

Python 大作业实验报告

项目简介

项目名称：基金吧舆论情感走势与基金净值走势的相关检验与长线预报

项目背景：随着支付宝、东方财富网购买基金功能的完善，越来越多的大学生加入了炒基金的行列。和传统炒股、炒基金模式不同的是，现在购买基金的 APP 往往提供“评论区”功能，用户可以发帖、回帖、分享自己的见解。这就给我们提供了基金相关的舆论信息，有了进行情感走势与基金净值走势的相关检验及长期走势的回归预报的可能。

项目目标：我们小组将在各个小组成员 Level 4、5 的成果基础上，综合利用爬虫、情感分析、机器学习以及其他图形库技术，对东方财富网基金吧内舆论进行情感走势与基金净值走势的相关检验，并进行长期走势的回归预报，将成果通过图形界面展示。将基金预测结合情感分析并图形化展示是我们项目的创新点，也是重点和难点。在个人购买基金时可以使用本工具参考，具有一定的实用性。

小组成员：黄瑞轩（PB20111686）、刘良宇（PB20000180）、刘阳（PB20111677）；详细分工见报告最后部分。

开发环境：VS Code + Python 3.10.2，具体库依赖见 `requirement.txt`。

交叉学科：计算机 + 统计 + 金融

核心思路

爬虫部分

在进行爬虫前需要进行分析，本项目需要分析的网页有三处：

- 基金代码查询网页结构
- 基金吧（讨论区）网页结构
- 基金净值统计网页结构

基金代码查询网页结构

代码查询页 url：`http://fund.eastmoney.com/allfund.html`

收藏本站 | 安全登录 | 免费开户 | 忘记密码 | 手机客户端

返回天天基金网 | 基金交易 | 产品导购 | 自选基金 | 帮助中心 | 无障碍阅读 | 网站导航

天天基金网

基金代码

基金 请输入基金代码、名称或简拼

搜索

基金代码 基金公司 收藏本站

基金数据 基金净值 净值估算 基金排行 定投 港股 评级 投资工具 自选基金 比较 资讯互动 要闻 观点 学校 专题 基金交易 活期宝 指数宝 稳健理财 高端理财 基金公司 基金品种 新发基金 申购状态 分红 公告 私募 基金筛选 收益计算 账本 基金研究 策略 私募 基金吧 直播 基金超市 基金导购 收益排行 热销基金 优选基金

闲钱理财存活期宝（关联基金）
2.58%
— 最高7日年化收益(06-24) —

稳健理财
6.00%
— 业绩基准(年化) —

大家都在买的基金
45.66%
— 近1年收益率 —

看好的盘不错过
115.00%
— 近3年收益率 —

按基金代码排序

按基金公司排序

请选择基金代码开头数字：

0

1

2

3

4

5

6

7

8

9

首字

(基金代码) 基金名称	(基金代码) 基金名称	(基金代码) 基金名称
(000001) 华夏成长混合 基金吧 档案	(000003) 中海可转债债券A 基金吧 档案	(000004) 中海可转债债券C 基金吧 档案
(000005) 嘉实增强信用定期债券 基金吧 档案	(000006) 西部利得量化成长混合A 基金吧 档案	(000008) 嘉实中证500ETF联接A 基金吧 档案
(000009) 易方达天天理财货币A 基金吧 档案	(000010) 易方达天天理财货币B 基金吧 档案	(000011) 华夏大盘精选混合A 基金吧 档案
(000013) 易方达天天理财货币R 基金吧 档案	(000014) 华夏聚利债券 基金吧 档案	(000015) 华夏纯债债券A 基金吧 档案
(000016) 华夏纯债债券C 基金吧 档案	(000017) 财通可持续混合 基金吧 档案	(000020) 景顺长城品质投资混合 基金吧 档案
(000021) 华夏优势增长混合 基金吧 档案	(000024) 大摩双利增强债券A 基金吧 档案	(000025) 大摩双利增强债券C 基金吧 档案
(000028) 华富安鑫债券 基金吧 档案	(000029) 富国宏观策略灵活配置混合A 基金吧 档案	(000030) 长城核心优选混合 基金吧 档案
(000031) 华夏复兴混合A 基金吧 档案	(000032) 易方达信用债债券A 基金吧 档案	(000033) 易方达信用债债券C 基金吧 档案
(000037) 广发景宁债券A 基金吧 档案	(000039) 农银高增长混合 基金吧 档案	(000041) 华夏全球股票(QDII) 基金吧 档案
(000042) 财通中证ESG100指数增 基金吧 档案	(000043) 嘉实美国成长股票人民币 基金吧 档案	(000044) 嘉实美国成长股票美元现汇 基金吧 档案

