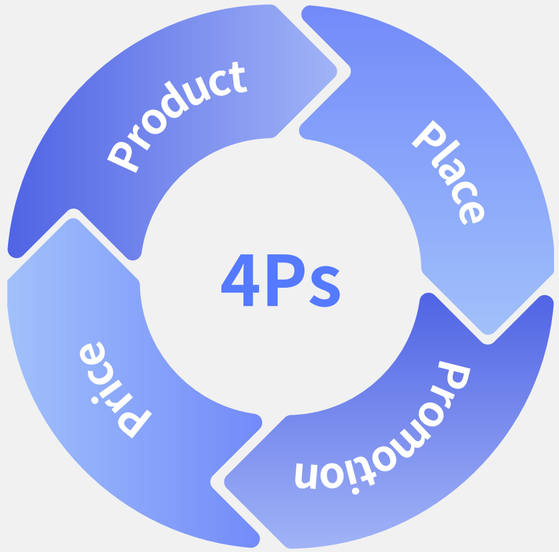
# 解决方案

## Python数据爬取

使用python的scrapy框架
csv文件作为缓存格式
导入hdfs数据库

## HTML前端

使用ECharts绘制K线图
利用jQuery向后端查询数据




## Spark+Hive数据处理

根据粒度将数据分片
形成K线图需要的数据格式
返回前端

## hdfs存储

搭建hadoop集群
hdfs存储
利用sqlite存用户信息

# 04 具体实现

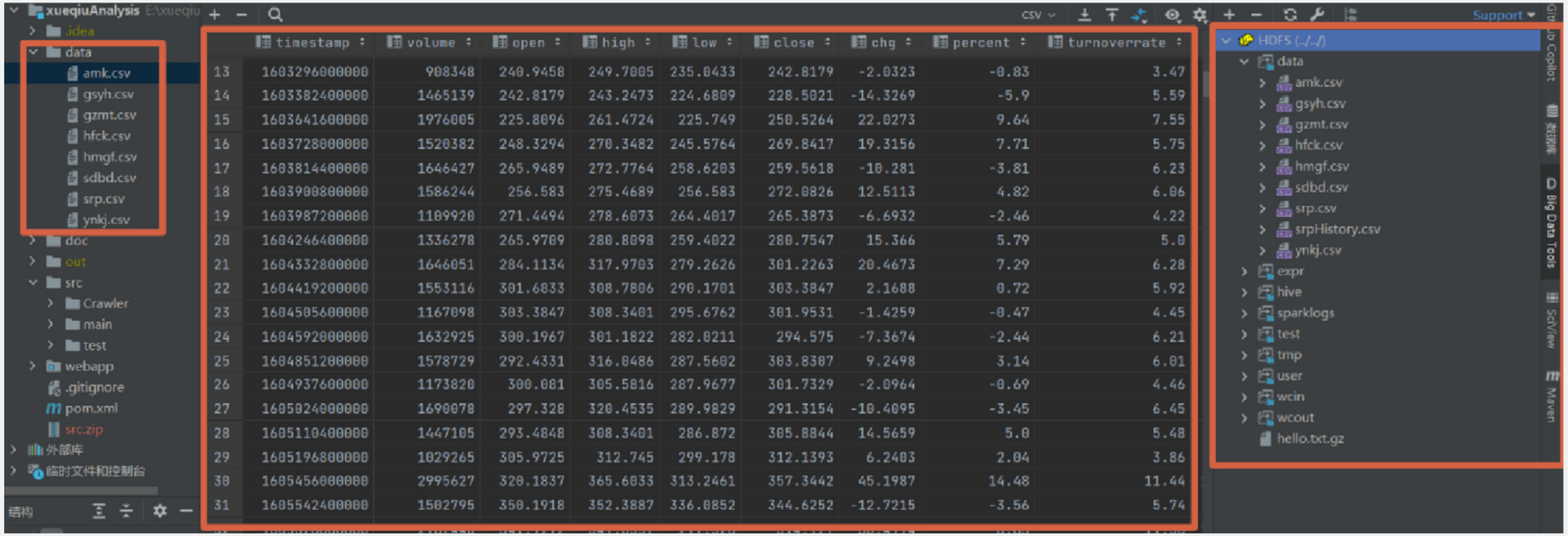
## Project Introduction

本组从获取、处理、雪球网股票信息入手，展示和分析国内电子科技企业的行情。

# Python scrapy爬虫

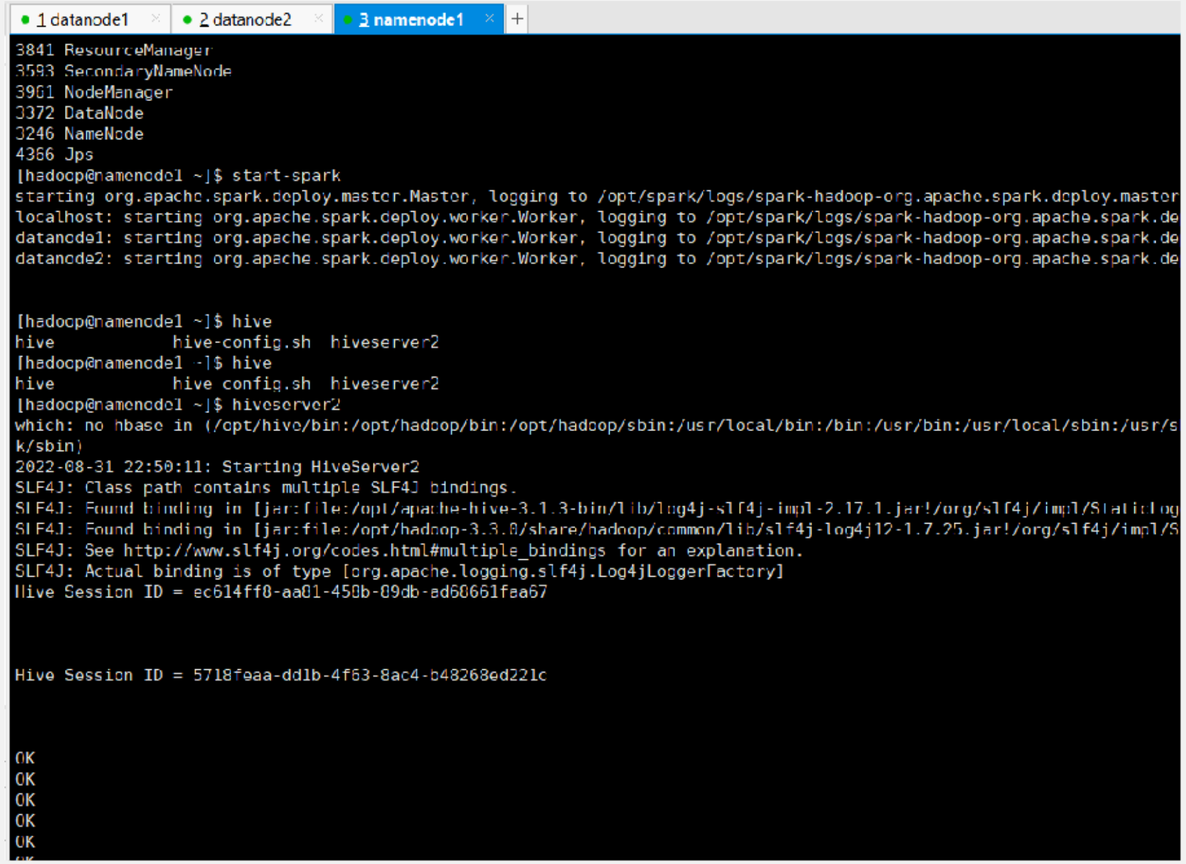
```python
class Spider_three(scrapy.Spider):
    name = "历史数据"

    def start_requests(self):
        # 爬虫url
        urls = [...]

        short_name = [...]

        def get_parse(name):
            def parse(response):
                js = json.loads(response.text)
                data = pd.DataFrame(data=js['data']['item'],
                                    columns=js['data']['column'])
                data.to_csv(f'../../data/{name}.csv', index=False)

            return parse

        for i in range(len(urls)):
            url = urls[i]
            print(url)

            yield scrapy.Request(url=url, headers=headers, cookies=cookies, callback=get_parse(short_name[i]))
```

