**目录**

[1 概要设计 3](#_Toc98457565)

[1.1 处理流程 3](#_Toc98457566)

[1.2 总体结构设计 3](#_Toc98457567)

[1.3 功能设计 4](#_Toc98457568)

[1.4 用户界面设计 4](#_Toc98457569)

[1.5 数据结构设计 9](#_Toc98457570)

[1.6 接口设计 9](#_Toc98457571)

[1.6.1 数据接口 10](#_Toc98457572)

[1.6.2 算法接口 10](#_Toc98457573)

[1.6.3 小程序接口 10](#_Toc98457574)

[1.7 系统配置策略 10](#_Toc98457575)

[1.8 系统部署方案 10](#_Toc98457576)

[2 详细设计 11](#_Toc98457577)

[2.1 期货百科功能模块 11](#_Toc98457578)

[2.1.1 功能描述 11](#_Toc98457579)

[2.1.2 性能描述 11](#_Toc98457580)

[2.1.3 输入 11](#_Toc98457581)

[2.1.4 输出 11](#_Toc98457582)

[2.1.5 程序逻辑 11](#_Toc98457583)

[2.1.6 限制条件 12](#_Toc98457584)

[2.2 期货数据功能模块 12](#_Toc98457585)

[2.2.1 功能描述 12](#_Toc98457586)

[2.2.2 性能描述 12](#_Toc98457587)

[2.2.3 输入 12](#_Toc98457588)

[2.2.4 输出 12](#_Toc98457589)

[2.2.5 程序逻辑 13](#_Toc98457590)

[2.2.6 限制条件 13](#_Toc98457591)

[2.3 财经快讯功能模块 13](#_Toc98457592)

[2.3.1 功能描述 13](#_Toc98457593)

[2.3.2 性能描述 13](#_Toc98457594)

[2.3.3 输入 13](#_Toc98457595)

[2.3.4 输出 13](#_Toc98457596)

[2.3.5 程序逻辑 14](#_Toc98457597)

[2.3.6 限制条件 14](#_Toc98457598)

[2.4 数据分析功能模块 14](#_Toc98457599)

[2.4.1 功能描述 14](#_Toc98457600)

[2.4.2 性能描述 14](#_Toc98457601)

[2.4.3 输入 14](#_Toc98457602)

[2.4.4 输出 14](#_Toc98457603)

[2.4.5 程序逻辑 15](#_Toc98457604)

[2.4.6 限制条件 15](#_Toc98457605)

[3 详细开发 16](#_Toc98457606)

[3.1 小程序开发 16](#_Toc98457607)

[3.2 后端开发 19](#_Toc98457608)

[3.3 内网穿透 23](#_Toc98457609)

[3.4 算法开发 24](#_Toc98457610)

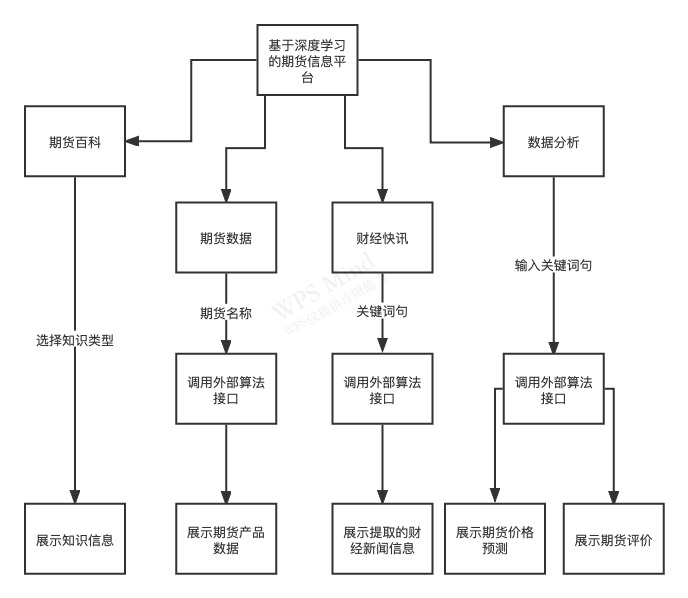
# 概要设计

期货产品相较于股票投资，涉众人数较少，其很大一部分原因在于客户在期货交易中，市场价格波动所带来的不确定因素。这种价格波动给客户带来交易[盈利](https://baike.baidu.com/item/%E7%9B%88%E5%88%A9)或损失的风险。因为杠杆原理的作用，风险极度放大，从而导致期货市场起点较高，且需要的实时动态信息资源丰富。本产品着眼于期货市场，希望通过给予专业的实时新闻信息提取分析、动态的数据查询和可视化还有理论上的预测，为用户入手期货市场提供稳定的信息保障。

## 处理流程

用户在用户界面通过选择不同的四个功能模块，分别进入本产品的四个主要不同功能。用户可以通过选择已有关键词或者通过手动输入自己感兴趣的关键词，系统获取到用户点击或输入的关键词后，提交给后端算法服务器后返回该功能模块提供的分析内容。

## 总体结构设计

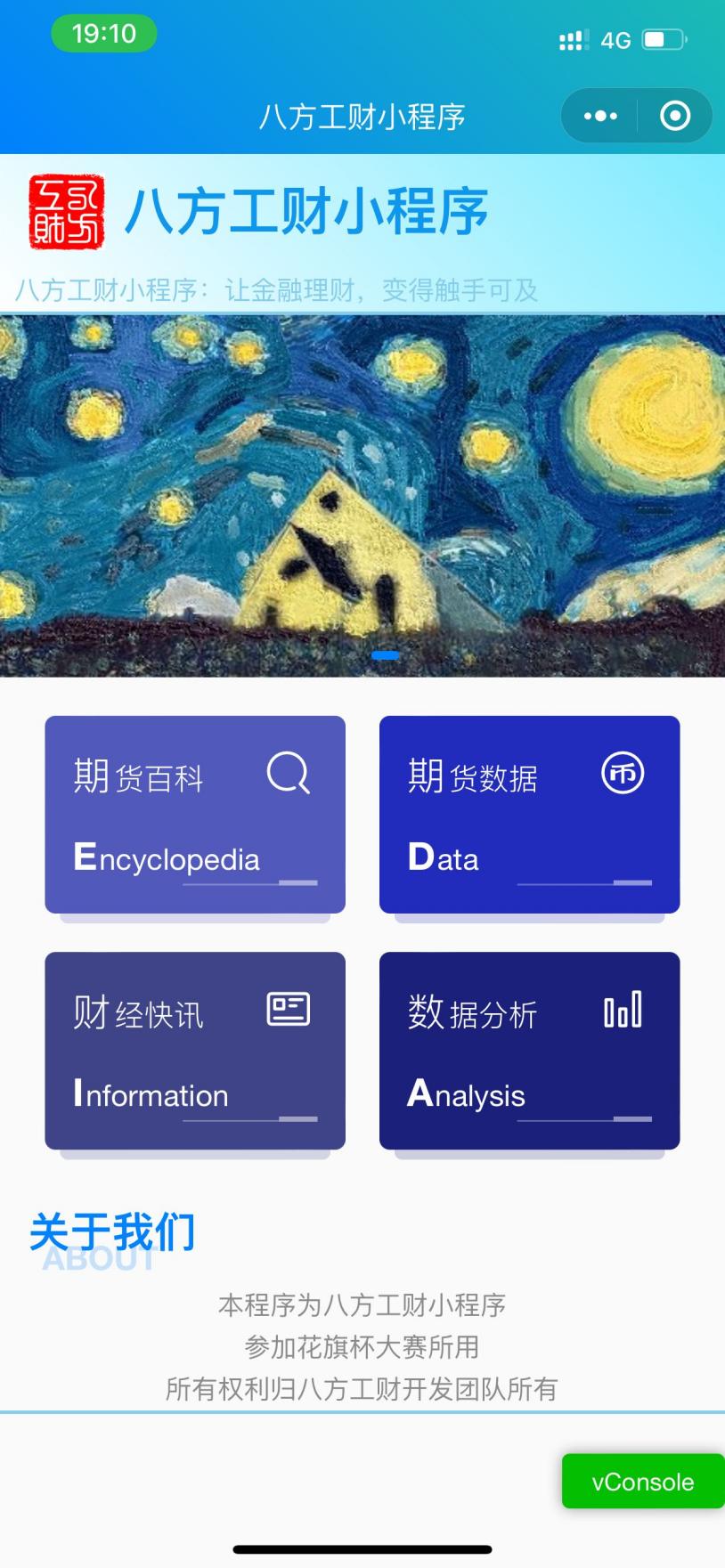


## 功能设计

主要分为四个功能模块。模块一期货百科，提供给定的基本期货知识关键词给用户进行选择，在用户选定后可以返回该知识在不同来源中的信息情况。模块二为期货数据模块，通过用户给定关键词，从而返回给定关键词对应的期货产品数据。模块三为财经快讯模块，通过用户给定的关键词输入来返回当前时段各主要财经新闻网站中涉及到的新闻信息。模块四为数据分析模块，通过用户给定关键字的方式返回指定产品的各项期货数据评估指标，并提供价格预测的功能。