我们可以从这个网页上爬取所有的基金代码和对应名称，这样即可本地实现搜索功能。

(090004) 大成精选增值混合 基金吧 档案	(090005) 大成货币A 基金吧 档案	(090006) 大成2020
(090007) 大成策略回报混合 基金吧 档案	(090009) 大成行业轮动混合 基金吧 档案	(090010) 大成中证
(090011) 大成核心双动力混合 基金吧 档案	(090012) 大成深证成长40ETF联接 基金吧 档案	(090013) 大成竞争价
(090015) 大成内需增长混合A 基金吧 档案	(090016) 大成消费主题混合 基金吧 档案	(090017) 大成可转债
(090018) 大成新锐产业混合 基金吧 档案	(090019) 大成景恒混合A 基金吧 档案	(090020) 大成健康产业
(090021) 大成月添利一个月滚动持有中 基金吧 档案	(090022) 大成现金增利货币A 基金吧 档案	(090023) 大成安汇
(091005) 大成货币B 基金吧 档案	(091021) 大成月添利一个月滚动持有中 基金吧 档案	(091022) 大成现金增
div.num_box 1000 × 7723 基金吧 档案	(092002) 大成债券C 基金吧 档案	(096001) 大成标普5
(100016) 富国天源沪港深平衡混合A 基金吧 档案	(100018) 富国天利增长债券 基金吧 档案	(100020) 富国天益价
(100022) 富国天瑞强势混合 基金吧 档案	(100025) 富国天时时货A 基金吧 档案	(100026) 富国天合
(100028) 富国天时时货B 基金吧 档案	(100029) 富国天成红利混合 基金吧 档案	(100032) 富国中证
(100035) 富国优化增强债券A/B 基金吧 档案	(100037) 富国优化增强债券C 基金吧 档案	(100038) 富国沪深3
(100039) 富国通胀通缩主题轮动混合A 基金吧 档案	(100050) 富国全球债券(QDII) 基金吧 档案	(100051) 富国可转债
(100053) 富国上证指数ETF联接A 基金吧 档案	(100055) 富国全球科技互联网(QDII) 基金吧 档案	(100056) 富国低碳
(100058) 富国产业债券A 基金吧 档案	(100060) 富国高新技术产业混合 基金吧 档案	(100061) 富国中国
(100066) 富国纯债债券发起A/B 基金吧 档案	(100068) 富国纯债债券发起C 基金吧 档案	(100073) 富国强回报
(100073) 富国强回报定投债C 基金吧 档案	(110001) 易方达平稳增长混合 基金吧 档案	(110002) 易方达策略

```
><li class="b"></li>
><li></li>
><li></li>
><li></li>
><li class="b"></li>
><li class="b"></li>
><li class="b">
  ><div>
    <a title="大成标普500等权重指数"
      数</a>
    " | "
    <a href="http://ijjinba.eastm
      " | "
    <a href="http://fundf10.eastm
    </div>
  </li>
</ul>
</div>
...
><div class="num_box"></div> == $0
><div class="num_box"></div>
><div class="num_box"></div>
><div class="num_box"></div>
><div class="num_box"></div>
><div class="num_box"></div>
><div class="num_box"></div>
><div class="num_box"></div>
```

该网站进行了分块，所以需要分别识别块的 xpath 和具体基金的 xpath。

- 块的 xpath: `'/html/body/div[9]/div[2]/div/div')`
- 具体基金代码和名称 xpath: `div/a/text()` , 这里需要匹配到所有的元素

基金吧网页结构

基金吧 url: `http://guba.eastmoney.com/list,of{ID}{_{Page}}.html` , 其中 `{ID}` 表示基金代码, 如果不是第一页, 则需要加上 `{_{Page}}` , `{Page}` 表示当前页号。

华安添鑫中短债A吧 040045.of | 债券型-中短债 加关注 票 购买 申购费率: 0.45% 0.045% 1折 10元起

盘中估值: **1.108** 0.06% 单位净值 (2022-06-24) 1.1074 0.01% 状态: 限大额 基金经理: 马晓璇,郑如熙 管理人: 华安基金

F10档案: 基本概况 基金经理 基金评级 历史净值 分红送配 阶段涨幅 季度涨幅 持仓明细 行业配置 资产配置 基金公告 规模变动 持有人结构 基金费率 更多 +