# 数据展示




| | timestamp | volume | open | high | low | close | chg | percent | turnoverrate |
|---|---|---|---|---|---|---|---|---|---|
| 13 | 1603296000000 | 908348 | 240.9458 | 249.7005 | 235.0433 | 242.8179 | -2.0323 | -0.83 | 3.47 |
| 14 | 1603382400000 | 1465139 | 242.8179 | 243.2473 | 224.6809 | 228.5021 | -14.3269 | -5.9 | 5.59 |
| 15 | 1603641600000 | 1976005 | 225.8096 | 261.4724 | 225.749 | 250.5264 | 22.0273 | 9.64 | 7.55 |
| 16 | 1603728000000 | 1520382 | 248.3294 | 270.3482 | 245.5764 | 269.8417 | 19.3156 | 7.71 | 5.75 |
| 17 | 1603814400000 | 1646427 | 265.9489 | 272.7764 | 258.6203 | 259.5618 | -10.281 | -3.81 | 6.23 |
| 18 | 1603900800000 | 1586244 | 256.583 | 275.4689 | 256.583 | 272.0826 | 12.5113 | 4.82 | 6.06 |
| 19 | 1603987200000 | 1109920 | 271.4494 | 278.6073 | 264.4017 | 265.3873 | -6.6932 | -2.46 | 4.22 |
| 20 | 1604246400000 | 1336278 | 265.9709 | 280.8098 | 259.4022 | 280.7547 | 15.366 | 5.79 | 5.0 |
| 21 | 1604332800000 | 1646051 | 284.1134 | 317.9703 | 279.2626 | 301.2263 | 20.4673 | 7.29 | 6.28 |
| 22 | 1604419200000 | 1553116 | 301.6833 | 308.7806 | 290.1701 | 303.3847 | 2.1688 | 0.72 | 5.92 |
| 23 | 1604505600000 | 1167098 | 303.3847 | 308.3401 | 295.6762 | 301.9531 | -1.4259 | -0.47 | 4.45 |
| 24 | 1604592000000 | 1632925 | 300.1967 | 301.1822 | 282.0211 | 294.575 | -7.3674 | -2.44 | 6.21 |
| 25 | 1604851200000 | 1578729 | 292.4331 | 316.0486 | 287.5602 | 303.8307 | 9.2498 | 3.14 | 6.01 |
| 26 | 1604937600000 | 1173820 | 300.081 | 305.5816 | 287.9677 | 301.7329 | -2.0964 | -0.69 | 4.46 |
| 27 | 1605024000000 | 1690078 | 297.328 | 320.4535 | 289.9829 | 291.3154 | -10.4095 | -3.45 | 6.45 |
| 28 | 1605110400000 | 1447105 | 293.4848 | 308.3401 | 286.872 | 305.8844 | 14.5659 | 5.0 | 5.48 |
| 29 | 1605196800000 | 1029265 | 305.9725 | 312.745 | 299.178 | 312.1393 | 6.2403 | 2.04 | 3.86 |
| 30 | 1605456000000 | 2995627 | 320.1837 | 365.6033 | 313.2461 | 357.3442 | 45.1987 | 14.48 | 11.44 |
| 31 | 1605542400000 | 1502795 | 350.1918 | 352.3887 | 336.0852 | 344.6252 | -12.7215 | -3.56 | 5.74 |

# 大数据集群




Hadoop 版本 3.3.0

Hive版本 3.1.3

Spark版本 3.1.2

# Hive SQL

```
String drv = "org.apache.hive.jdbc.HiveDriver";                    //Hive驱动名称
String url = "jdbc:hive2://192.168.17.10:10000/xv_rong";            //默认端口号10000
String usr = "hive";
String pwd = "123456";
Class.forName(drv);
Connection conn = DriverManager.getConnection(url, usr, pwd);
Statement stmt = conn.createStatement();
ResultSet rs;

//创建表
String sql = "CREATE TABLE IF NOT EXISTS " + tblName + " ( " +
        "timestamps TIMESTAMP, " +
        "volume int, " +
        "opens double, " +
        "high double, " +
        "low double, " +
        "closes double, " +
        "chg double, " +
        "percents double, " +
        "turnoverrate double, " +
        "amount double, " +
        "volume_post double, " +
        "amount_post double " +
        ") " +
        "ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde' " +
        "TBLPROPERTIES ('skip.header.line.count'='1')";

stmt.executeUpdate(sql);
System.out.println("create table over");
```

