

曲阜师范大学

本科生毕业论文（设计）



题 目 股票预测系统的设计与实现

姓 名 王易凡 学号 2019414076

院 系 网络空间安全学院

专 业 软件工程

指导教师 高 鹏 职 称 副教授

2023 年 5 月 20 日

曲阜师范大学教务处制

目 录

摘要	1
关键词	1
Abstract	1
Key words	1
1 相关概念	2
1.1 pytorch.....	2
1.2 MVC 设计模式.....	2
1.3 长短期记忆网络 LSTM	3
1.4 单点登录功能	3
2 功能分析	4
2.1 系统概述.....	4
2.2 功能目标.....	5
3 项目介绍	7
3.1 系统开发所用软件与版本	7
3.2 数据库设计与定义	7
3.2.1 基本数据字段设计	8
3.2.2 股票表字段设计	8
3.2.3 Redis 结构设计	8
4 股票数据分析与训练.....	8
4.1 数据来源	9
4.2 基本数据分析	9
4.3 模型训练	11
5 总结与展望	12
5.1 总结	12
5.2 展望	12
致谢	13
参考文献	13

股票预测系统的设计与实现

软件工程专业学生 王易凡

指导教师 高鹏

摘要: 随着我国经济的发展,尤其是改革开放之后股市与人们生活越来越贴近,本文就提供了一种股票数据可视化预测系统的搭建方式,帮助用户更好地分析股市的发展。本系统采用 B/S 架构和多节点部署的方式,功能包括单点登录、数据可视化以及模型预测等,并给出了核心数据库表的设计字段。其中模型预测采用的为长短期记忆网络,并在训练模型之前先验证了模型预测的可行性,验证方法为对部分数据集进行了分析,包括计算相关系数和可视化展示。最后在测试集中得到收盘价预测值与收盘价真实值的平均方差为 5.283。有较高的准确性。

关键词: 循环神经网络 长短期记忆网络 MVC 设计模式 单点登录

Design and Implementation of a Stock Forecasting System

Student majoring in Software Engineering

Wang Yi-Fan

Tutor

Gao Peng

Abstract: With the economic development of our country, especially after the reform and opening up, the stock market and people's life close to more and more, this article provides a kind of setting up of stock data visualization prediction system, to help users better analyze the development of the stock market. The system adopts B/S architecture and multi-node deployment, and its functions include single sign-on, data visualization and model prediction, etc., and gives the design fields of some database tables. The model was predicted using long and short term memory network, and the feasibility of the model prediction was verified before the model training. The verification method was to analyze part of the data set, including calculation of correlation coefficient and visual display. Finally, the average variance between the predicted closing price and the real closing price is 5.283. It has high accuracy.

Key words: Recurrent Neural Network; LSTM; Springboot Framework; SSO

引言

改革开放之后，尤其是在建成小康社会之后，我国逐渐富裕起来，我国的经济也多样化了起来。每个家庭也不再是局限于温饱，有许多支出可以用于投资等方面，股票投资也是其中一个主要方面。因此如何进行合理投资，能够在未来得到一定的收益，也要慎重地考虑。因此搭建一个为此服务的可视化的平台是一个非常重要事情。

现在的时代是大数据的时代，数据之间存在着一定关联，如何将数据联系在一起，挖掘其内部的信息是非常关键的。而如今机器学习算法及深度学习算法的兴起，为数据挖掘提供了可能性。例如传统算法中的多项式回归算法，支持向量机（Support Vector Machine, SVM）算法；神经网络算法中的长短期记忆人工神经网络（Long Short-Term Memory, LSTM）算法都能很好地对股票数据类型的数据进行拟合。除了传统网络之外，也有许多研究人员在想得到更好地网络来预测股票的趋势，杨志勇^[1]使用 BiLSTM-SA-TCN 时间序列，提出了在 LSMT 深度学习网络模型的基础上，融合两种新的机制的新网络，对招商银行和国金证券第二天收盘价进行预测使得在股票价格发生剧烈波动时也能很好地对其进行模拟。

对于数据在模型建立之前的预处理是十分重要的，同时数据对于用户的展示和管理也是十分重要的，这就需要搭建一个完善的管理系统。利用模型—视图—控制器（Model View Controller, MVC）进行分层能迅速且很好的开发一个管理系统。同时利用构件模块化能更好地对软件进行维护，如今已经有了完善的技术为其提供可能性。

单纯的数字信息是很难有说服力的，只有对数据展示出来，才能更好的进行分析，因此对数据的可视化也是十分重要的。这就要求搭建完善前端，vue 框架是当今主流的前端开发框架，与 e-charts 可视化组件相结合，这是目前数据大屏的主流组合方式，合理利用这些与股票可视化相结合，就能使用户清楚明了的了解当今最合适的动作，有利于社会更好发展。

因此本文采用前后端分离的技术，搭建一个基于 B/S 架构的系统软件。后端运用 MVC 框架，前端运用 e-charts 组件做可视化处理，采用 LSTM 人工智能神经网络算法模拟股票走势。

1 相关概念

1.1 Pytorch

Pytorch 是由 Facebook 人工智能研究院开发的一个库，支持多种语言编程，常用 python 语言进行编写。Pytorch 相较于同样广泛使用的 tensorflow 库来说代码更加简洁^[2]，且 pytorch 速度相对较快，其语法更符合程序员的理解与学习。同时它支持 gpu 计算，能极大提高训练和运行速度。它有着完整的生态，能够用其搭建完善的神经网络模型，包括卷积神经网络模型，循环神经网络模型，对抗神经网络模型等等。

1.2 MVC 设计模式

MVC^[3]是一种理论框架，不同于传统的实际存在的 ssh 与 ssm 框架，他没有具体的代码，而是对代码进行了分层表示，规定了每一层的任务，让每一层只用完成自己的任务，同时极大地提高了代码的可读性。其层次分别为 Model, View, Controller 三层。与该理论框架适应性最高的代码性框架是 SpringMVC 架构。其含有 Controller 层, Service 层, Mapper 层, Dao 层。Controller 层一般不涉及逻辑代码的编写，而是用来做接口设计；Service 层则一般用来编写和业务相关的逻辑，被 Controller 层的函数直接调用；Mapper 层则是与数据