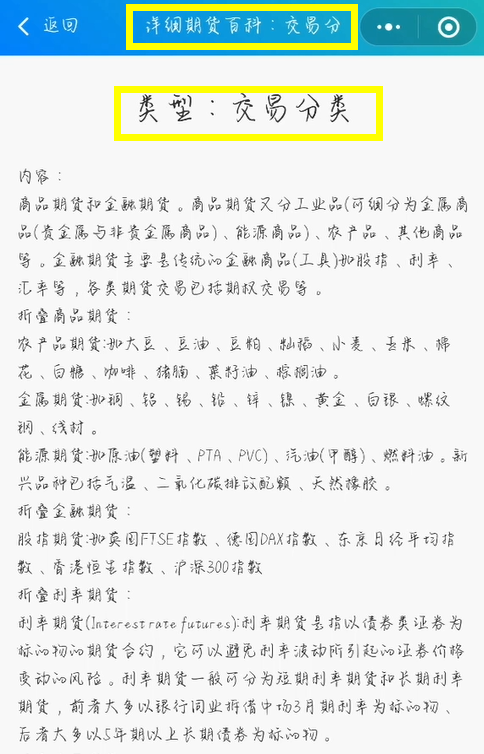
## 用户界面设计

小程序初始界面设计：



期货百科界面设计：



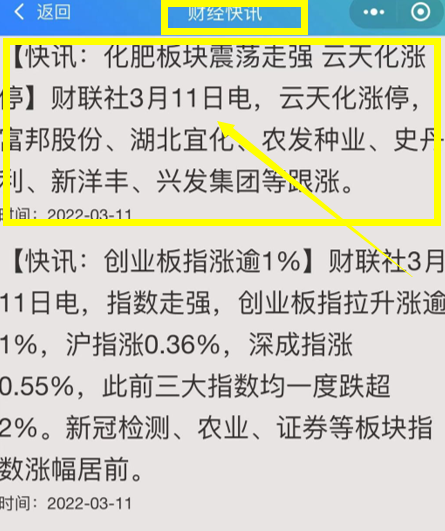


期货数据界面设计：



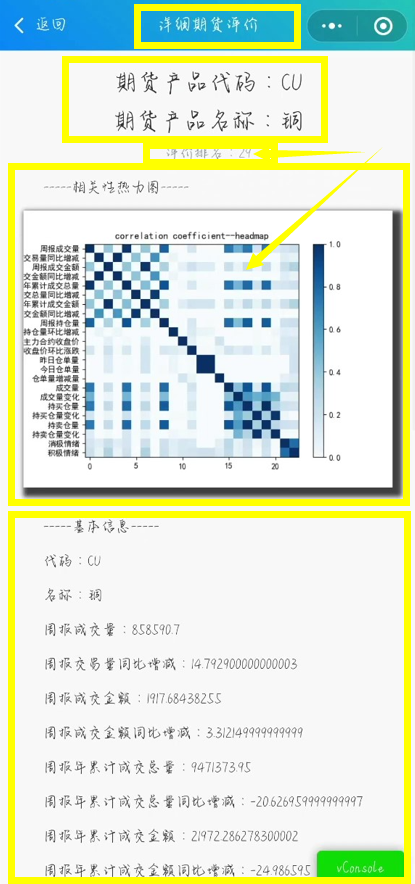
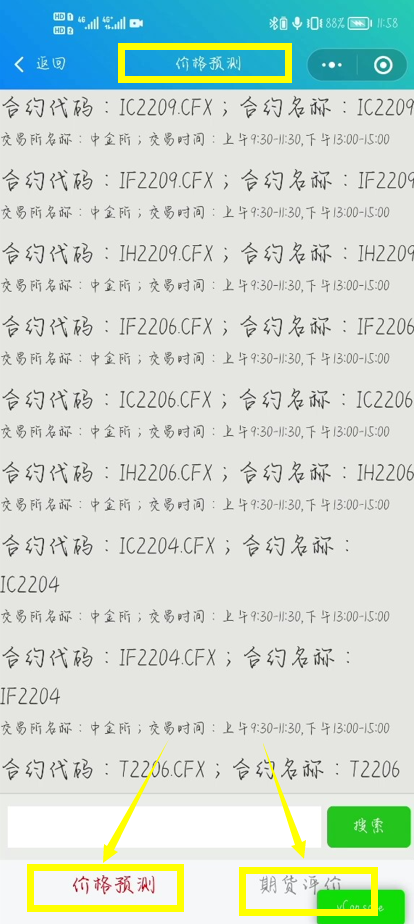


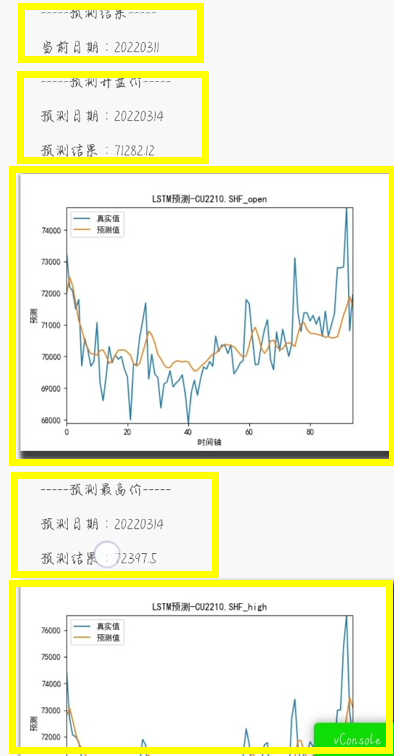
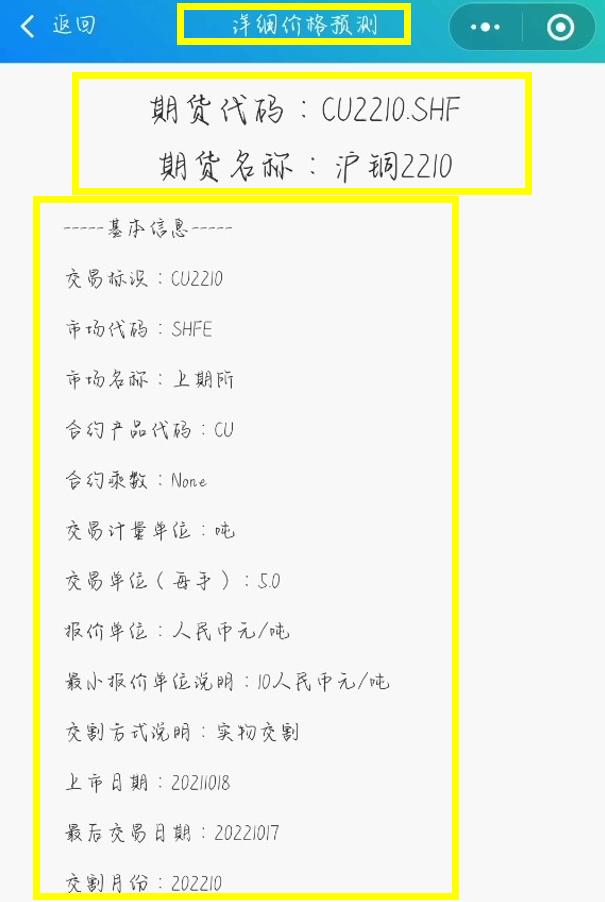
财经快讯界面设计：





数据分析界面：





## 数据结构设计

在该项目的开发过程中，我们主要使用的数据结构有以下几种：

* 字符串：

我们的项目中，生成的图片的文件名和URL，都是使用字符串进行数据的传递。文本参数的传递也是通过字符串进行传递。

* 数组：

在我们的项目中，大部分获取并返回的数据，都是通过LIST存储，以便直接通过下标按顺序进行读取。并且LIST的数据较为规整，便于直接转化为文件进行存储。

* JSON数据：

我们的后端采用FLASK框架进行搭建，因此所有的数据都会以LIST进行封装，并转化为JSON包，通过FLASK的API接口进行传输和获取。

## 接口设计

我们的最终作品主要包含了三种类型的接口：数据接口、算法接口、小程序接口，如下所示：

### 数据接口

该接口主要是部署了Python编写的爬虫代码，通过调用该接口，对相关需要的数据进行爬虫爬取，并返回。该接口主要是被FLASK框架中的功能函数中调用来获取实现功能所需要的数据。

除此之外，我们还开放了IIS服务，并通过内网穿透，将图片的文件夹以URL的形式暴露在公网中。我们的小程序无需下载图片，即可通过该URL浏览查看图片。

### 算法接口

该接口主要是编写了对训练好的算法模型的调用代码。我们会定期获取相关的数据，并以此训练模型。在功能函数需要使用该算法时，通过调用该算法接口，并传入参数，即可实现该算法的调用，并获得运算的最终结果。

### 小程序接口

该接口部署在FLASK框架的后端中，供小程序调用。当小程序需要使用某功能时，即可调用该功能对应的接口，即可得到返回的数据结果，从而实现小程序的功能。

## 系统配置策略

因为我们的作品基于微信小程序开发，因此系统无需进行配置，用户只需要进入微信，适配IOS和Android各版本。只需要进入微信，搜索到我们的小程序，或者扫描二维码，即可使用我们的作品。

