## 基于 Python 的金融数据采集与分析

1. **开发环境：**
   1. Windows 10 64 位系统 / macOS High Sierra 10.13.2
   2. Python 3.6.x ( 64 位)，需要安装 requirements.txt 中列明的依赖库

可以直接 pip install xxx 一个个安装，附上开发时用的版本

* 1. PyCharm 2017.3.2，可直接创建 Django 项目，免去许多配置
  2. MySQL 5.7.20，数据库

为保证正确运行，请尽量使用相同的开发环境。 涉及到python 的路径设置，请在 Pycharm中调试时，选择正确的运行路径。

注意：涉及到中文文本的处理，请使用 UTF-8 编码！

1. **项目结构与文件说明：（基于 Django 开发的Web界面）**
2. analysis 包，进行股票数据的分析
3. data 包，获取股票数据
4. fin 包，基于 Django 框架的主要业务逻辑与后端处理
5. Finance 包，项目的相关设置
6. static 文件夹，网页端使用到的 css 样式和一些 js 代码，用于渲染网页
7. templates 文件夹，html 网页代码，通过 Django 注入数据，作为前端 UI
8. util 包，提供一些小的功能函数
9. db.sqlite3，Django 默认需要一个数据库设置，可以不用管
10. main.py，非网页 UI 的测试，在没有 stocklist.txt 的时候，需要先执行，做好数据的准备工作
11. manage.py，Django Web 运行的入口
12. requirements.txt，python 环境与需要安装的第三方依赖包
13. stocklist.txt，保存了沪深股票的股票代码和股票名称
14. **主要代码文件说明**

本项目是 Django 框架开发的，其中的 main.py 可供测试用，命令行输出对应的结果。

* 1. analysis 中的 analysis.py，实现两个函数，analysis\_stock 根据股票代码，查找数据库并从外部获取最新的股票行情，得到实际股价、预测股价等数据；analysis\_stock\_news 则根据选择的股票，从新浪股吧获取股民的讨论，综合这些讨论分析情感记性，预测涨或者跌的可能性；
  2. data 中的 get\_data.py，获取对应股票的一定时间段内的行情，并保存在本地 MySQL 数据库中；
  3. fin 中，主要是 views.py，根据前端的数据，后台处理，并返回处理结果（预测股价等信息），供前端解析渲染，其他的文件都是 Django 项目创建时生成的；此外，Finance 中的代码中，也基本上保持 Django 项目创建时的内容，仅修改了 urls.py，添加了一些路径映射；
  4. static 下的 js/stock.js，可以根据数据在前端异步渲染数据
  5. util 下的 utils.py 中实现了一些辅助函数，get\_stock\_list 可以生成股票名称与代码文件：stocklist.txt

主要的代码解释可以阅读相关注释。

1. **程序运行方式**

可以通过 Pycharm 导入打开，支持命令行运行和 Web UI 运行，推荐在 Pycharm 中调试与运行，保持默认配置即可。

*注意：由于 python 环境和项目设置的路径不同，需要根据实际情况修改。直接运行可能会出错。有的时候需要重新选择一下 Python interpreter 就没有问题了。*

1. 准备工作
   1. 搭建好 python 环境和 MySQL 数据库，设置好数据库的用户名和密码，在 data/get\_data.py和 analysis/analysis.py 中的相关代码处填入对应的用户名、密码及地址（localhost）,端口（port，默认为3306），创建数据库 create database finance;
   2. 命令行使用 main.py

如果没有 stocklist.txt 文件，则需要首先执行 main.py 中的 utils.get\_stock\_list()。\其余的代码可供简单的测试。

1. Web UI 演示运行，专业版 Pycharm 可以直接运行 Django 项目，也可以在当前目录下，使用 “python manage.py runserver 8000” 运行，当有如下的提示后，可以访问 <http://127.0.0.1:8000> ，看到项目的 UI 界面，下面的操作请直接在网页上操作即可。