库相交互，可以被 Service 层调用；Dao 层则直接定义了数据类，且 Dao 层中定义的数据类也可以分为多种类型，其中常用类型可分为三类。第一种是与数据库表所直接对应的数据类，是最基础的类型且每个软件中都必定包含此类型，被称为 PO；第二种是业务类数据类型，是在对业务相关数据进行处理转换时不可避免地生成的新的数据组合，被称为 BO；第三种是展示给用户的类型，就是直接通过接口所返回的数据组合，且没有在 PO 中直接定义。除此之外还有着 DTO、DAO、POJO 等多种类型格式。因为其优秀的模块化的解耦性，使得其成为在软件开发中的首选框架。

1.3 长短期记忆网络 LSTM

随着深度学习的发展，基本的神经网络已经无法满足复杂多变的业务场景，众多新的网络模型也在被不断创新提出，如 RNN 循环神经网络模型。对于语句理解的场景时，输入的数据为多个字符或多个单词，且每个字符的单个含义可能无法代表整个语句的含义，对于传统的全链接神经网络很难完成这种任务，而循环神经网络则能很好的联系语句的上下文，从而提取语句的整体语义。同时对于股票预测来说也是这样，股票的升涨趋势不但与当前的数据相关，也与其之前的数据有关，运用 RNN 模型能很好地对股票涨幅进行拟合。

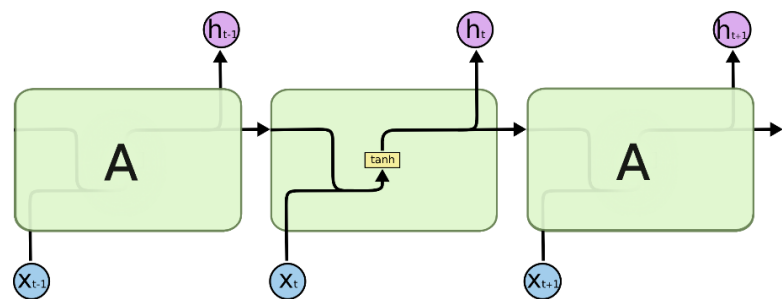


图 1-1 RNN 神经网络结构

传统 RNN 网络虽然运用了 tanh 激活函数，一定程度上避免了梯度爆炸与梯度弥撒的现象。但对于较长的语句中，且相关单词或字符在该语句中相隔较远的情况时，传统 RNN 网络就稍显薄弱。LSTM^[4]网络是在 RNN 网络基础上改进得到的，增加了忘记门、输入门与输出门三种结构。可以理解为将语句内的关键信息记录而淘汰了无用信息，因此能更好地模拟前文所提到的语句类型。对于股票类型数据，它同样存在着相关数据相隔较远但语义相关的情况，因此本文中用 LSTM 模型来完成对股票实时预测的操作。

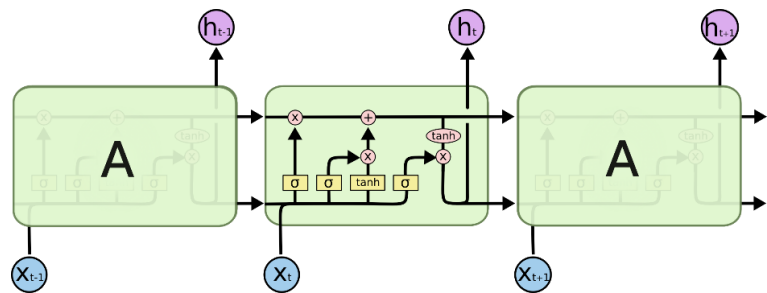


图 1-2 LSTM 神经网络结构

1.4 单点登录功能

HTTP 协议^[5]具有无状态的特点，即建立的 HTTP 连接服务器并不知道其曾服务过谁，或之前有没有服务过当前发起请求的客户机，在每次建立连接时对于 HTTP 服务器来说都是服务一个新的客户端。但一般业务中存在大量这样情景，即需要之前的服务信息来进行下一步的请求。单点登录就考虑了以上的情况，并一般采用 Redis 辅助功能的实现。在用户第一次登录时生成一个 token 令牌，即用户的唯一通行证，存放在 Cookie 与 Session 中，在每次进行微服务时都要验证该通行证，保证安全性。并且要求在用户注销或长时间不进

行新的操作行为时将 Cookie 与 Session 中的通行证信息删除掉，保证用户数据的安全。单点登录对于单点部署或分布式情况都同样适用，有较高的鲁棒性。其对应 UML 时序图如 1-3 图所示。