## 系统部署方案

我们的后端需要部署在Windows操作系统中，并配置Python3.7环境，安装相应的包。除此之外，还需要布置内网穿透，实现接口在公网的暴露，从而实现小程序上的跨网段访问。

# 详细设计

## 期货百科功能模块

期货交易之所以相对于股票交易存在普及不足的差距，很大程度在于其专业知识获取的难度，相关信息存在分散不易整合的特点。本产品期货百科模块就是希望将相关的知识内容整合到同一板块中方便初入该行业的投资者更快、更有效地获取到有关期货交易的信息。

### 功能描述

集成提供有关期货的最重要的十七类知识，并集成多个官方定义的期货信息提供给用户，方便用户快速着手，学习相关的期货知识。

### 性能描述

以测试用机型HUAWEI-VOG-AL10型号为例，CPU为HUAWEI Kirin 980，内存为8G，在100M宽带的测试情况下，数据往返时间为200ms左右。

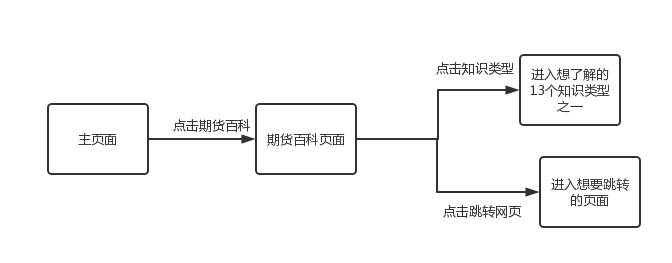
### 输入

输入为给定十七种常见知识类型或者是给定的三个官方给定的期货定义。

### 输出

输出为指定知识的具体内容，以及对内容的关键词句的提取和情感的分析情况。

### 程序逻辑



### 限制条件

期货百科知识列表数据不能为空。

## 期货数据功能模块

### 功能描述

通过用户指定关键字搜索，按关键字匹配由高到低依次返回给用户相关期货的信息，用户可以选定感兴趣的信息进行进一步的查阅，系统可以返回给用户该产品的具体信息，其中包括基本信息和二十交易日数据。

### 性能描述

以测试用机型HUAWEI-VOG-AL10型号为例，CPU为HUAWEI Kirin 980，内存为8G，在100M宽带的测试情况下，数据往返时间为1500ms左右。

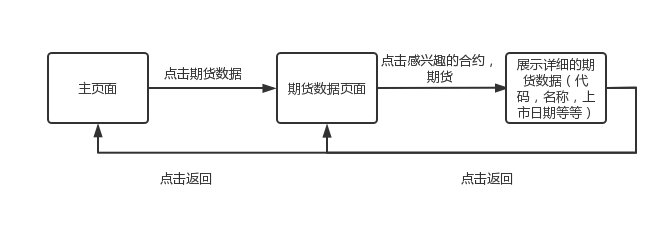
### 输入

输入为用户指定的期货产品的关键字。

### 输出

输出为按关键字重合度从高到低返回的相关期货产品的信息。用户可根据自身兴趣和需求选取具体的某一条期货产品信息，从而反馈给用户该产品的详细信息：基本信息，有交易标识、市场代码、市场名称、合约产品代码、合约录数、交易计量单位、交易单位、报价单位、最小报价单位说明、交割方式说明、上市日期、最后交易日期、最后交割日和交割时间说明等基本信息；二十交易日的数据，包括昨日收盘价、昨日结算价、开盘价、最高价、最低价、收盘价、结算价、涨跌收盘价-昨日结算价、涨跌结算价-昨日结算价、成交量、成交金额、持仓量、持仓量变化和交割结算价等信息。

### 程序逻辑



### 限制条件

期货数据列表的内容不为空。

## 财经快讯功能模块

### 功能描述

通过用户给定的关键词句，在当前多个主流财经网站中爬取包含相关关键字的新闻信息，并按关键字重合度高到低返回相关的财经新闻快讯。用户可以进一步选取感兴趣的新闻快讯阅读。

### 性能描述

以测试用机型HUAWEI-VOG-AL10型号为例，CPU为HUAWEI Kirin 980，内存为8G，在100M宽带的测试情况下，数据往返时间为1000ms左右。

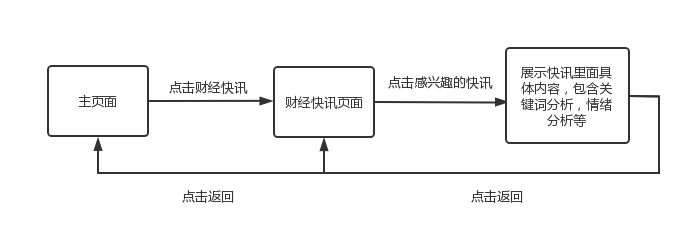
### 输入

输入为用户感兴趣的关键词句。

### 输出

输出为来自几个主流财经网站的、与用户指定词句相关的财经新闻快讯。用户可以进一步选择具体新闻进行阅读，系统将返回给用户原新闻经自然语言处理后的、简洁明了的新闻信息，并一并将新闻的关键词、关键句以及情感分析一并反馈给用户，为用户提供一个高效便捷的新闻检索、浏览的应用环境。

### 程序逻辑



### 限制条件

财经快讯列表的内容不为空

## 数据分析功能模块

### 功能描述

提供给用户两个功能，分别是价格预测和期货评价。其中价格预测功能可以给予用户提供的关键词句提供不同的与关键词句有关的期货产品的信息及基于LSTM的预测分析。

### 性能描述

以测试用机型HUAWEI-VOG-AL10型号为例，CPU为HUAWEI Kirin 980，内存为8G，在100M宽带的测试情况下，数据往返时间为2500ms左右。

### 输入

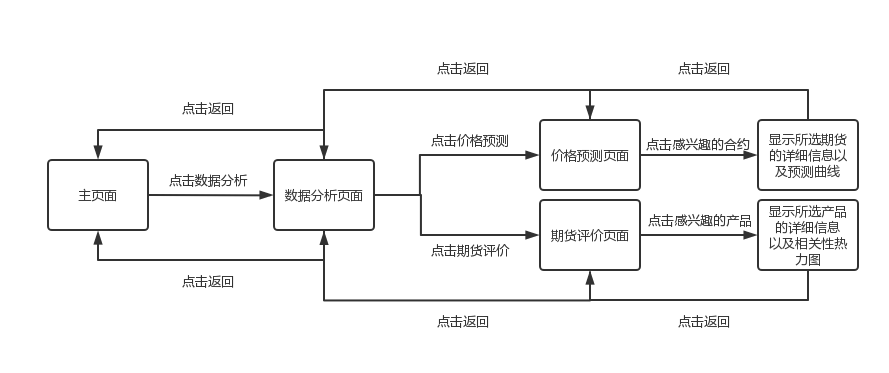
输入为用户指定的期货产品相关的关键词句。

### 输出

价格预测模块可以在用户指定的关键词句的基础上按关键词重合度检索出相关的期货产品列表，在用户选定具体期货产品后，返回如下信息：基本信息：基本信息，有交易标识、市场代码、市场名称、合约产品代码、合约录数、交易计量单位、交易单位、报价单位、最小报价单位说明、交割方式说明、上市日期、最后交易日期、最后交割日和交割时间说明等基本信息；预测结果：当前日期，预测日期，预测开盘价、最高价、最低价结果和对应预测模型LSTM在历史数据上的预测结果与实际情况对比的可视化图表展示。

期货评价模块可以在用户指定的关键词句的基础上按关键词重合度检索出相关的期货产品列表，在用户选定具体期货产品后，返回如下信息：相关性热力图和具体基本信息数据，包含周报成交量、周报交易量同比增减、周报成交金额与成交金额同比增减、年累计成交总量、成交总量同比增减、年累计成交金额与交金额同比增减、周报持仓量、持仓量环比增减、主力合约收盘价、收盘价环比涨跌、昨日仓单量、今日仓单量、成交量、成交量变化、持买仓量、持买仓量变化、持卖仓量、持卖仓量变化、积极情绪和消极情绪；综合评价，包含共九个因子的评估得分，并结合九个因子的得分最终给出该期货产品的总的评价分数。