# Spark SQL Standalone模式

```
if (result.get(tblName) == null) {
  println("spark start")
  val conf = new SparkConf()
    .setMaster("spark://192.168.17.10:7077")
    .setAppName("TaskTwo")
    .setJars(Seq("E:\\xueqiuAnalysis\\target\\xueqiuAnalysis-1.0-SNAPSHOT.jar")) //程序导出的 Jar包路径，根据实际情况修改
    .setIfMissing("spark.driver.host", "192.168.17.1") //设置IDEA所在机器与集群通信的网卡IP（VMnet8虚拟网卡）
    .setExecutorEnv( variable = "HADOOP_USER_NAME", value = "hadoop") //设置Hadoop环境变量，用于访问HDFS等
  //      .setExecutorEnv("SPARK_WORKER_MEMORY", "2G") //设置Executor内存，用于访问HDFS等

  val spark = SparkSession.builder().config(conf).getOrCreate() //创建 SparkSession 对象 //此处的spark是SparkSession对象，用于隐式转换

  import spark.implicits._

  val df = spark.read.option("delimiter", ",").option("header", "true").csv( path = s"hdfs://192.168.17.10:9000/data/$tblName.csv")

  df.createTempView(tblName)

  val df2 = spark.sql( sqlText = s"select timestamp, open, close, low, high from $tblName order by timestamp")
```

# 后端消息

```java
//组织json数据
try {
    int taskType = Integer.parseInt(request.getParameter( name: "task"));
    switch (taskType) {
        case 1:
            TaskOne.main(request, response);
            break;
        case 2:
            TaskTwo.main(request, response);
            break;
        case 3:
            TaskThree.main(request, response);
            break;
    }
} catch (Exception e) {
    e.printStackTrace();
    System.out.println("获取数据出错");
}
```

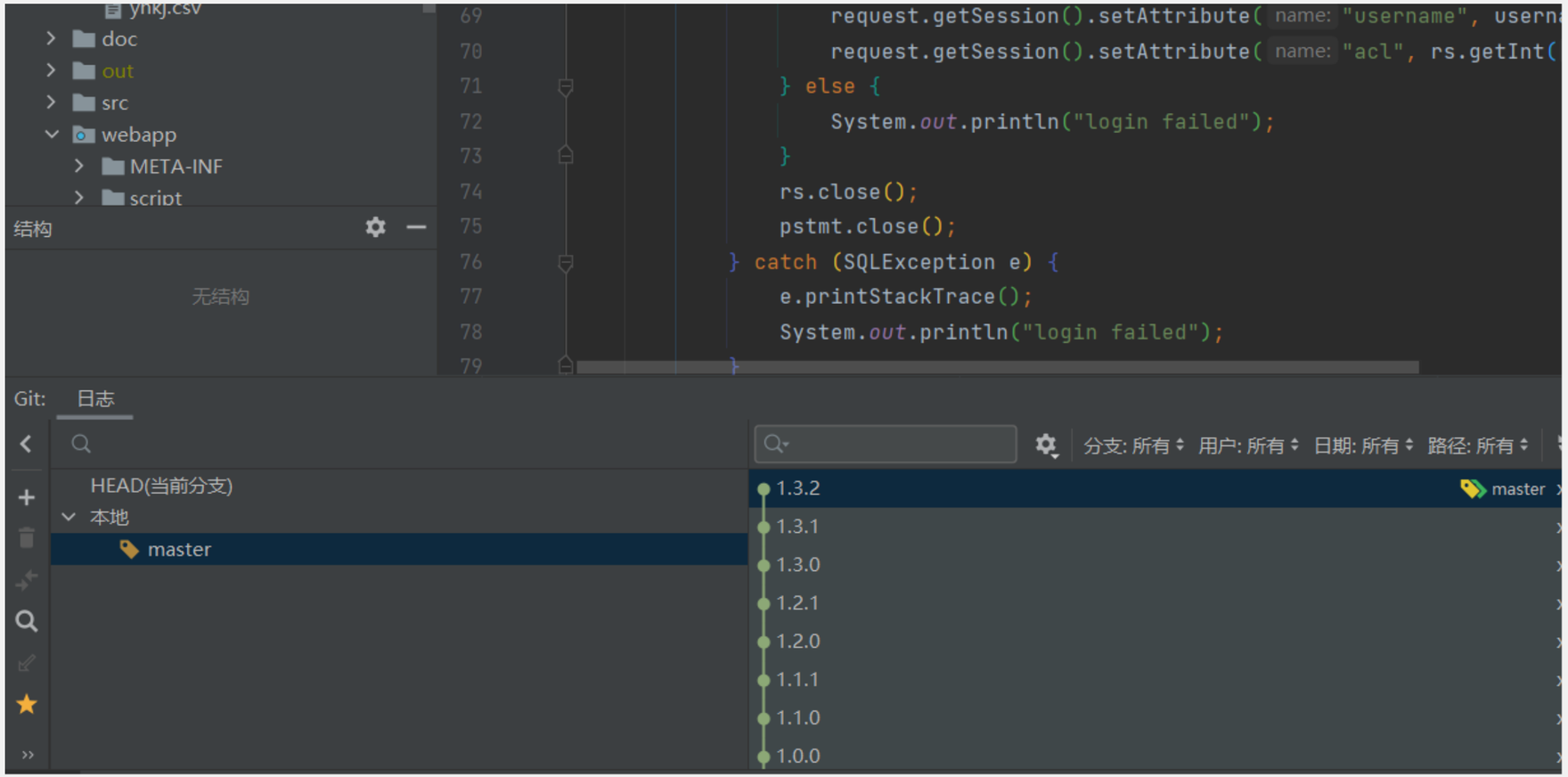
```
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
upload amk.csv to hdfs
upload gsyh.csv to hdfs
upload gzmt.csv to hdfs
upload hfck.csv to hdfs
upload hmgf.csv to hdfs
upload sdbd.csv to hdfs
upload srp.csv to hdfs
upload ynkj.csv to hdfs
Create Database OK!

[2022-08-31 11:43:00,291] 工件 xueqiuAnalysis:Web exploded: 工件已成功部署
[2022-08-31 11:43:00,291] 工件 xueqiuAnalysis:Web exploded: 部署已花费 40,366 毫秒
login failed
login failed
register successfully
```