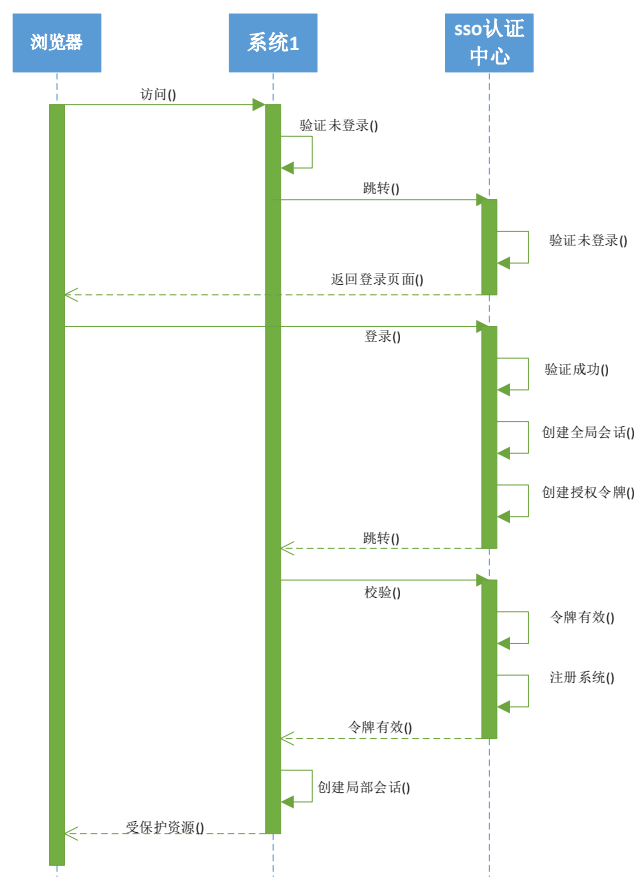


图 1-3 单点登录 UML 时序图

2 功能分析

2.1 系统概述

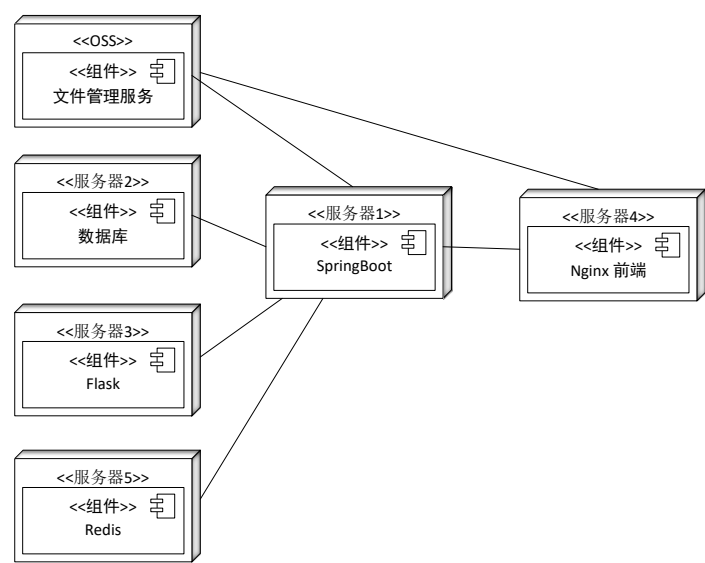


图 2-1 股票预测系统 UML 部署图

系统的目标是为了更好地提供股票数据展示和一定程度上通过提供未来预测走势为用户股票投资提供建议。

该系统只需要数据展示，对于管理人员角色并没有额外突出功能，不需要额外去单独设计，因此目前本系统不为多角色系统，只存在一种角色。但在软件设计模式中存在三大原则：开闭原则、依赖倒置原则以及接口隔离原则。其中开闭原则的定义可以理解为开发软件时应该能更好的为以后扩展做基础，并不能随意修改已经建立的系统^[6]。因此本系统先搭建出多角色逻辑，但目前只存放一种角色。能够根据角色在页面中不同来展示不同的信息。

在架构方面采用多节点部署的方式，可以用采用当前热门框架，如前端采用 vue 框架并通过 Nginx 部署，后端则是 Springboot 框架直接打包成 jar 包进行部署，训练好的 LSTM 模型封装在 flask 框架中进行部署，文件管理服务采用阿里云的 OSS 服务，除此之外数据库也要部署在服务器中。其部署图如图 2-1 所示。

2.2 功能目标

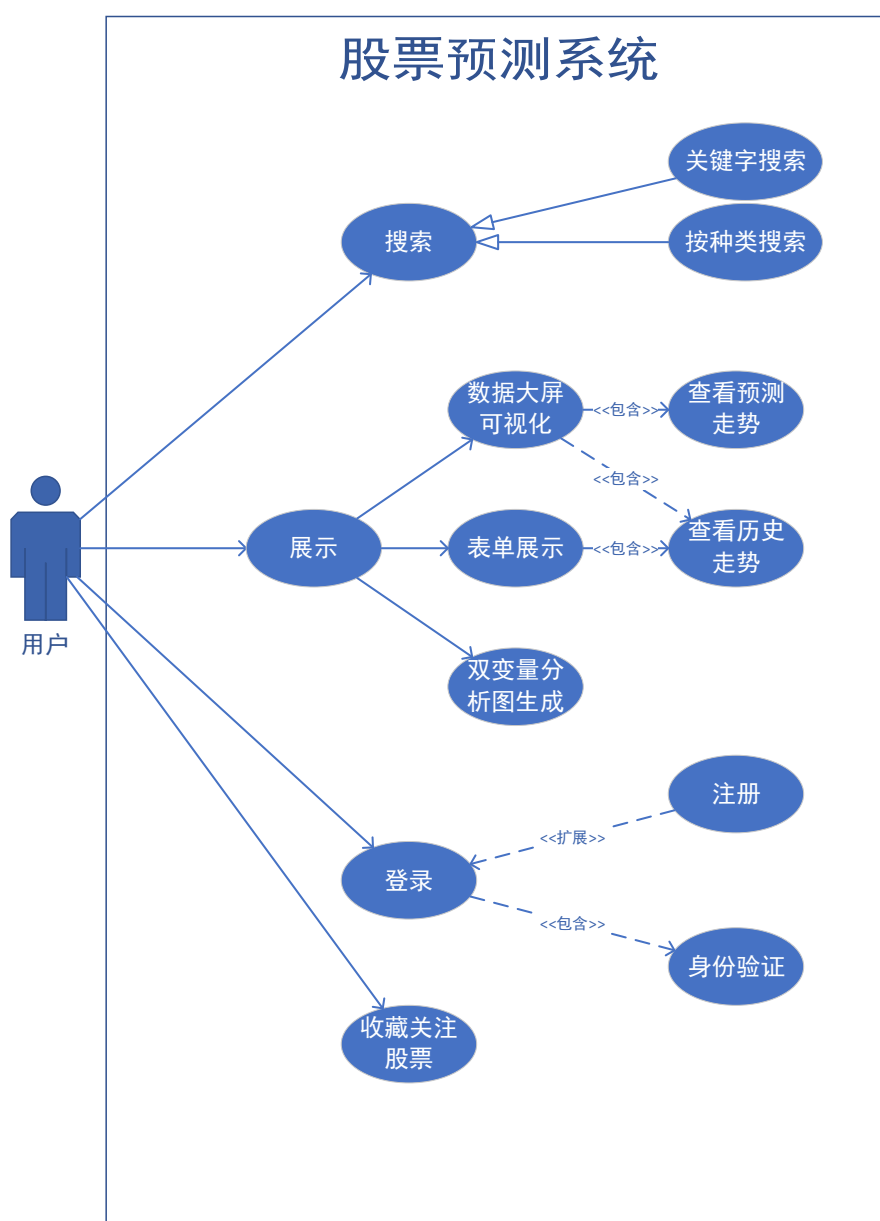


图 2-2 股票预测系统 UML 用例图

该系统主要可以分为三个功能模块，第一个功能为用户登录，以便于让用户收藏自

已关注的信息。并在登录功能基础上延申出注册功能，并需要身份验证功能来保证每个微服务准确无误地服务对应的用户。第二个功能为搜索，搜索分为两种方式，一是可以通过股票类型来构建出对应树形结构后，进行逐层查找；二是关键字搜索，通过关键字模糊查询直接获得相关的股票信息。第三个功能是展示，主要分为三种形式，一种是通过数据大屏展示出股市的走向，并展示出系统实时预测的未来发展趋势；二是直接在表单中展示出相关数据，可以直接观察到全部直观数据；三是可以通过后端生成选定股票的双变量分析图，让用户更好地结合其它股票变动趋势来预测目标股票的变化趋势，其详细分析过程见第 4.2 节。其对应 UML 用例图如图 2-2 所示。