### 程序逻辑



### 限制条件

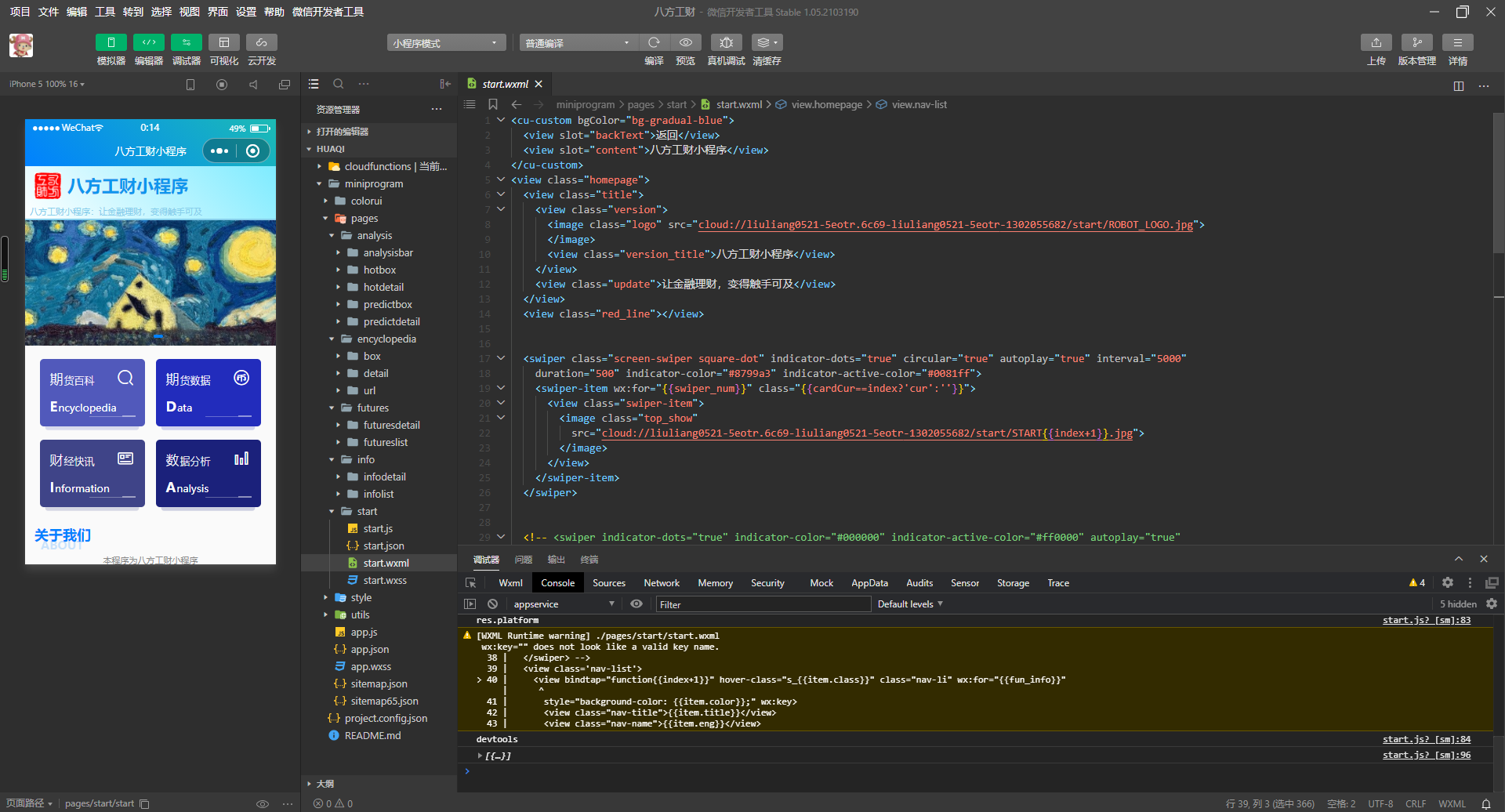
期货合约列表的内容不为空。

# 详细开发

我们的开发主要分为四个部分：

1. 小程序开发
2. 后端开发
3. 内网穿透
4. 算法开发

## 小程序开发

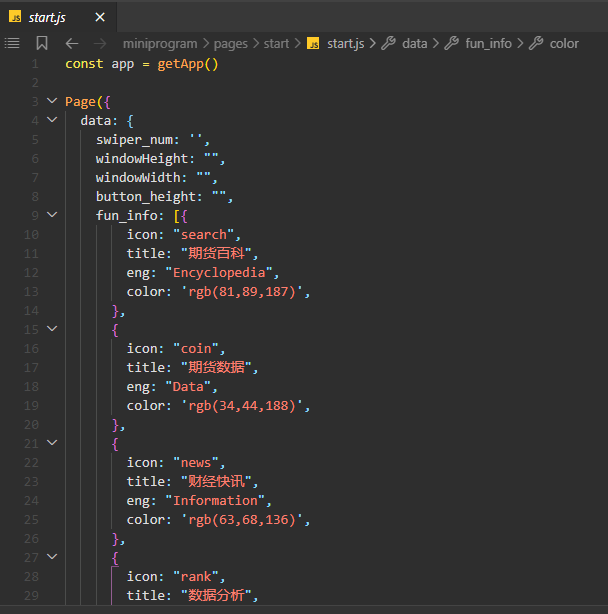


我们的小程序基于微信小程序的框架进行搭建，每一个页面由四部分组成：

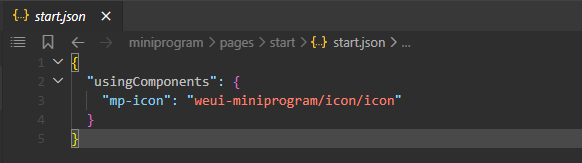
* JS文件
* JSON文件
* WXML文件
* WXSS文件



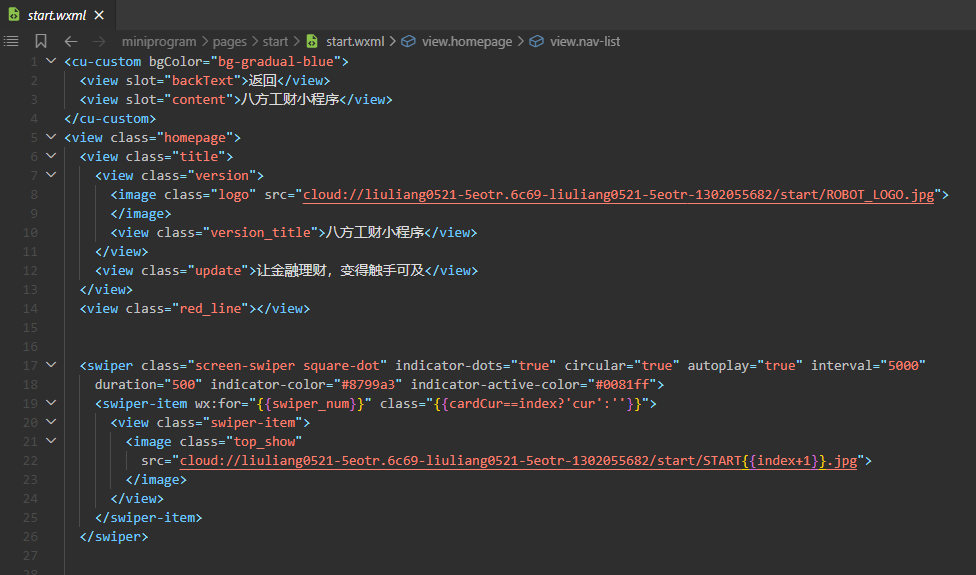
其中，JS文件主要负责编写页面的逻辑代码，包括变量、函数方法等。



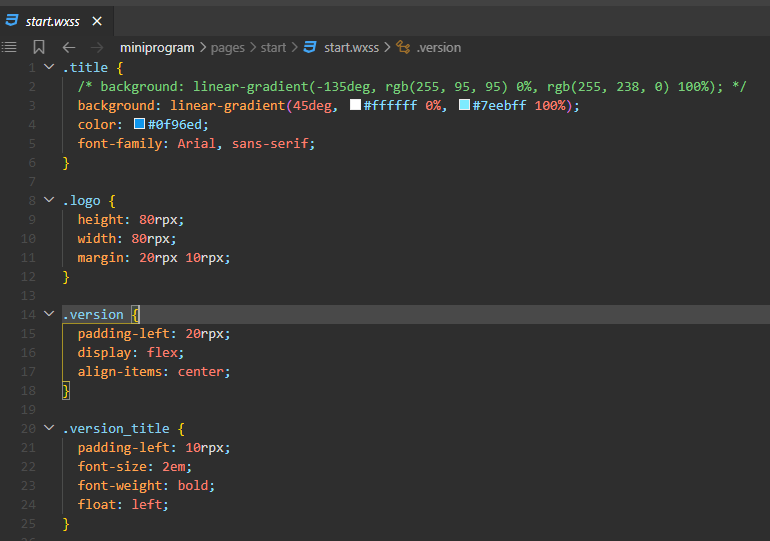
JSON文件主要负责对页面进行配置，例如其他的组件库或插件的调用，其他功能【如定位】的配置。