全部 公告 | 查看关于华安基金公司的全部讨论 排序: 评论时间 发新帖

阅读	评论	标题	作者	最后更新
4671	3	讨论 好礼悬赏 新能源车vs光伏, 你更看好谁?	建信基金	06-25 14:22
146	0	公告 华安添鑫中短债债券型证券投资基金(华安添鑫中短债A)	基金资讯	06-13 09:08
169	0	公告 华安添鑫中短债债券型证券投资基金招募说明书更新	基金资讯	06-13 09:05
304	0	公告 华安添鑫中短债债券型证券投资基金托管协议更新	基金资讯	06-11 09:43
207	0	公告 华安添鑫中短债债券型证券投资基金基金合同更新	基金资讯	06-11 09:43
194	0	公告 关于调整华安添鑫中短债债券型证券投资基金基金费率并	基金资讯	06-11 09:36
875	0	和王心凌一样再度翻红! 短债基金有何魅力?	华安基金	05-26 08:30
323	0	公告 华安添鑫中短债债券型证券投资基金2022年第一季度报告	基金资讯	04-21 11:38
402	0	公告 华安添鑫中短债债券型证券投资基金2021年年度报告	基金资讯	03-30 19:45
486	0	公告 华安添鑫中短债债券型证券投资基金(华安添鑫中短债A)	基金资讯	03-04 09:14
500	0	公告 华安添鑫中短债债券型证券投资基金招募说明书更新	基金资讯	03-04 09:07
639	0	公告 关于华安添鑫中短债债券型证券投资基金暂停大额申购、	基金资讯	01-28 07:48
585	0	公告 华安添鑫中短债债券型证券投资基金2021年第四季度报告	基金资讯	01-21 09:33
1227	0	21年债券回报不错, 22年机遇仍在?	iGeek异行妄想	01-01 17:19

基金估值

开启净值估算须知

净值估算是按照基金历史定期报告公布的持仓和指数走势预测当天净值, 预估数值不代表真实净值, 仅供参考, 实际涨跌幅以基金净值为准。

立即开启

代码简称拼音 净值 查询

历史净值

净值日期	单位净值	累计净值
2022-06-24	1.1074	1.3775
2022-06-23	1.1073	1.3774
2022-06-22	1.1071	1.3773
2022-06-21	1.1070	1.3772
2022-06-20	1.1069	1.3771

查看更多

基金吧内容展示是通过 request 请求可以直接获取的, 所以只需要定位我们需要的数据的 xpath 信息, 即可拿到原始数据。同理分析, 提取我们感兴趣的数据项有:

- 最大页号 xpath: `//*[@id="articlelistnew"]/div[82]/span/span/span/span`
- 标题 xpath: `//*[@id="articlelistnew"]/div[{num}]/span[3]/a/text()` , `num` 是帖子序号。
- 发帖时间 xpath: `//*[@id="articlelistnew"]/div[2]/span[5]` , 这里的 2 处为帖子位置。

基金净值网页结构

爬基金净值的网页 url: `https://www.dayfund.cn/fundvalue/{ID}.html` , 其中 `{ID}` 表示基金代码。

基金净值网页内容是动态加载的, 通过抓包解析可发现如下 post 内容:

```
▼ jQuery18303393265759350257_1654438283384({Data: {,...}, ErrCode: 0, ErrMsg: null, TotalCount: 1709, Expansion: null, PageSize:
  ▼ Data: {,...}
    Feature: "020,050,051,054"
    FundType: "001"
    ▶ LSJZList: [{FSRQ: "2022-05-05", DWJZ: "1.0918", LJJZ: "2.8079", SDATE: null, ACTUALSYI: "", NAVTYPE: "1",...},...]
    SYType: null
    isNewType: false
    ErrCode: 0
    ErrMsg: null
    Expansion: null
    PageIndex: 2
    PageSize: 20
    TotalCount: 1709
```

只要通过 request 库发起一个 post 请求，更改不同的 `PageIndex` 即可获得不同页面。同时，对于每次请求的结果，需要定位我们感兴趣数据的 xpath 信息。我们感兴趣的数据项有：

- 日期 xpath: `//*[@id="his_nav_table"]/tbody/tr[1]/td[1]`
- 增长率 xpath: `//*[@id="his_nav_table"]/tbody/tr[1]/td[4]`