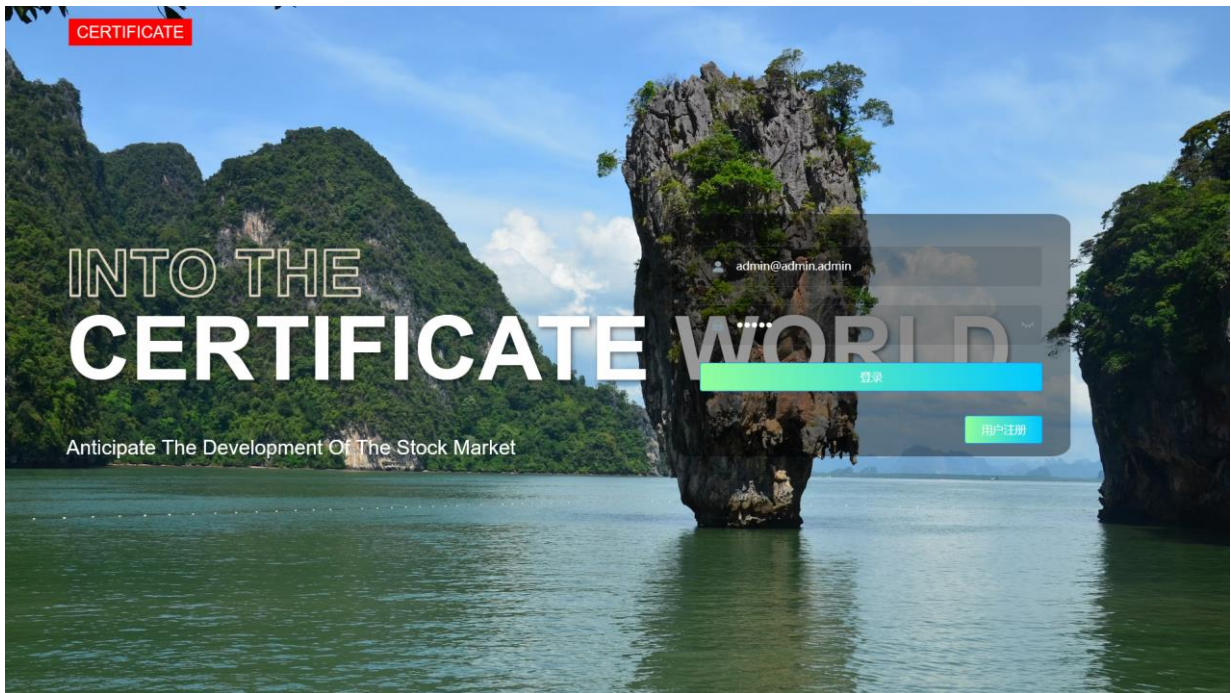


图 2-2 股票预测系统登录页面

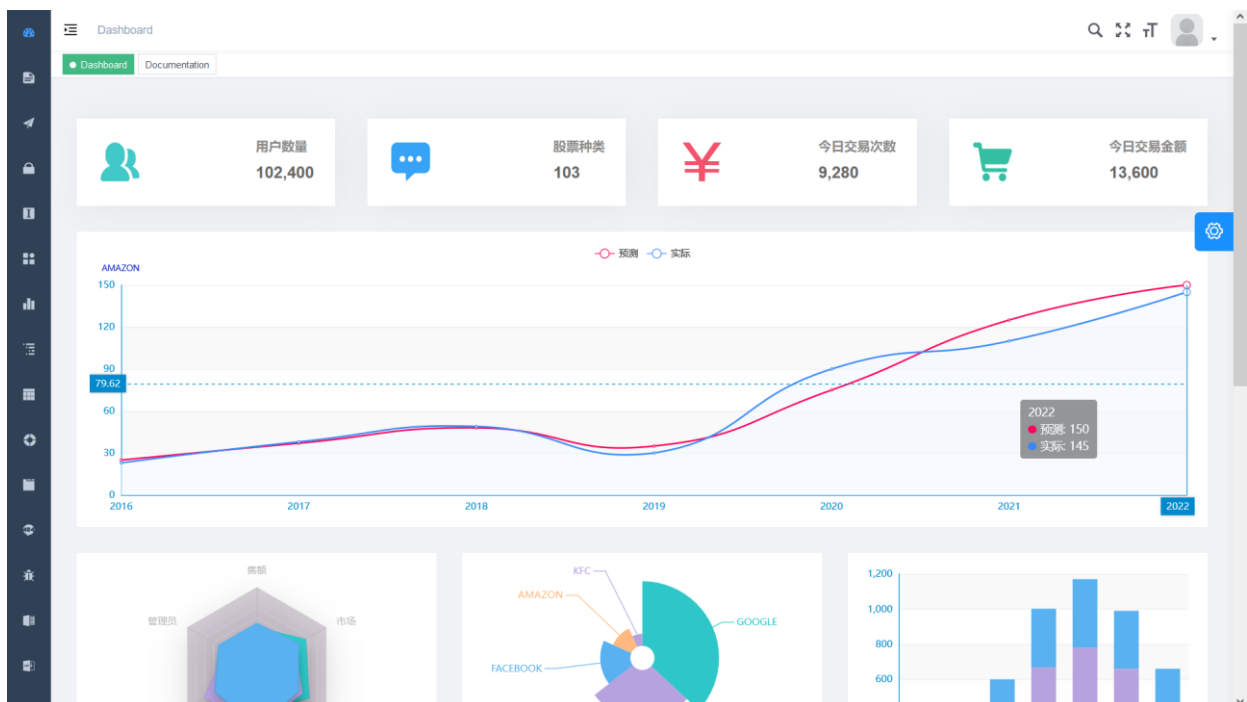


图 2-3 股票预测系统首页

本系统的登录页面与首页分别如图 2-2 和图 2-3 所示。可以清晰找到登录和注册按钮。

再首页也可以看到本系统的基本信息，包括随机股票信息展示，用户数量、股票收录种类数量，今日交易次数，今日交易金额等。

基于 e-charts 的股票可视化展示图如图 2-4 所示，先输入要查询的股票名称，然后选择报告类型，分为日报、周报、月报、年报，点击搜索之后便可以出现对应数据。展示图中为月报展示。