WXML文件主要是页面的布局开发，在该页面下布置控件及页面的参数、文字、图片等。



WXSS文件主要负责对空间的样式设置，对前端的页面进行优化，包括大小、字体、颜色、边框、ICON图片等设置。

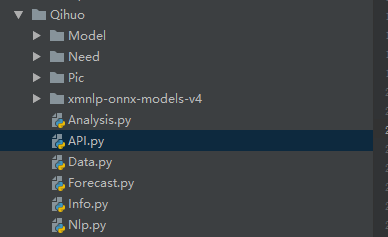


我们的小程序的PAGES的文件结构如下：

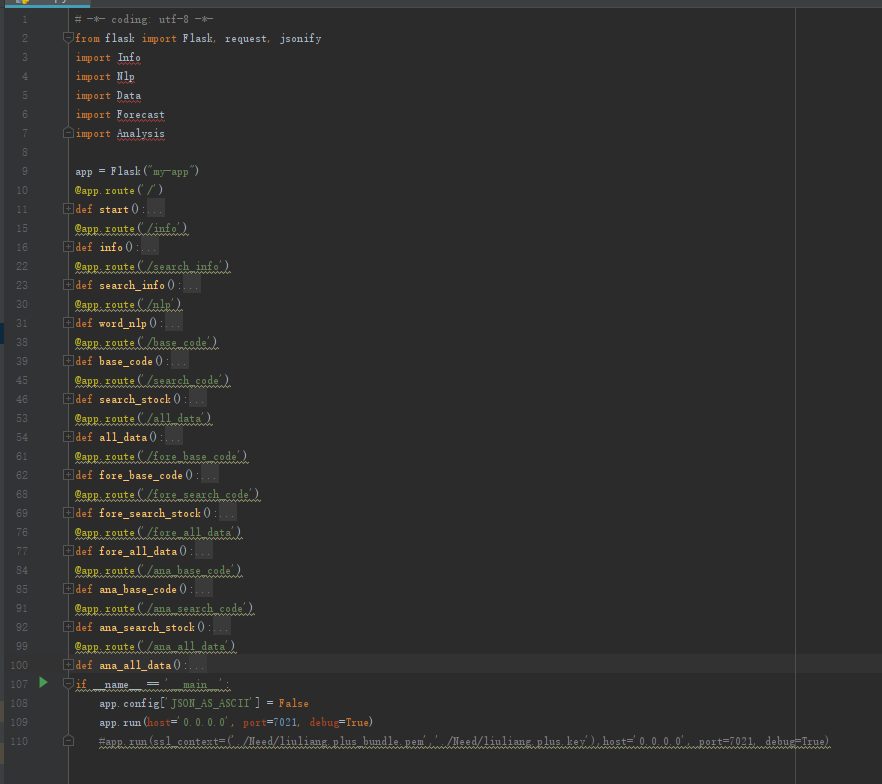
* analysis：主要实现了数据分析模块。
* analysisbar：实现了数据分析模块的Tarbar框架，通过选择类型转换不同的数据分析内容，包括价格预测和期货评价。
* hotbox：实现了期货评价模块的LIST页面，获取基本数据，通过点击进入详细页面查看详细信息。
* hotdetail：实现了期货评价模块的详细页面，可以查看详细信息。
* predictbox：实现了价格预测模块的LIST页面，获取基本数据，通过点击进入详细页面查看详细信息。
* predictdetail：实现了价格预测模块的详细页面，可以查看详细信息。
* encyclopedia：主要实现了期货百科模块。
* box：实现了期货百科模块的LIST页面，获取基本数据，通过点击进入详细页面查看详细信息。
* detail：实现了期货百科模块的详细页面，可以查看详细信息。
* url：实现外部链接的跳转。
* futures：主要实现了期货数据模块。
* futuresdetail：实现了期货数据模块的详细页面，可以查看详细信息。
* futureslist：实现了期货数据模块的LIST页面，获取基本数据，通过点击进入详细页面查看详细信息。
* info：主要实现了财经快讯模块。
* infodetail：实现了财经快讯模块的详细页面，可以查看详细信息。
* infolist：实现了财经快讯模块的LIST页面，获取基本数据，通过点击进入详细页面查看详细信息。
* start：主要实现了主页面。

## 后端开发

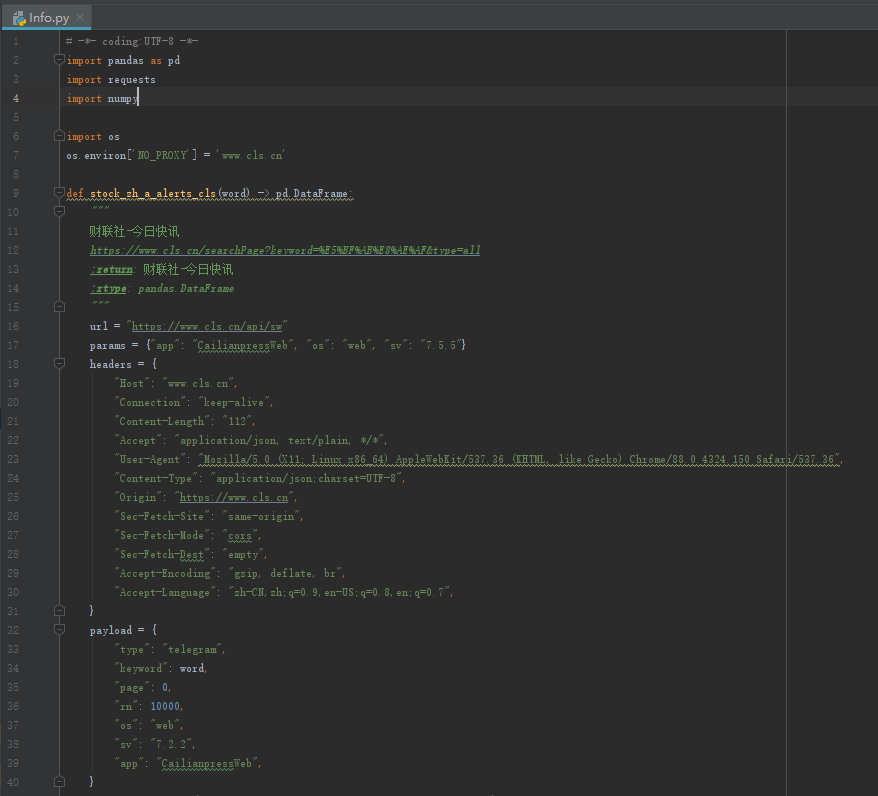
后端的结构：

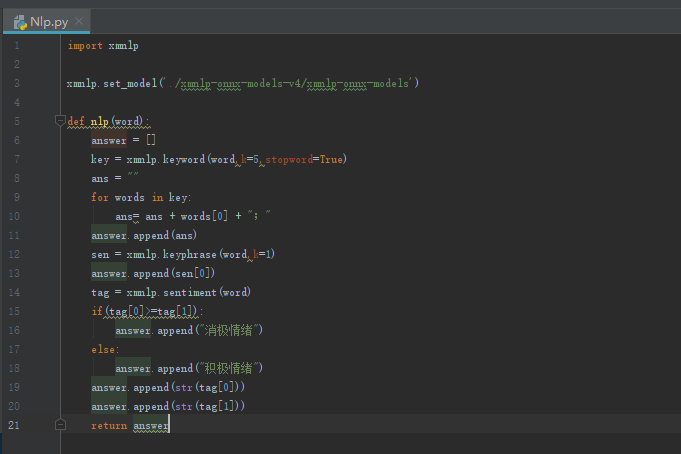


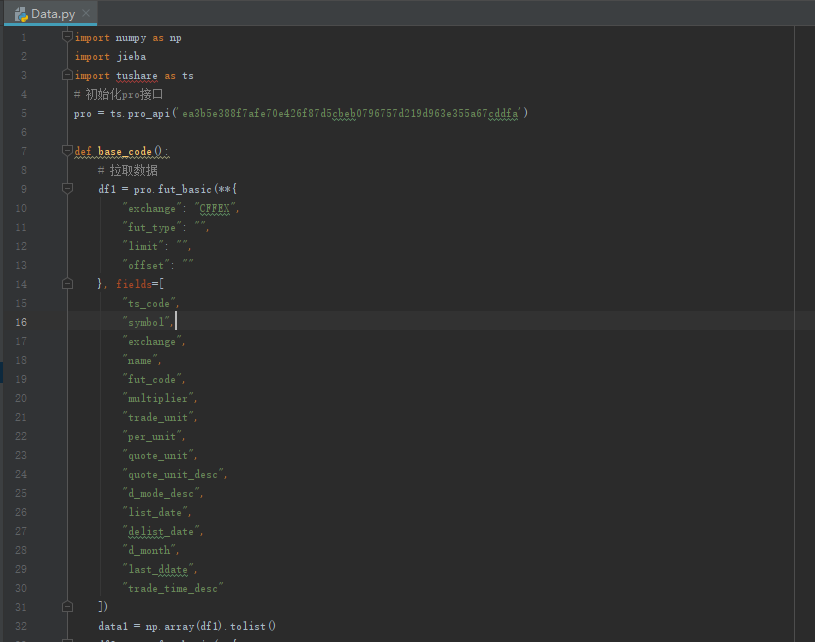
其中，Pic为图片文件夹，xmnlp为自然语言模型文件夹，Model为算法脚本模块，Need为SSL证书文件夹。后端在API.py中：