信息收集、统计分析和预报模块

为了代码的可维护性以及代码的易用性，小组成员分工合作。一方进行模块化的封装，并编写恰当的接口；一方根据模块编写上层逻辑。

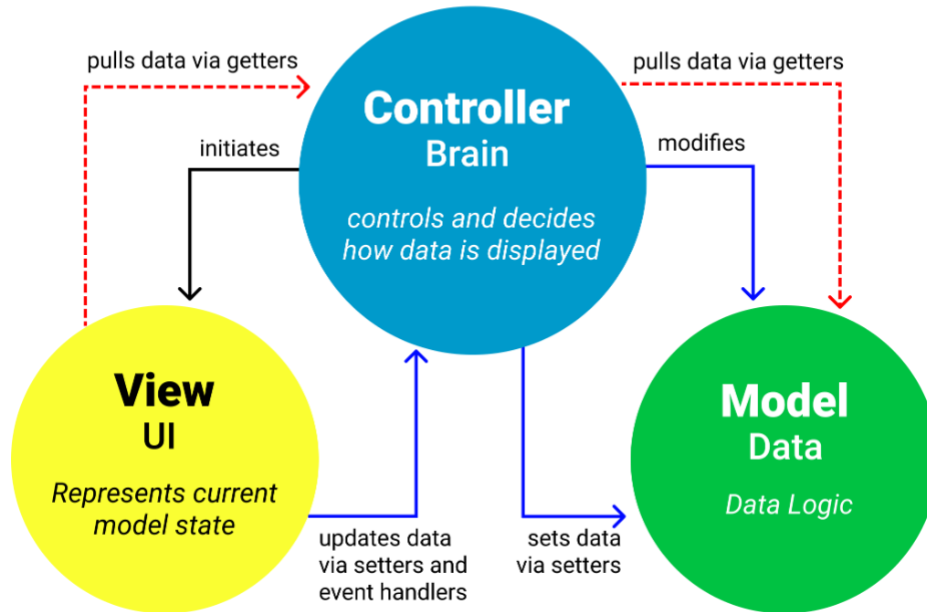
- **数据采集和预处理模块：**
 - 数据获取：需要对获取 Http 请求的功能进行封装，同时，为了数据可持久化的需求以及便于程序各模块之间的数据交互，还需要将数据保持在 CSV 文件中。为了获取讨论区每一面的帖子，本项目实现了 `talk_bar` 类用于讨论区数据获取的封装；为了实现 CSV 文件的读写，本项目封装了 CSV 的读写类。
 - 金融用词处理：金融领域包含许多专业词汇，而且基民还创建了一些如“打板”、“抄底”、“吃面”这样的特殊词汇，基金吧中获取的评论文本具有不规范、口语化等不利于模型训练的特点，常用的分词工具并不能提供准确的分词结果，所以需要搜集网络上的金融领域名词和基民自创词汇，在送入 Transformers Pipeline 之前就进行替换，以达到对基金吧评论高准确度情感分类的目的。
 - 数据清洗：对获取到的数据需要进行数据清洗，主要有两方面。首先是基金吧里有一些非评论的帖子，比如【*好礼悬赏新能源车vs光伏，你更看好谁？*】这种是基金公司发出来的活动帖，一般在置顶位置，需要排除；其次是一些灌水的文本，如同一用户短时间内的大量相同标题行为，所以需要在每个 CSV 组中进行查重合并。
- **情感分析模块：**Transformers 情感分析库、有道翻译 API
 - 情感分析：用 Transformers 的 `sentiment-analysis` 流水线工具可以以一个字符串为输入，得到一个表明 POSITIVE 或 NEGATIVE 的 `label` 和指示得分的 `score`。这个数据对于本项目来说非常重要。
 - 翻译：考虑到 Transformers 对英文的分析准确度较高，本项目通过向有道翻译 API 发起 request 请求的方式来获取翻译文本，提高情感分析准确度。
- **周围信息计算模块：**Matplotlib 库、Numpy 库、Jieba 库、Pyecharts 库
 - 评论区的舆论情感倾向总结（看涨、看跌的分布饼状图）
 - 评论区的舆论词云图（所用停用词表是对 github 上四个停用词表的整合）
 - 评论区的舆论情感分时变化曲线
 - 基金净值的分时变化曲线、以及预报的净值曲线
- **长线预报模块：**Transformers 库、Numpy 库
 - 需要说明的是，根据小组对现有工作的调研及请教相关学者，使用单一经济学模型往往预测周期很长，对于短期的市场波动不能及时的反应，并且预测精度也不高，使用单一的神经网络模型就存在过拟合和模型可解释性弱的问题。因此本项目将根据更长的历史数据进行长线预测，所用模型为北京邮电大学提出的 **ARIMA-LSTM** 的组合预测模型，该模型能够充分利用基金价格中的线性部分和非线性部分，提高长线预测的准确度并减少误差。