# HTML前端

```
<body>
<!-- 为ECharts准备一个具备大小（宽高）的Dom -->
<div id="main" style="...">
<%--    顶部栏--%>
    <div id="topbar" style="..."...>
<%--侧边栏--%>
    <div id="sidebar" style="..."...>
<%--    K线图--%>
    <div id="kchart" style="..."></div>
</div>

<%--js脚本--%>
<script type="text/javascript">
    const upColor = '#ec0000';
    const upBorderColor = '#8A0000';
    const downColor = '#00da3c';
    const downBorderColor = '#008F28';

    $(function () {...});
</script>
</body>
```

```
$('#hfck').bind('click', function () {
    clickbtn('华峰测控', 'hfck', 'spark');
});

function clickbtn(name, short_name, type) {
    if (type === "hive") {
        $.getJSON("TaskServlet?task=1&tblName=" + short_name, function (ori_data) {
            let data = splitHiveData(ori_data);
            draw_k(name, data);
        })
    } else if (type === "spark") {
        $.getJSON("TaskServlet?task=2&tblName=" + short_name, function (ori_data) {
            let data = splitSparkData(ori_data);
            draw_k(name, data);
        })
    }
}
```

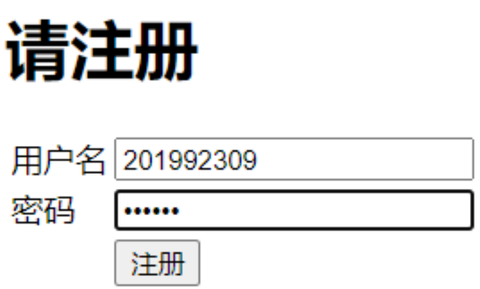
# Git版本管理


doc
out
src
webapp
META-INF
结构
无结构
request.getSession().setAttribute( name: "username", usern
request.getSession().setAttribute( name: "acl", rs.getInt(
} else {
System.out.println("login failed");
}
rs.close();
pstmt.close();
} catch (SQLException e) {
e.printStackTrace();
System.out.println("login failed");
Git: 日志
分支: 所有
用户: 所有
日期: 所有
路径: 所有
HEAD(当前分支)
本地
master
1.3.2
master
1.3.1
1.3.0
1.2.1
1.2.0
1.1.1
1.1.0
1.0.0