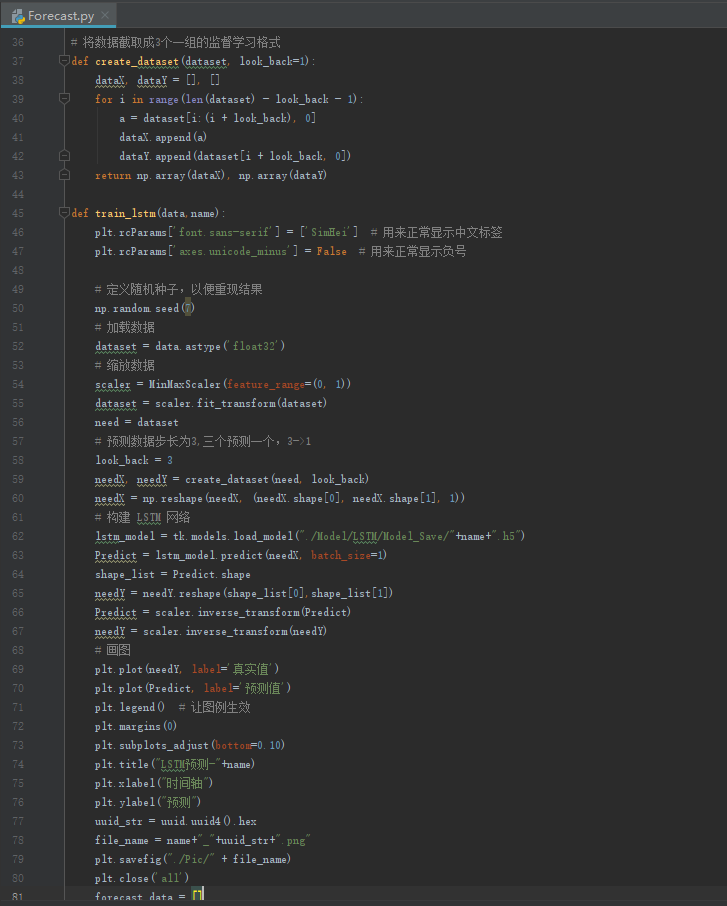


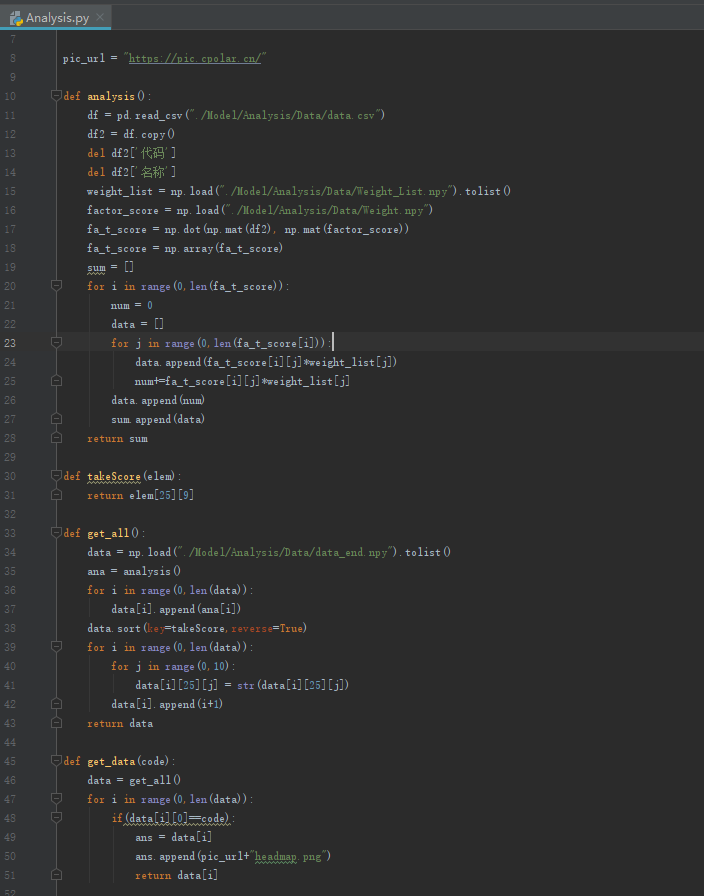
其中，功能函数分别在Info.py、Nlp.py、Data.py、Forecast.py、Analysis.py中实现对应的功能函数：