GUI 交互及数据流

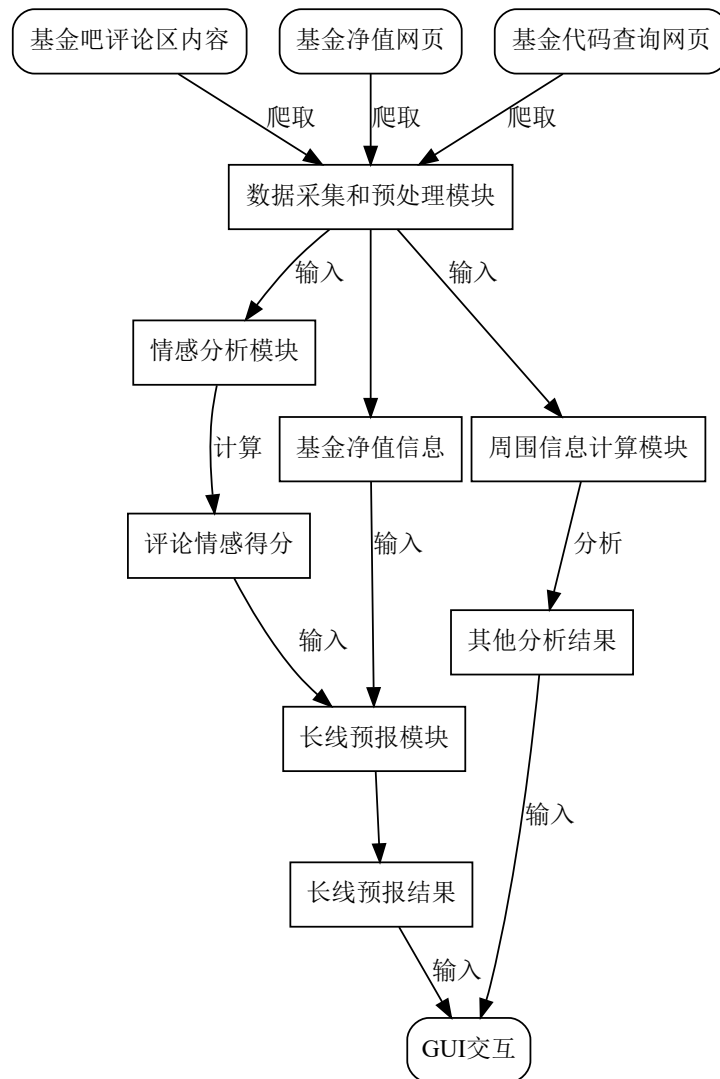
本项目利用 PyQt5 实现了 GUI 交互，便于用户使用。具体运行截图可参见本文的运行截图部分。

本项目借鉴了软件工程中的 MVC 模式，对核心功能函数进行了抽象封装，图形界面基本只需要调用这些核心函数。

MVC Architecture Pattern



具体模块关系和数据流转图如下所示：



- 数据采集和预处理模块: `url.py`、`mycsv.py`
- 情感分析模块: `analysis.py`、`translate.py`
- 周围信息计算模块: `cloud.html`、`mplcanvas.py`、`analysis.py`
- 长线预报模块: `analysis.py`、`predict.py`
- GUI交互: `gui.py`

运行截图

爬取信息



如上展示的是输入名称查询对应基金代码功能。

CSV 数据存储

这里展示 CSV 数据存储文件格式：

讨论区数据，包含日期，标题：

```
1 date,title
2 02-07 21:31,大家今天回血了多少？（2月7日基金复盘）
3 02-07 21:28,2.7收评：基建，新能源，白酒，医药，半导体，煤炭基金分析
4 02-07 21:28,明天肯定跌。
5 02-07 21:27,明天危险了。
6 02-07 21:21,#虎年祝福#虎年运虎头--钱吹吹神吹来临
7 02-07 21:14,慢慢来小仓位打其他原先白酒基金看今天净值高于估值1-2个点，看来是抛
8 02-07 21:08,2.07后市看法及操作策略：白酒、医疗、新能源车！
9 02-07 21:07,大爷
10 02-07 21:05,节后A股第一波反弹能持续多久？附中证500分析
11 02-07 21:04,1月28日大头个人基金操作策略
12 02-07 20:46,【车多桩少！春节返乡路上补能难题再现，新能源汽车的续航焦虑何解？】
13 02-07 20:37,原来所谓的基金经理，什么学历的经理都是靠懵，靠运气，垃圾
14 02-07 20:32,收评|市场的下跌已步入“尾声”，2月或将迎农历新年行情开启？市场行情回暖
15 02-07 20:29,象样地做个人吧
```

