*Biểu mẫu BTL01*

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP

**KHOA ĐIỆN TỬ**

**Bộ môn: Công nghệ Thông tin**.

**BÀI TẬP KẾT THÚC MÔN HỌC**

MÔN HỌC

**KHOA HỌC DỮ LIỆU**

Sinh viên: ĐINH TRƯỜNG AN. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

Lớp: K57-KMT. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

Giáo viên giảng dạy: NGUYỄN VĂN HUY. . . . . . . . . . . . . . . . . . . .

**Thái Nguyên – 2025**

*Biểu mẫu BTL02*

|  |  |
| --- | --- |
| **TRƯỜNG ĐHKTCN** | **CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM** |
| **KHOA ĐIỆN TỬ** | ***Độc lập - Tự do - Hạnh phúc*** |

**BÀI TẬP KẾT THÚC MÔN HỌC**

**MÔN HỌC: KHOA HỌC DỮ LIỆU**

BỘ MÔN : CÔNG NGHỆ THÔNG TIN

*Sinh viên: Đinh Trường An*

*Lớp*: K57KMT ……………………………………………. *Ngành:Kỹ Thuật Máy Tính*

*Giáo viên hướng dẫn:Nguyễn Văn Huy*

*Ngày giao đề :20/05/2025*  *Ngày hoàn thành :26/05/2025*

*Tên đề tài : Xây dựng web phân tích và dự giá cổ phiếu.*

*Yêu cầu :*

***Đầu vào:***

* *Dữ liệu từ Stock Market Data - Kaggle*

***Đầu ra:***

* *Dự báo giá cổ phiếu và biểu đồ giá theo thời gian.*

|  |
| --- |
| **GIÁO VIÊN HƯỚNG DẪN** |
| *(Ký và ghi rõ họ tên)* |

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

### Thái Nguyên, ngày….tháng…..năm 20....

## GIÁO VIÊN HƯỚNG DẪN

*(Ký ghi rõ họ tên)*

MỤC LỤC

[CHƯƠNG I. GIỚI THIỆU ĐẦU BÀI 6](#_Toc199395687)

[**1.1 Đặt vấn đề** 6](#_Toc199395688)

[**1.2 Mục tiêu đề tài** 6](#_Toc199395689)

[**1.3 Phạm vi và giới hạn** 7](#_Toc199395690)

[**1.4 Hướng tiếp cận và thách thức** 7](#_Toc199395691)

[CHƯƠNG II. CƠ SỞ DỮ LIỆU 9](#_Toc199395692)

[**2.1 Pandas – Thư viện xử lý dữ liệu** 9](#_Toc199395693)

[**2.2. Thư viện ta – Chỉ báo kỹ thuật tài chính** 9](#_Toc199395694)

[**2.3. XGBoost – Mô hình học máy** 10](#_Toc199395695)

[**2.4. Plotly – Trực quan hóa dữ liệu** 10](#_Toc199395696)

[**2.5 Flask – Xây dựng Web App** 10](#_Toc199395697)

[**2.6 Kiểu dữ liệu Python – List, Dict** 11](#_Toc199395698)

[**2.7 Dữ liệu sử dụng** 11](#_Toc199395699)

[CHƯƠNG III. THIẾT KẾ XÂY DỰNG CHƯƠNG TRÌNH 12](#_Toc199395700)

[**3.1. Sơ đồ khối hệ thống** 12](#_Toc199395701)

[**3.2 Sơ đồ khố các thuật toán chính** 13](#_Toc199395702)

[**3.3 Cấu trúc dữ liệu** 16](#_Toc199395703)

[**3.3.1. DataFrame chứa dữ liệu gốc về giá cổ phiếu** 16](#_Toc199395704)

[**3.3.2. Dữ liệu đầu vào cho mô hình dự báo** 17](#_Toc199395705)

[**3.3.3. Dữ liệu đầu ra từ mô hình dự báo** 17](#_Toc199395706)

[**3.4. Chương trình** 17](#_Toc199395707)

[***3.4.1. Hàm get\_tickers()*** 18](#_Toc199395708)

[***3.4.2. Hàm load\_data(ticker)*** 18](#_Toc199395709)

[***3.4.3. Hàm check\_stock\_active(df)*** 18](#_Toc199395710)