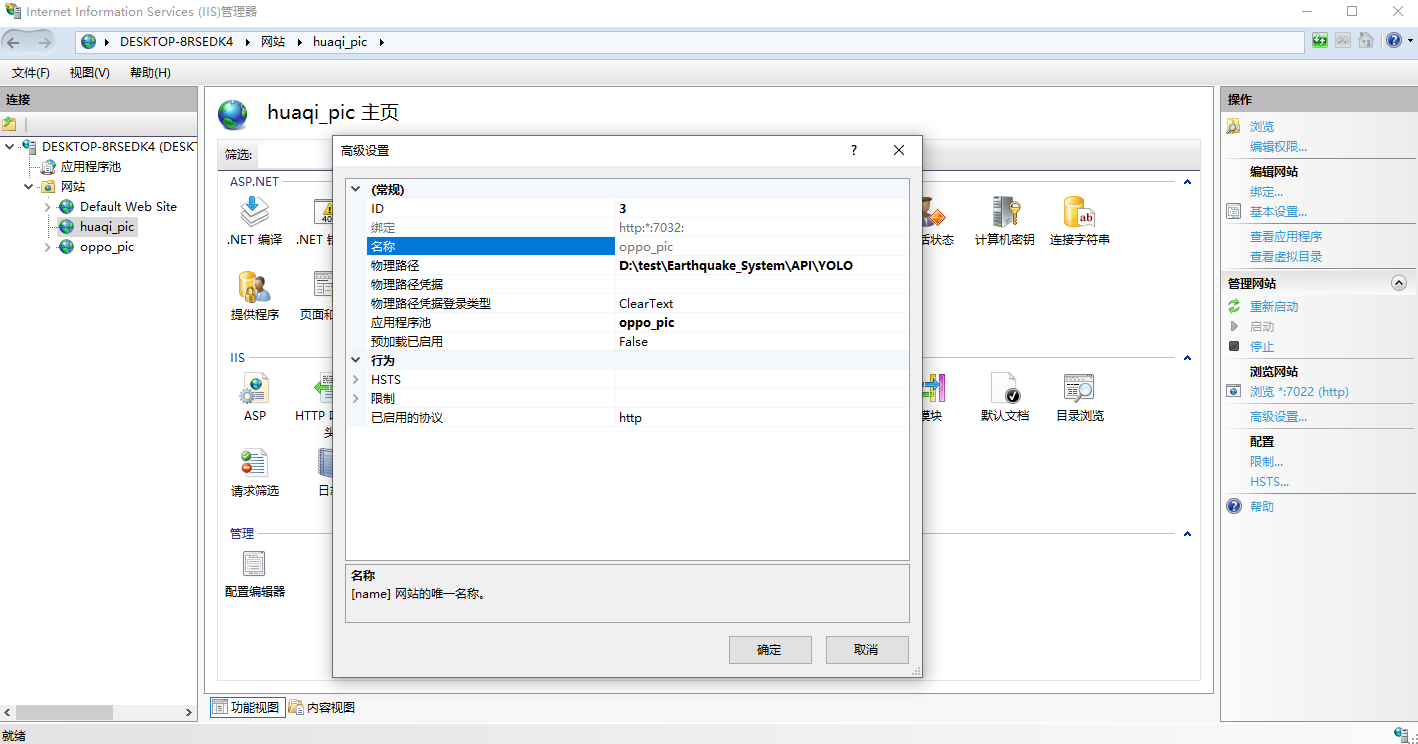




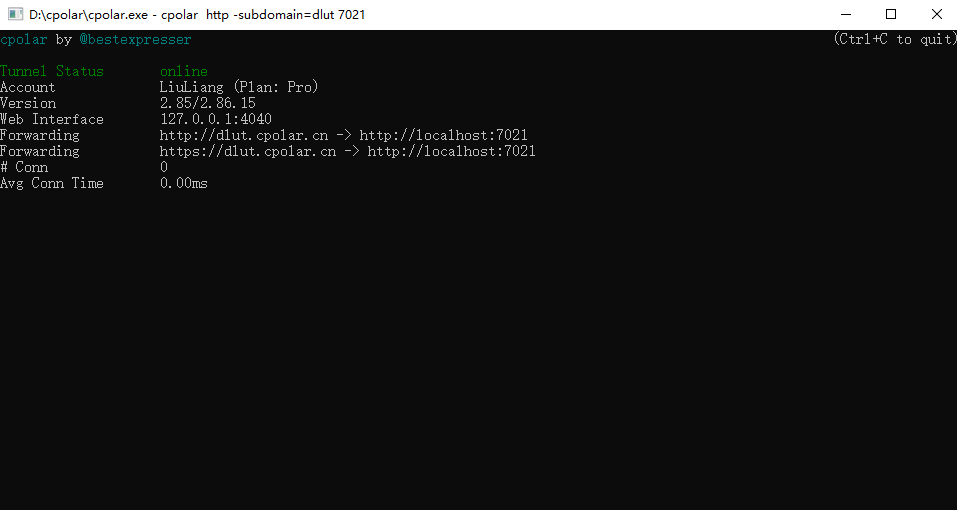


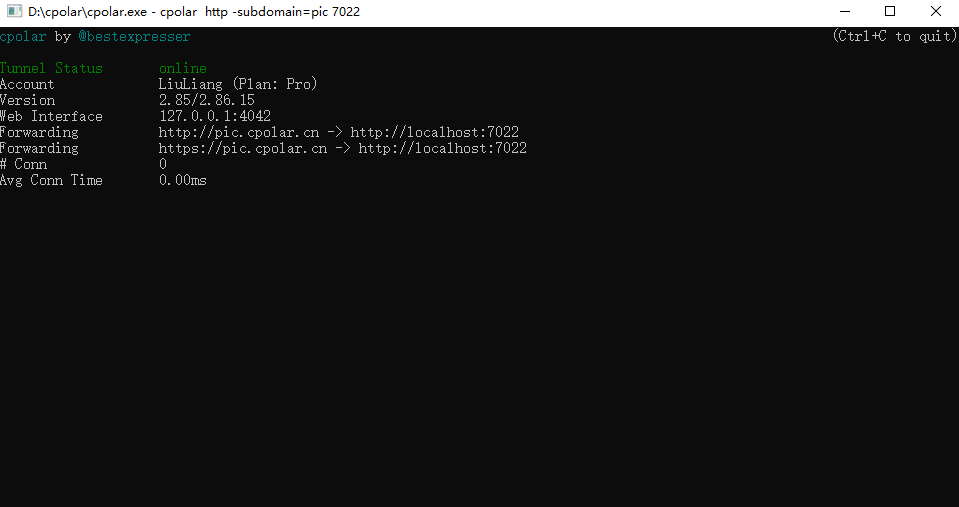
## 内网穿透

首先我们在本地服务器上开通IIS服务，将本地的图片文件夹以端口进行开放。



在此基础上，我们采用内网穿透，对后端FLASK框架的运行接口和图片文件夹接口进行穿透，暴露为公网的URL。



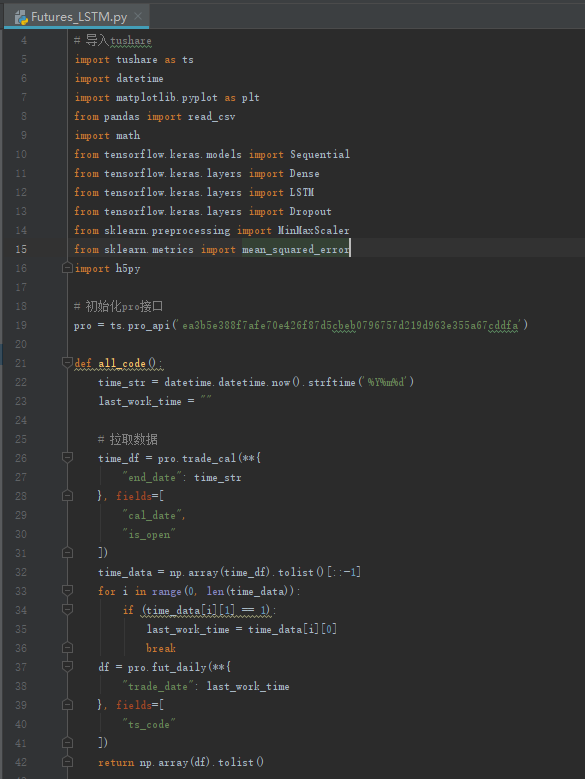


这样，我们就可以通过公网URL跨网段访问API接口和浏览图片。

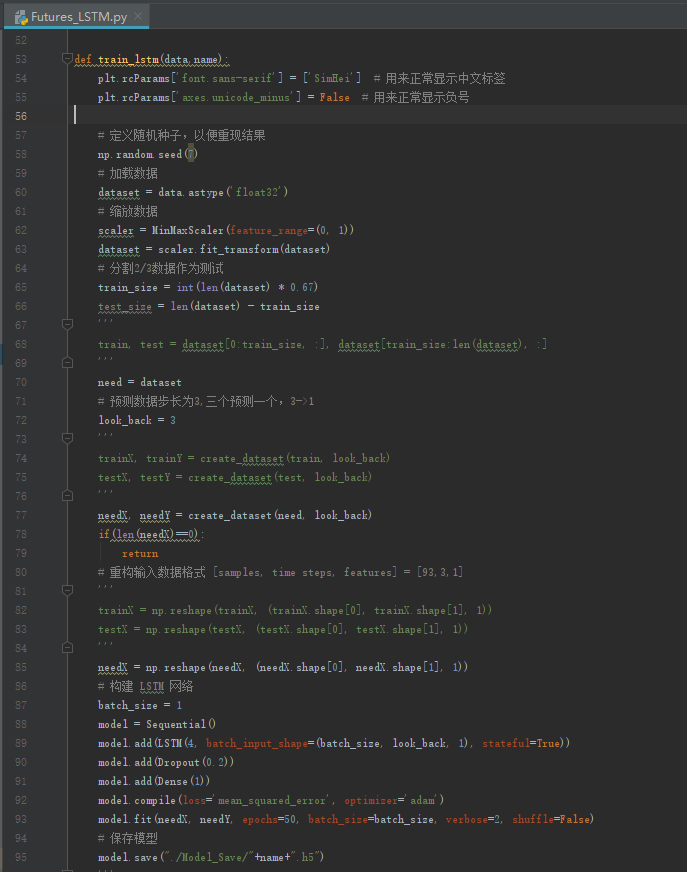
## 算法开发

* LSTM模型：

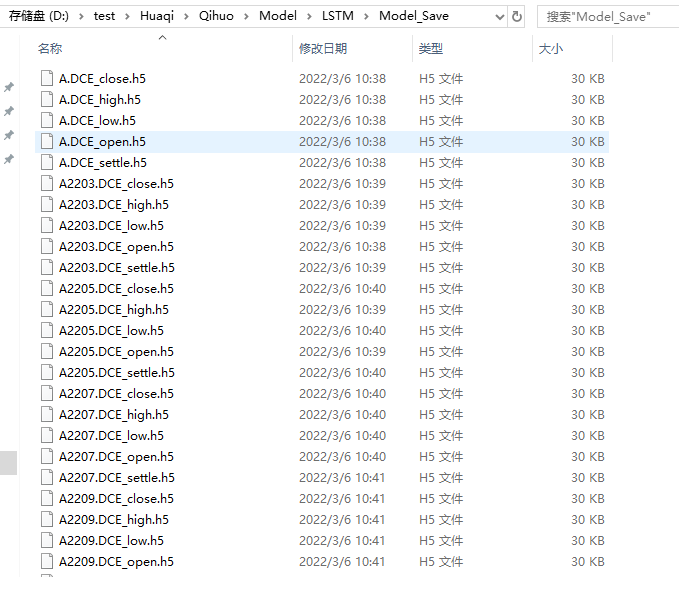
我们通过爬虫，获取LSTM模型所需要的数据：



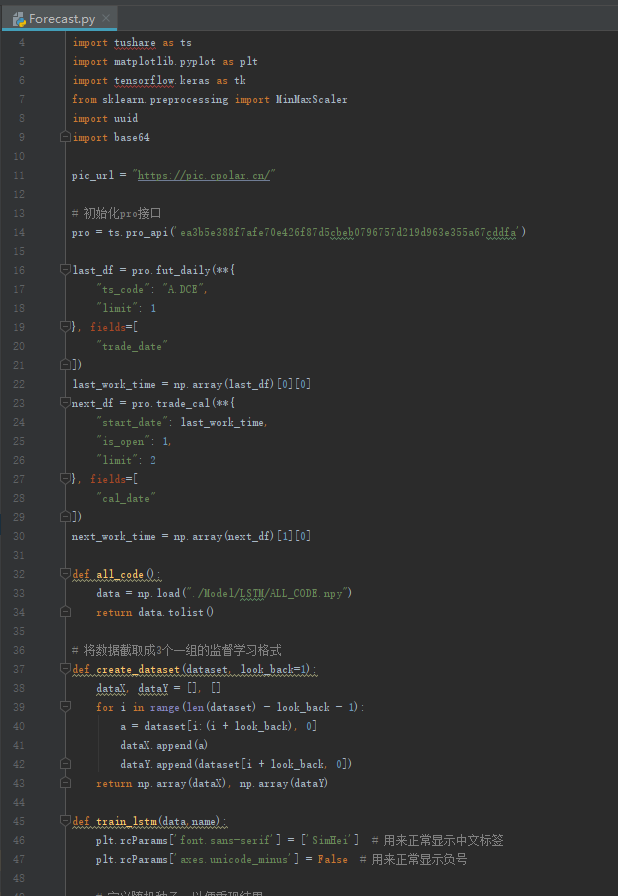
基于此，我们搭建LSTM网络，并进行模型的训练：



训练模型的结果保存在指定的文件夹中：

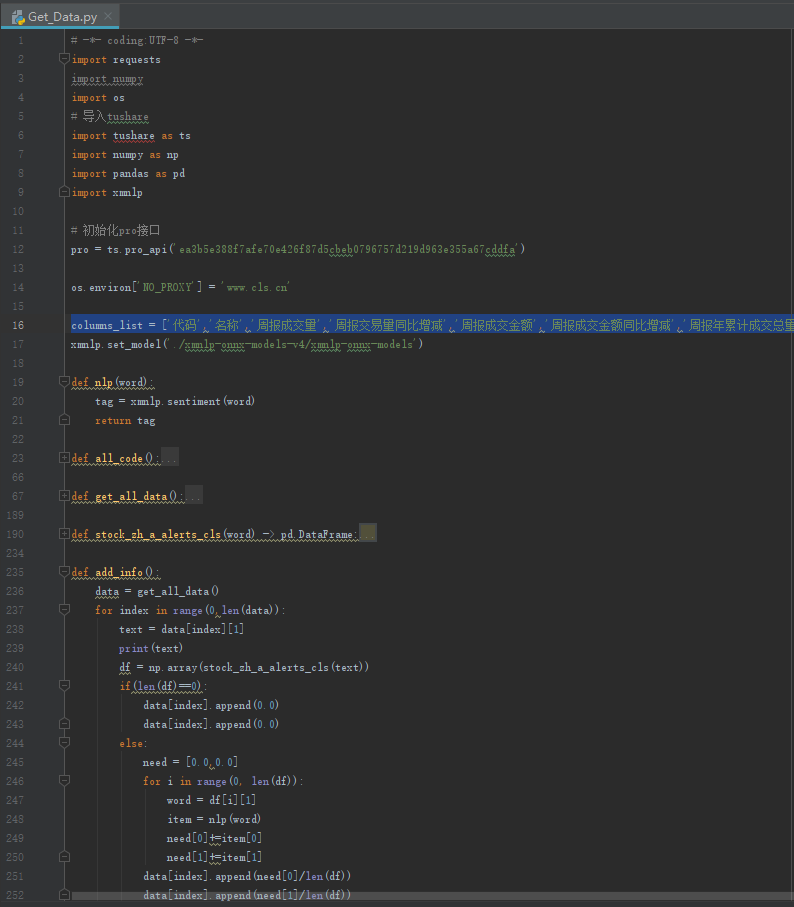