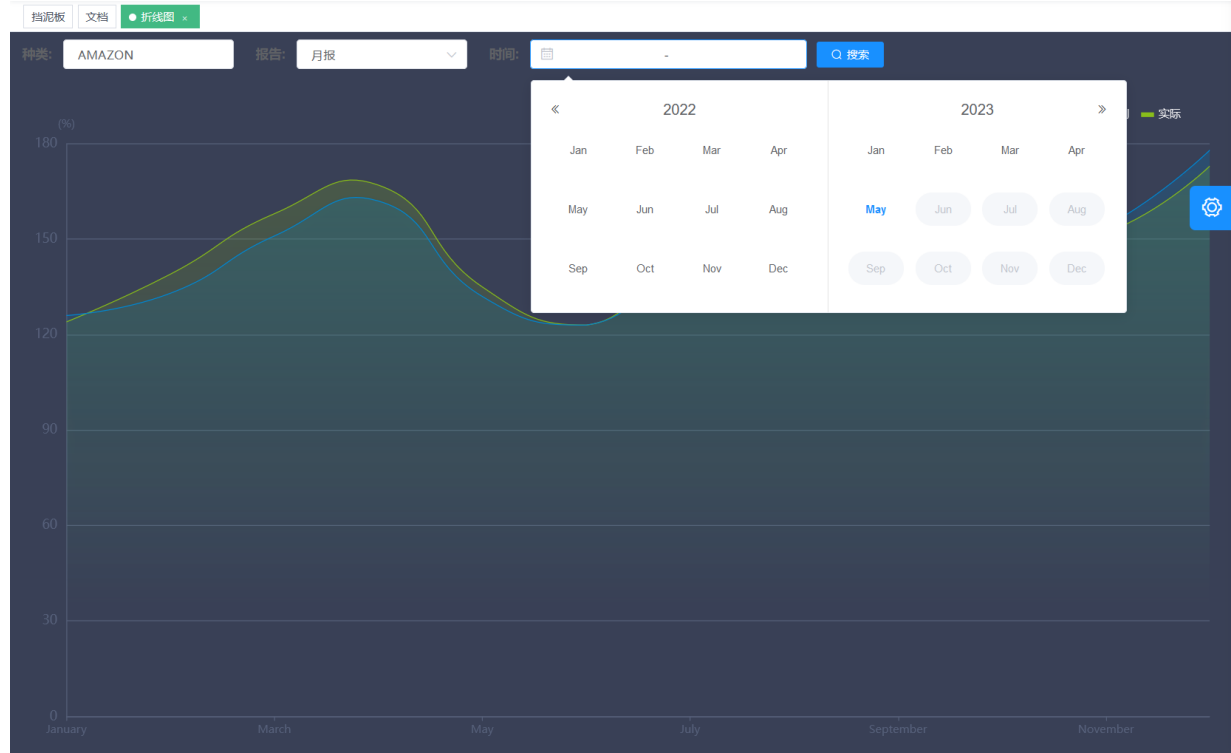


图 2-4 股票预测可视化展示

3 项目介绍

3.1 系统开发所用软件与版本

本系统开发需要借助一些列工具，其推荐所使用的软件如表 3-1 所示。

表 3-1 开发软件信息

开发工具	版本
IDEA	IntelliJ IDEA 2022.3.1 x64
MySQL	mysql-5.7.20
Java	jdk1.8.0-181
Vue	2.2
Springboot	2.1.4
Navicat	11.2.7.0 企业版
Chrome	92.0.4515.107（正式版本）
Pycharm	PyCharm 2022.3.1

3.2 数据库设计与定义

3.2.1 基本数据字段设计

数据库字段在设计时不但需要与业务相关的字段，同时也一般需要七个基本字段，该字段与业务本身无关，只是为了更好地维护数据本身，为以后系统扩展提供帮助。其对应基本字段如表 3-2 所示。

表 3-2 基本数据字段结构设计

名称	描述	数据类型	长度	可否自增	主键与否	可否为空	可否只读
id	用户 ID	VARCHAR	255	TRUE	TRUE	FALSE	FALSE
version	乐观锁	Integer	255	FALSE	FALSE	FALSE	TRUE
deleted	逻辑删除	Integer	1	FALSE	FALSE	FALSE	FALSE
create_by	创建者	VARCHAR	255	FALSE	FALSE	FALSE	TRUE
create_time	创建日期	Datetime	255	FALSE	FALSE	FALSE	TRUE
update_by	更新人员	VARCHAR	255	FALSE	FALSE	TRUE	TRUE
update_time	更新日期	Datetime	255	FALSE	FALSE	TRUE	TRUE

id 字段一般作为表的主键，唯一表示某一行数据；deleted 表示逻辑删除，即在做删除操作时，并不是真正的将其在数据库中删除，而是将该字段值改为 0，这样做的好处是能保护数据，适应当今大数据时代，获得更多的数据，也能为数据回复操作提供更加便利方式；createBy 与 createTime 字段是标志该行数据被谁创建并何时创建；updateBy 与 updateTime 字段则是标志该行数据最后一次修改是谁进行并何时进行。为了保证规范性其余所有表都要包含这几个字段。

3.2.2 股票表字段设计

股票信息存储一共设计两张表，第一张表 share_certificate_name 用来存储股票类型，以及他们的分类情况；第二张表 share_certificate_content 存储股票的相关信息，并设计外键与 share_certificate_name 表进行关联。如表 3-3 与 3-4 所示。

表 3-3 share_certificate_name 表结构设计

名称	描述	数据类型	长度	可否自增	主键与否	可否为空	可否只读
certificate_name	股票名称	VARCHAR	255	FALSE	FALSE	FALSE	TRUE
certificate_code	股票代码	VARCHAR	255	FALSE	FALSE	FALSE	TRUE
certificate_class	股票分类	VARCHAR	255	FALSE	FALSE	FALSE	TRUE