# 05 效果展示

## Project Introduction

本组从获取、处理、雪球网股票信息入手，展示和分析国内电子科技企业的行情。




# 注册页面


用户：202092079

# 登录成功界面


用户：202092079

# 切换vip界面


登出
渴望力量！！！
用户：2020920
贵州茅台
工商银行
爱美客
斯达半导
思瑞浦
禾迈股份
昱能科技
华峰测控
贵州茅台
日K MA5 MA10 MA20 MA30
1,940
1,927.69
1,880
1,860
1,840
1928
1876
1940
1907
1858
2022/07/28
日K
open 1,921.1
close 1,898
lowest 1,892.66
highest 1,938
MA5 1,931.8919999999998
MA10 1,939.282
MA20 -
MA30 -
2022/07/28
2022/08/02
2022/08/04
2022/08/08
2022/08/10
2022/08/12

# 普通用户界面


用户：2020
贵州茅台
工商银行
爱美客
斯达半导
思瑞浦
禾迈股份
昱能科技
华峰测控
贵州茅台
日K MA5 MA10 MA20 MA30
3,000
2,500
2,258.76
2,000
1,500
1,000
500
0
2549.25
2576
624
76
2013/05/07
日K
open 124.334
close 130.3555
lowest 124.334
highest 130.807
MA5 122.65878000000001
MA10 122.19205000000002
MA20 120.35365000000002
MA30 119.03150666666667
2011/02/16
2012/01/11
2013/05/07
2014/10/21
2015/09/14
2016/08/10
2017/07/12
2018/06/08
vip用户查询结果


贵州茅台
工商银行
爱美客
斯达半导
思瑞浦
禾迈股份
昱能科技
华峰测控
禾迈股份
日K
MA5
MA10
MA20
MA30
1,100
1,000
900
800
700
600
500
400
300
1065
730
362
1040
367.34
2022/04/27
2022/05/11
2022/05/20
2022/05/31
2022/06/10
2022/06/21
2022/06/30
2022/07/11
2022/07/20
2022/07/29
2022/08/09

# spark查询结果

# 06 分析总结

## Project Introduction

本组从获取、处理、雪球网股票信息入手，展示和分析国内电子科技企业的行情。

# 分析总结

## 分析总结

作为专业方向设计大作业，我们按照既定了流程完成了整个建模预测分析任务。我们也在这一过程中加深了对所学知识的理解，也总结了丰富的经验，这必将帮助我们在将来的路上披荆斩棘，走到更远。

## 未来展望

- 跨平台移植
- 加强UI丰富度与交互效果
- 进行分析对象迁移与拓展
- 申请著作权并在各大应用商城上架

# 分析总结

## 分析总结

作为专业方向设计大作业，我们按照既定了流程完成了整个建模预测分析任务。我们也在这一过程中加深了对所学知识的理解，也总结了丰富的经验，这必将帮助我们在将来的路上披荆斩棘，走到更远。

# 成员分工

- 曲容升 集群搭建、后端
- 宋俊炜 爬虫
- 曲容升 集群搭建、后端
- 崔书鑫 数据库、项目文档
- 杨杰 前端、演示文稿

# 分析总结

## 分析总结

作为专业方向设计大作业，我们按照既定了流程完成了整个建模预测分析任务。我们也在这一过程中加深了对所学知识的理解，也总结了丰富的经验，这必将帮助我们在将来的路上披荆斩棘，走到更远。

## 遇到的困难

- 同时引入Spark，Hive，Hadoop依赖，需要解决依赖冲突
- 雪球网有反爬机制，需要引入Cookie
- 小组成员对前端架构不熟悉，需要从零学习JavaScript和jQuery

# 感谢指导

## Thank You

汇报人第九组

曲容升-集群搭建、后端　　宋俊炜-爬虫　　崔书鑫 -数据库、项目文档　　杨杰 -前端、演示文稿