最终，我们把算法的调用封装为算法接口，以便直接调用：

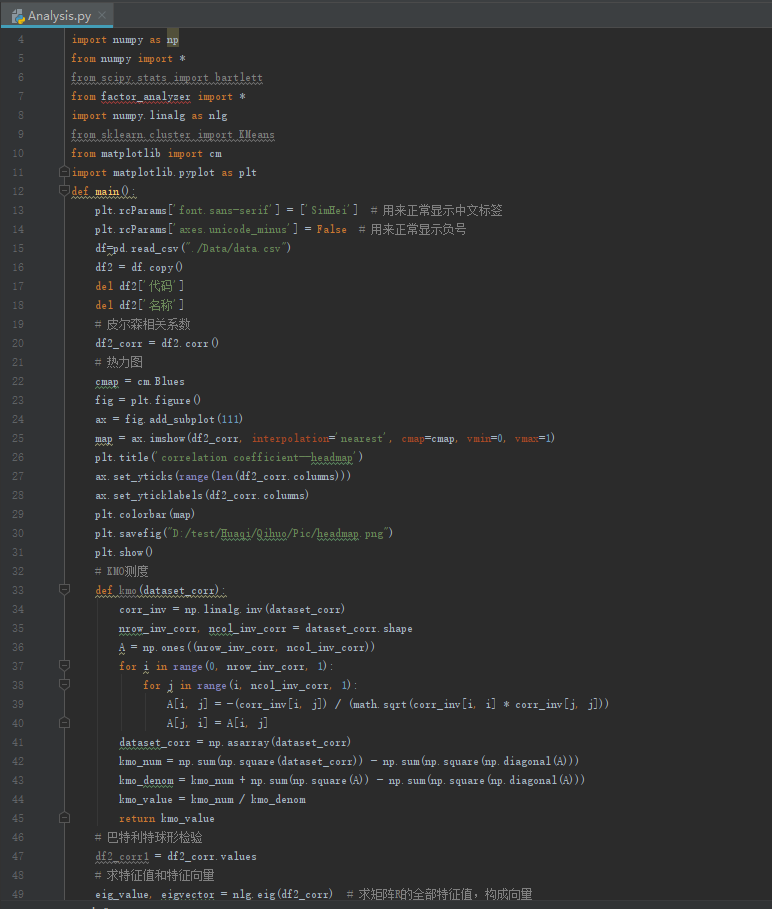


* 因子分析模型：

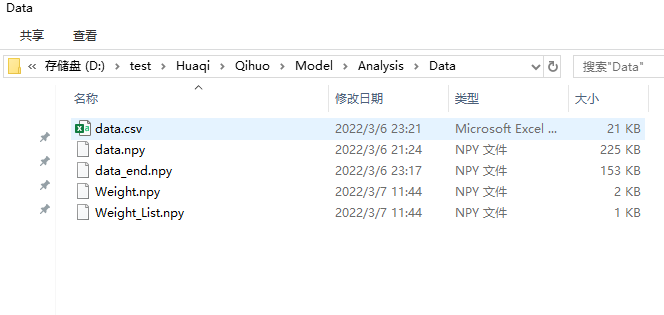
我们通过爬虫，获取因子分析模型所需要的数据：



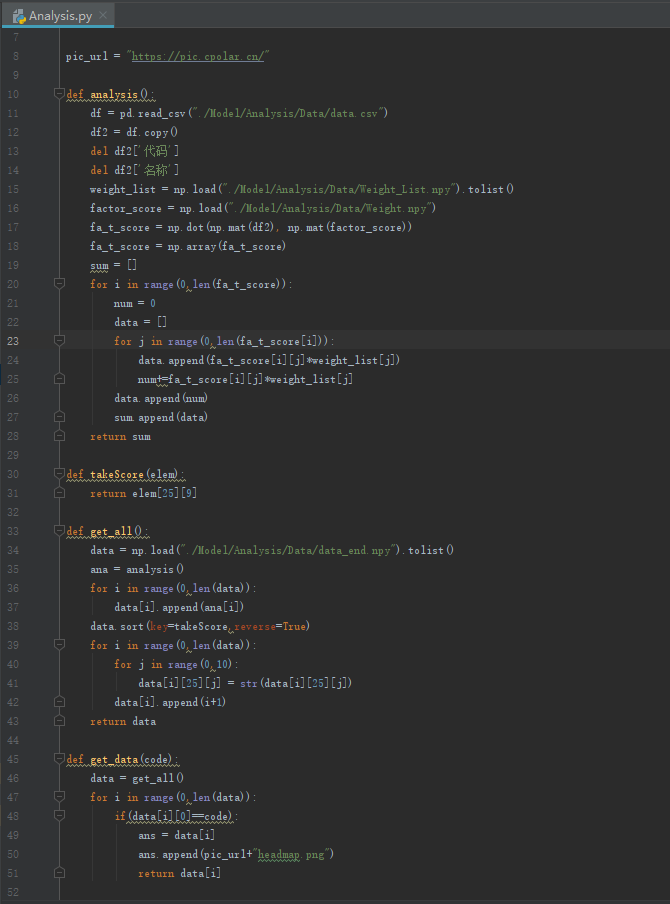
基于此，我们搭建因子分析模型，并进行因子分析法的模型构建：



训练模型的结果保存在指定的文件夹中：

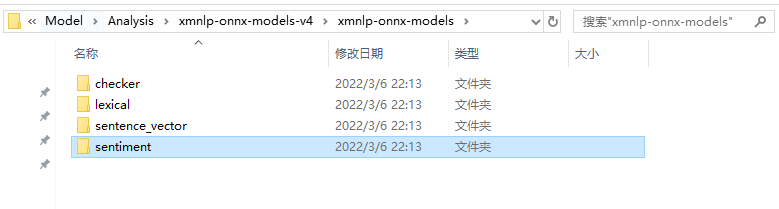


最终，我们把算法的调用封装为算法接口，以便直接调用：

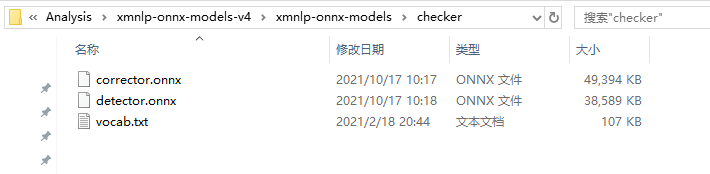


* 自然语言处理模型：

自然语言处理，即抽取关键词、关键句、情绪分析。我们将文本数据转化为词向量，并通过Seq2seq模型，搭载Attention机制，实现对文本数据的关键词句抽取。并采用Bert模型训练情感分类模型，将文本数据转化为向量特征进行计算。最终的模型：



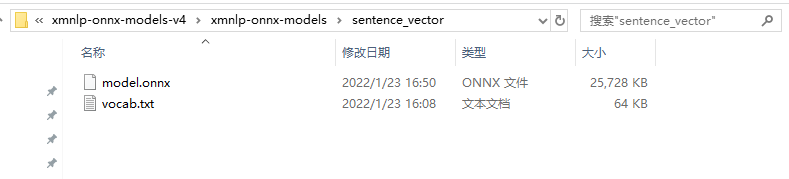
在该模型中，有校对模块：



有词汇模块：



有句向量模块：



有情绪模块：