基金历史涨跌数据，包含日期，净值，净值变化：

1	date,total,rate
2	2022-06-24,2.9306,1.49%
3	2022-06-23,2.9128,0.94%
4	2022-06-22,2.9017,-1.22%
5	2022-06-21,2.9163,-0.46%
6	2022-06-20,2.9218,2.03%
7	2022-06-17,2.8978,2.15%
8	2022-06-16,2.8729,-0.29%
9	2022-06-15,2.8763,1.33%
10	2022-06-14,2.8611,0.90%
11	2022-06-13,2.8500,-2.16%

情感分析综合预测

四个小分区：分别是按照基金代码和面数查询、舆论历史记录（红色表示分析为积极的，绿色表示分析为消极的）、情感得分曲线和饼状图以及词云图。



三种词云图：



并且还可根据爬取的文件进行词频分析，并生成图片格式报告，比词云图更加直观。这里用到了 PIL 库。

看看这些热聊的关键词

[illegible]

\$招商中证白酒指数(LOF)A(OTCFUND|161725)\$2022.5.2

67

\$白酒基金LOF(SZ161725)\$昨天割了，是真

39

美股三大指数集体重挫，纳指跌超4.7%，A股今天怎么走？

■■■■■■■■■■■ 36

5.25操作贴：白酒、医疗、半导体、新能源可以加仓吗？

35

5.25操作贴：白酒、医疗、半导体、新能源可以加仓吗？

■■■■■■■■■■■■■■■■■■■■■ 32

5.25白酒、医疗、半导体、基建板块分析

■■■■■■■■■■■ 26

美股三大指数集体重挫，纳指跌超4.7%，A股今天怎么走？

■■■■■■■■■■■ 25

5.25操作贴：白酒、医疗、半导体、新能源可以加仓吗？

■■■■■■■■■■■ 23

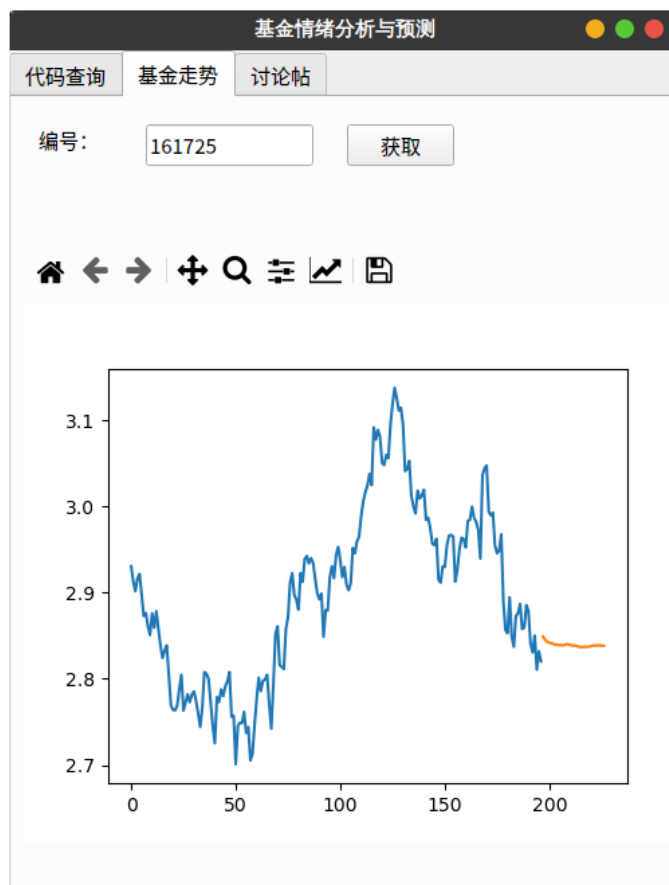
5.25白酒、医疗、半导体、基建板块分析

■■■■■■■■■■ 22

百万基金实盘：小高理财5月25日投资分享，震荡行情这样做！

USTC@Python大作业 PB20111686 黄瑞轩 & PB20000180 刘良宇

如图，蓝色部分为历史记录，橙色部分为长线预报结果。



实现细节

这里挑选本项目有代表性的一些代码进行讲解。

图形界面

实现多个 Tab 的切换：

```
class Window(QWidgets.QTabWidget):
    def __init__(self):
        super().__init__()
        self.setWindowTitle("基金情绪分析与预测")
        self.resize(500, 600)
        self.query_page = QtWidgets.QWidget()
        self.trend_page = QtWidgets.QWidget()
        self.post_page = QtWidgets.QWidget()
        self.setup_data()
        self.setup_query_page()
        self.setup_trend_page()
        self.setup_post_page()
        self.setup_tab_widget()

    def setup_data(self):
        self.all_fund = all_fund()

    # 查询基金标签页
    def setup_query_page(self):
        .....

    # 基金趋势标签页
    def setup_trend_page(self):
        .....

    # 讨论版分析标签页
    def setup_post_page(self):
        .....
```

```
def setup_tab_widget(self):
    .....
```