表 3-4 share_certificate_content 表结构设计

名称	描述	数据类型	长度	可否自增	主键与否	可否为空	可否只读
certificate_id	股票主键	VARCHAR	255	FALSE	FALSE	FALSE	FALSE
data	日期	Datetime	255	FALSE	FALSE	FALSE	TRUE
open	开盘价	Decimal	255	FALSE	FALSE	FALSE	TRUE
close	收盘价	Decimal	255	FALSE	FALSE	FALSE	TRUE
low	最低价	Decimal	255	FALSE	FALSE	FALSE	TRUE
high	最高价	Decimal	255	FALSE	FALSE	FALSE	TRUE
volume	成交量	Integer	255	FALSE	FALSE	FALSE	FALSE
change	涨跌	Decimal	255	FALSE	FALSE	FALSE	TRUE
change_ratio	涨幅比	Decimal	255	FALSE	FALSE	FALSE	TRUE

3.2.3 Redis 结构设计

在本系统中 Redis 内存数据库主要负责三个业务，第一个业务为单点登录服务，存储已经登录用户的信息，详细内容已在上文说明。第二个是存储 24 小时内股票的走势的变化。第三个是存储模拟预测的数据。

Redis 作为内存数据库，对于数据频繁查询效率要明显好于 Mysql 数据库。对于用户整体来说，他们往往会更加关心近期股票的变化，会进行频繁地查询。另外模拟预测的数据会因为当天真实数据的更新，而要重新进行模拟预测未来的走势变化，期间该类型数据会发生频繁变化。因此采用 Redis 作为这两种情况的数据存储能更好地提高系统的性能。

4 股票数据分析与训练

4.1 数据来源

	Open	High	Low	Close	Adj Close	Volume	company_name
Date							
2023-01-17 00:00:00-05:00	98.680000	98.889999	95.730003	96.050003	96.050003	72755000	AMAZON
2023-01-18 00:00:00-05:00	97.250000	99.320000	95.379997	95.459999	95.459999	79570400	AMAZON
2023-01-19 00:00:00-05:00	94.739998	95.440002	92.860001	93.680000	93.680000	69002700	AMAZON
2023-01-20 00:00:00-05:00	93.860001	97.349998	93.199997	97.250000	97.250000	67307100	AMAZON
2023-01-23 00:00:00-05:00	97.559998	97.779999	95.860001	97.519997	97.519997	76501100	AMAZON
2023-01-24 00:00:00-05:00	96.930000	98.089996	96.000000	96.320000	96.320000	66929500	AMAZON
2023-01-25 00:00:00-05:00	92.559998	97.239998	91.519997	97.180000	97.180000	94261600	AMAZON
2023-01-26 00:00:00-05:00	98.239998	99.489998	96.919998	99.220001	99.220001	68523600	AMAZON
2023-01-27 00:00:00-05:00	99.529999	103.489998	99.529999	102.239998	102.239998	87678100	AMAZON

图 4-1 数据集用例展示

对于系统运行时股票相关数据，是通过调用接口直接获得的实际数据，并存储在数据库中。但在构建阶段，为了方便验证模型的可行性则是采用的 **Kaggle** 中开放的数据集。其数据格式如图 4-1 所示。

数据集内的字段与本文数据库构建的字段相符，都具有开盘价、收盘价、最低价、最高价、成交量数据。与本系统兼容度较高，可以用来验证本文构建模型的有效性。

4.2 基本数据分析

在前端除了基本数据展示之外，还提供了额外的可供用户进行分析的图表。在给定的数据中，收盘价是股票在正常交易日内交易的最后价格。股票的收盘价是投资者用来追踪其一段时间内表现的标准基准。图 4-2 为用 **pandas** 及 **matplotlib** 可视化图像展示。

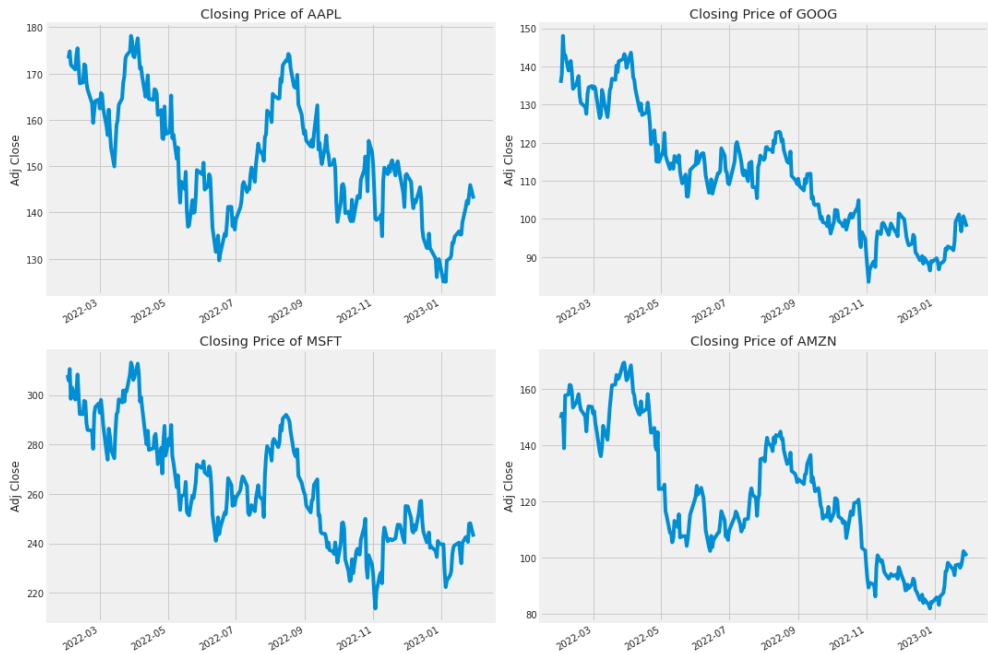


图 4-2 收盘价数据可视化展示

从图 4-2 中可以直观得出，虽然每种股票走势不完全相同，但其大致走向趋势相似，因此股票之间存在某种关联关系。

为了更直观地观测其存在相关性，可以采用双变量分布图，外侧为两个变量直方图，内部为散点图的方式进行表示，当两组数据相关度越高时，通过散点图中的点做出的最小二乘法^[7]斜率越接近 1，该图像可根据用户选取股票不同，后端自动生成并传递到前端进行

展示。如图 4-3 所示，两组完全相同数据所模拟出直线斜率为 1，完全相关。

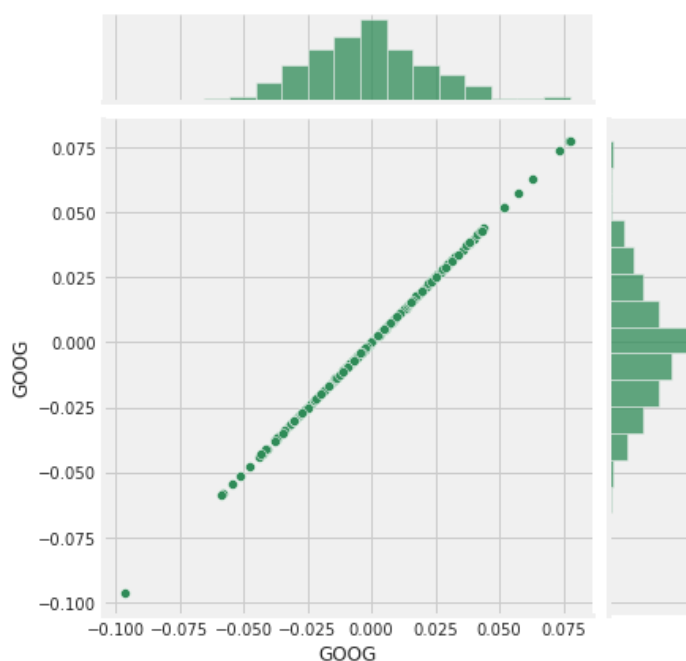


图 4-3 两组完全相同数据相似度可视化

之后建立四组不同的双变量分布图，如图 4-4 所示，任何两组不同股票数据之间都存在着正相关关系。由此可直观得出不同股票之间的变化存在较高的相关联关系，股票的发展趋势出现相似现象，所以其它股票数据也能为目标股票数据模拟时提供一定积极作用，在单只股票数据量较小的情况下，为更好的股票模拟预测提供了可行性。其详细相关系数^[8]热力图如图 4-5 所示。

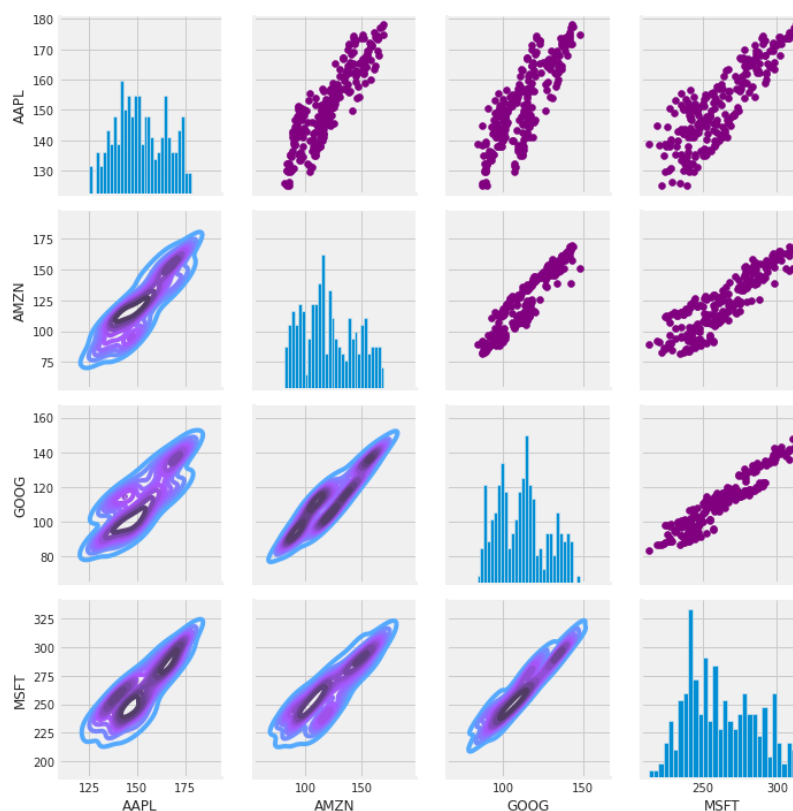


图 4-4 不同数据相关性展示



图 4-5 详细相关系数热力图

4.3 模型训练

用数据集中 AAPL 数据作为构造模型的数据，输入数据为之前股票的收盘价信息，对应标签数据为当日股票的收盘价，是一个回归任务。在开始时，将数据划分为两部分训练集和验证集^[9]。训练集数据用来生成模型的参数，验证集用来验证模型的准确性。若直接用训练集进行验证时，会让模型重新前向传播一次在训练阶段训练的数据，该数据符合模型参数，会导致正确率虚高，无法证明模型的通用性，因此必须采用之前未在训练过程中使用过的数据进行验证。在本文中训练集与验证集比例为 19: 1。

之后进行模型训练，将初始学习率设为 0.001，模型训练时的梯度下降算法采用 adam 优化器^[10]算法加速训练过程。具体训练代码如图 4-6 所示。

```
data = df.filter(['Close'])
dataset = data.values
training_data_len = int(np.ceil( len(dataset) * .95 ))
scaler = MinMaxScaler(feature_range=(0,1))
scaled_data = scaler.fit_transform(dataset)
train_data = scaled_data[0:int(training_data_len), :]
x_train = []
y_train = []

for i in range(60, len(train_data)):
    x_train.append(train_data[i-60:i, 0])
    y_train.append(train_data[i, 0])
    if i<= 61:
        print(x_train)
        print(y_train)
        print()

x_train, y_train = np.array(x_train), np.array(y_train)
x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1], 1))
model = Sequential()
model.add(LSTM(128, return_sequences=True, input_shape= (x_train.shape[1], 1)))
model.add(LSTM(64, return_sequences=False))
model.add(Dense(25))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mean_squared_error')
model.fit(x_train, y_train, batch_size=1, epochs=1)
```