实现 Qt 展示 Matplotlib 并可以动态交互:

```
class MplCanvas(FigureCanvasQTAgg):

    def __init__(self, parent=None, width=5, height=4, dpi=100):
        fig = Figure(figsize=(width, height), dpi=dpi)
        self.axes = fig.add_subplot(111)
        super(MplCanvas, self).__init__(fig)

class Window(QMainWindow):
    def __init__(self):
        """....."""

    def setup_trend_page(self):
        """....."""
        sc = MplCanvas(self.trend_page, width=5, height=4, dpi=100)
        toolbar = NavigationToolbar(sc, self.trend_page)
        layout = QVBoxLayout()
        layout.addWidget(toolbar)
        layout.addWidget(sc)
        widget = QWidget(self.trend_page)
        widget.setLayout(layout)
        widget.move(0, 100)
```

爬虫爬取

封装与 html 文档的交互:

```
# request获得html对象
def get_html(_url):
    return requests.get(url=_url, headers=headers)

# 获得html源文字
def get_html_text(_url):
    return requests.get(url=_url, headers=headers).text

# 获得html源文字
def get_html_text_post(_url, num:str):
    return requests.post(url=_url, headers=get_header(num)).text

# 创建xpath可读取的对象
def get_xpath_etree(_html):
    if _html[0:3] == '\r\n\r\n':
        _html = _html[4:]
    return etree.HTML(_html)

# 根据xpath获取内容
def get_content_by_xpath(_etree, _xpath):
    return _etree.xpath(_xpath)
```

针对爬取的不同网页类型封装不同的类, 通过 requests 请求爬取, lazy 用于缓存:

```
class all_fund:
    url = "http://fund.eastmoney.com/allfund.html"

    def __init__(self) -> None:
        try:
            self.lazy = get_xpath_etree(get_html_text(self.url))
        except:
            print("Error! Check network.")
```

```

# [(code, name), ...]
def query(self, s: str) -> list[tuple[str, str]]:
    try:
        result = []
        pages = get_content_by_xpath(self.lazy, '/html/body/div[9]/div[2]/div/div')
        for page in pages:
            funds = page.xpath('ul/li')
            for fund in funds:
                """... 尝试获取"""
        return result
    except:
        return None

```

翻译与情感分析

我们针对基金价格和讨论帖分别约定了 CSV 格式，因此根据格式不同封装成类读取：

```

class analysier:
    an = analysis.sentiment()
    tr = translate.Translate()

    def __init__(self, id: str):
        self.id = id

    def get_interested_files(self, usage: str, path='./') -> list[str]:
        """....."""

    def set_data(self) -> numpy.ndarray:
        """....."""

    def set_jz(self) -> numpy.ndarray:
        """....."""

    def set_jz_simple(self) -> numpy.ndarray:
        """....."""

    # 日期 映射到 [(标题, 情感), ...]
    def trans_tk(self) -> dict[str, list[tuple[str, float]]]:
        """....."""

```

针对相应的文件，可以给出分析结果：

```

# 日期 映射到 [(标题, 情感), ...]
def trans_tk(self) -> dict[str, list[tuple[str, float]]]:
    # 先创建所有基金净值出现的天对应的字典
    # {std_date: np.array}, np.array中收录了这一天所有的舆论得分信息，positive为正，negative为
    负。
    filename_list: list = self.get_interested_files('rate')
    interested_dict: dict[str, list[tuple[str, float]]] = {}
    # interested_dict.get(f'{date.today()}_sa', None)
    try:
        for filename in filename_list:
            with open(filename, 'r', newline='') as f:
                reader = csv.DictReader(f)
                for row in reader:
                    _date: str = convert_date(row['date'])
                    interested_dict[_date] = []
                break # 只用最新的数据

    # 打开所有舆论文件
    filename_list: list = self.get_interested_files('talk')
    for filename in filename_list:
        with open(filename, "r", newline='') as f:
            reader = csv.DictReader(f)
            for row in reader:

```

```

        print('-----')
        print(f'now processing {filename} row {row}')
        std_date = convert_date(row['date'])
        if std_date not in interested_dict:
            print(f'{std_date} is not in dict!')
            continue
        std_talk = row['title'] # 中文
        result = self.tr.t(std_talk)
        print(f'success translated, sleep 0.1s')
        label = self.an.get_label(result)
        score = self.an.get_score(result)
        print(f'{label}, {score}, {result}')

        interested_dict[std_date].append(std_talk)
        if label is True:
            interested_dict[std_date].append((std_talk, score))
        else:
            interested_dict[std_date].append((std_talk, -1 * score))

    with open('./log', "w+") as f:
        f.write(interested_dict.__repr__())
    return interested_dict

except:
    with open("./log", "w+") as f:
        f.write(interested_dict.__repr__())
    return interested_dict

```

这里针对可能出现的错误进行了 `try` - `except` 用于处理

翻译另外用到了翻译类，这里调用了有道翻译的 api:

```

class Translate:
    def __init__(self) -> None:
        pass

    # api 用户信息加密
    def encrypt(self, signStr):
        hash_algorithm = hashlib.sha256()
        hash_algorithm.update(signStr.encode('utf-8'))
        return hash_algorithm.hexdigest()

    # api 签名验证
    def truncate(self, q):
        if q is None:
            return None
        size = len(q)
        return q if size <= 20 else q[0:10] + str(size) + q[size - 10:size]

    # api 请求
    def do_request(self, data):
        headers = {'Content-Type': 'application/x-www-form-urlencoded'}
        return requests.post(YOUDAO_URL, data=data, headers=headers)

    # api 翻译
    def connect(self, word):
        data = {}
        data['from'] = 'zh-CHS'
        data['to'] = 'en'
        data['signType'] = 'v3'
        curtime = str(int(time.time()))
        data['curtime'] = curtime
        salt = str(uuid.uuid1())
        signStr = APP_KEY + self.truncate(word) + salt + curtime + APP_SECRET
        sign = self.encrypt(signStr)
        data['appKey'] = APP_KEY
        data['q'] = word
        data['salt'] = salt

```

```
data['sign'] = sign

response = self.do_request(data)
contentType = response.headers['Content-Type']
ans = response.json()
return ans["translation"][0]
```

趋势预报

使用了 `numpy` 等科学计算库加速, 使用 `transformer` 的模型, 抽象出如下接口等:

```
def get_exp_seires(score: float):
    """..."""

def get_inner_product(score: float, array: np.ndarray, tag: bool = True) -> float:
    """..."""

def computeCost(X, y, theta):
    """..."""

def predict(score: float, array: np.ndarray, tag: bool = True) -> float:
    """..."""
```

项目总结与展望

过去对于我们小组成员来说, “爬虫”只是一个经常听到的概念。通过这个项目, 我们亲身实践了网络爬虫, 并掌握了网络爬虫的使用逻辑以及编程范式。作为一个大作业, 这个项目包含了完整的 `概念 -> 流程图 -> 实际模块建立 -> 上层逻辑` 的全过程, 令小组成员收获颇丰。

作为交叉学科的应用, 我们从自己的生活实际出发, 尝试从基金讨论区挖掘可能的对我们有用的数据。我们也了解到现在有如 Transformers 等这样的机器学习框架可供方便的使用, 扩展了我们的眼界。同时在调研的时候我们也了解到一开始提出的 ARMA 模型的局限性, 转而进行长线的宏观预测。我们认为本项目对类似基金涨跌这样长线有规律短期有较多波动的问题具有积极的探索意义。

未来如果有机会, 我们将进一步优化本项目给出的趋势信息, 努力使其更加直观, 并考虑给出投资策略及其在相应数据集的验证。

本项目的代码统计如下, 满足 `requirement.txt` 的依赖之后即可在线获取数据进行实时分析测试。

```
$ python3 gui.py
```

代码统计

```
├─ analysis.py
├─ color_gen.py
├─ gui.py
├─ mpl_canvas.py
├─ mycsv.py
├─ painter.py
├─ predict.py
├─ requirement.txt
├─ stop_words.txt
├─ translate.py
└─ url.py
```

行数统计:

Language	files	blank	comment	code
Python	9	114	74	813

项目分工

黄瑞轩 (PB20111686) 完成了爬虫的分析和实现（数据采集和预处理模块）、情感分析模块（Transformers 测试、相应模块封装等）以及长线预报概念和词频分析实现。（40%）

刘良宇 (PB20000180) 完成了周围信息计算模块和 GUI 交互部分，组织模块（上层逻辑）以及长线预报的实现。（40%）

刘阳 (PB20111677) 完成了前期调研、代码的调试以及最终测试。（20%）

实验报告撰写：黄瑞轩、刘良宇

实验报告审阅：刘阳

2022.06.25 @USTC 面向交叉学科的 Python 程序设计与跨学科实践 大作业报告
(完)