图 4-6 训练代码

训练完毕后进行验证阶段，验证时采用方差作为衡量模型当日收盘价预测值和实际收盘价值误差的指标，最终标准差约为 5.283。具体验证代码如图 4-7 所示，且预测数据与真实数据对比如图 4-8 所示。

```

test_data = scaled_data[training_data_len - 60: , :]

x_test = []
y_test = dataset[training_data_len:, :]
for i in range(60, len(test_data)):
    x_test.append(test_data[i-60:i, 0])

x_test = np.array(x_test)
x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1))

predictions = model.predict(x_test)
predictions = scaler.inverse_transform(predictions)

rmse = np.sqrt(np.mean(((predictions - y_test) ** 2)))

```

图 4-7 验证代码

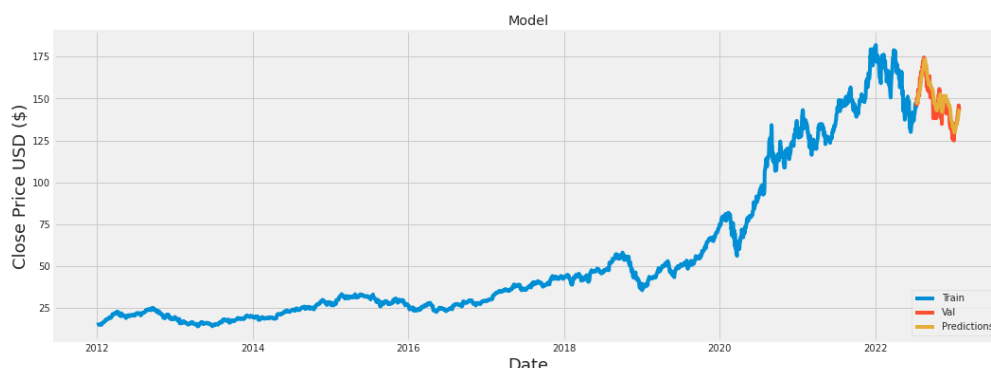


图 4-8 模型预测可视化

从平均方差数值和可视化图像可得出 LSTM 对股票预测准确性较高，可用于给用户提取一定的投资建议。之后只需调用接口收集到对应股票数据后，用该方法进行训练部署即可。系统每日实时模拟出未来股市的可能走向后存储在 Redis 中，之后通过 Springboot 后端调取展示到前端页面中，供用户参考完成全部需求。

5 总结与展望

5.1 总结

在该系统设计与搭建的过程中，我复习了许多知识，包括 Springboot 框架，Vue 框架以及 Pytorch 深度学习库。将其运用到了实践当中，并在开发过程中注重代码可读性，运用 MVC 架构将后端代码分层，并前端尽可能实现组件化增加代码的复用度，更易于后期完善。最后在项目部署时了解了 Linux 基本命令以及 nginx 的应用。

同时也有许多不足。例如该系统仍是相对简陋，在整体功能而言，只是提供了数据查询、数据可视化展示、单点登录功能，并未实现模拟股票交易的细节。除功能缺陷以外在开发过程中也遇到过许多问题，为解决问题查询了大量文档，尤其在模型训练时，为了能更好地分析数据，去查询了解了许多有关股票知识。最后得到了较为合理的可视化展示图像。五个多月转瞬即逝，软件开发过程不乏存在挫折，但也充满了希望。每当解决一个问题时，都会有强烈的满足感支撑自己继续前进。

5.2 展望

当今时代是信息的时代，是智能的时代，传统人工智能算法和神经网络算法广泛应用，同时现在软件开发技术也已经相当成熟。所有的一切都在以肉眼可见的速度发展，我想要继续在该行业该领域发展，并为其做出一定的贡献，所以我要继续理解现有技术的原理，并学习新的技术，使得自己能在该领域独挡一面。不能只因学到本科阶段的一点知识而满

足沾沾自喜。在未来研究生阶段，我想主要在软件开发领域发展，不会辜负本科期间所付出的努力。如今的自己各方各面还有所欠缺，惭愧之余更要奋勇前进提升自我，一定会在未来不断发展自我。

致谢

2019 年步入曲园，在这里完成本科四年的学习，迈出了在该行业发展的第一步。从初入校园的懵懂到现在逐渐能独挡一面，离不开曲园老师们的谆谆教诲。感谢母校给了全面发展的机会，许多本科成绩的实现离不开学院老师的支持。曲园良好的学习氛围是我之前所没有体验到的，使我有不断前进的动力。未来也许会有很多各式各样的经历，更多不一样的生活环境，但我相信我不会遗忘曲园四年对我的教导。

同时我也要感谢高鹏老师，在本篇文章书写时提供了许多指导，使得我能在经历瓶颈时或遇到无法解决的问题时突破难关。为我的论文提供里程碑式的建议。能够顺利完成本文的书写。最后衷心祝高老师事业进步，万事如意。也祝愿母校曲园在未来能再创辉煌，取得新的成就。

参考文献

- [1] 杨智勇,叶玉玺,周瑜.基于 BiLSTM-SA-TCN 时间序列模型在股票预测中的应用[J/OL].南京信息工程大学学报(自然科学版):1-12[2023-04-12].
- [2] 张康林,叶春明,李钊慧等.基于 Pytorch 的 LSTM 模型对股价的分析与预测[J].计算机技术与发展,2021,31(01):161-167.
- [3] 于昕,廖晨伶,周卫丽.基于 MVC 的软件架构重构与优化研究[J].吉林化工学院学报,2021,38(07):49-52.
- [4] 隋金城. 基于 LSTM 神经网络的股票预测研究[D].青岛科技大学,2020.
- [5] 徐智宇. 基于 B/S 架构的工具管理系统设计与实现[D].北京交通大学, ,2021.
- [6] 郑朝霞.论开闭原则在 JAVA 程序设计中的应用[J].电脑知识与技术,2016,12(30):262-264.
- [7] 陈广义,吴继周,董德发等.模型系数的最小二乘法拟合[J].石油学报,1994(02):161-165.
- [8] 杨红伟,江涛,王励励.股市相关结构:动态演化过程与稳定性特征——基于随机矩阵理论与偏相关系数矩阵方法对中国 2015 年股灾的分析[J].商业经济与管理,2018,No.320(06):83-97.
- [9] 杨杏丽. 基于正则化交叉验证的模型选择方法 [D]. 山西大学,2021.DOI:10.27284/d.cnki.gsxiu.2021.000465.
- [10] 李明,来国红,常晏鸣等.深度学习算法中不同优化器的性能分析[J].信息技术与信息化,2022,No.264(03):206-209.