[***3.4.4. Hàm add\_technical\_indicators(df)*** 18](#_Toc199395711)

[***3.4.5. Hàm train\_xgboost(df)*** 19](#_Toc199395712)

[***3.4.6. Hàm forecast\_xgboost(df, model, features)*** 19](#_Toc199395713)

[***3.4.7. Hàm index() (Flask route /)*** 19](#_Toc199395714)

[***3.4.8. Hàm autocomplete() (Flask route /autocomplete)*** 20](#_Toc199395715)

[***3.4.9 Biểu đồ hiển thị:*** 20](#_Toc199395716)

[CHƯƠNG IV: THỰC NGHIỆM VÀ KẾT LUẬN 21](#_Toc199395717)

[**4.1. Thực nghiệm** 21](#_Toc199395718)

[***a) Môi trường thực nghiệm*** 21](#_Toc199395719)

[***b) Kịch bản thử nghiệm*** 21](#_Toc199395720)

[***c) Giao diện người dùng*** 21](#_Toc199395721)

[***d) Biểu đồ minh họa*** 22](#_Toc199395722)

[**4.2. Kết luận** 24](#_Toc199395723)

[***a) Đánh giá kết quả*** 24](#_Toc199395724)

[***b) Hạn chế*** 24](#_Toc199395725)

[***c) Định hướng phát triển*** 24](#_Toc199395726)

[LỜI CẢM ƠN 25](#_Toc199395727)

# **CHƯƠNG I. GIỚI THIỆU ĐẦU BÀI**

## **1.1 Đặt vấn đề**

Thị trường chứng khoán là một trong những lĩnh vực tài chính có sức hấp dẫn lớn với nhà đầu tư cá nhân và tổ chức. Tuy nhiên, biến động giá cổ phiếu thường xuyên và chịu ảnh hưởng từ nhiều yếu tố vĩ mô lẫn vi mô, dẫn đến việc dự đoán giá cổ phiếu là một thách thức lớn. Nhờ vào sự phát triển mạnh mẽ của lĩnh vực Khoa học Dữ liệu và các mô hình học máy (Machine Learning), việc xây dựng các hệ thống dự báo giá cổ phiếu ngày càng trở nên khả thi.

Trong bối cảnh đó, đề tài "Phân tích và dự báo giá cổ phiếu" nhằm xây dựng một ứng dụng web giúp người dùng xem và dự báo giá cổ phiếu của một công ty bất kỳ, dựa trên dữ liệu lịch sử. Người dùng có thể nhập mã cổ phiếu, xem biểu đồ giá theo thời gian và nhận được dự báo giá trong tương lai gần.

## **1.2 Mục tiêu đề tài**

* Phân tích dữ liệu lịch sử giá cổ phiếu từ bộ dữ liệu Stock Market Dataset (Kaggle).
* Tiền xử lý dữ liệu để phục vụ cho mô hình dự báo (làm sạch, chuyển đổi định dạng ngày, xử lý thiếu).
* Xây dựng các mô hình học máy phù hợp như trung bình trượt (SMA), hồi quy tuyến tính hoặc LSTM để dự báo giá cổ phiếu.
* Trực quan hóa biểu đồ giá cổ phiếu và kết quả dự báo bằng thư viện P
* Xây dựng giao diện web đơn giản, thân thiện người dùng bằng Flask.

## **1.3 Phạm vi và giới hạn**

***Phạm vi:***

* Sử dụng dữ liệu lịch sử giá cổ phiếu từ Kaggle (Stock Market Dataset).
* Tập trung dự báo giá cổ phiếu trong ngắn hạn (1-5 ngày tới).
* Áp dụng các phương pháp học máy cổ điển (SMA, Linear Regression), hoặc mô hình deep learning cơ bản (LSTM nếu có thời gian).

***Giới hạn:***

* Không tính đến yếu tố tin tức tài chính, tâm lý thị trường hay thông tin ngoài dữ liệu định lượng.
* Mô hình chỉ mang tính chất tham khảo, không dùng để đầu tư tài chính thật.
* Website mang tính minh họa cho dự án khoa học dữ liệu, chưa tối ưu trải nghiệm người dùng và bảo mật.

## **1.4 Hướng tiếp cận và thách thức**

***Hướng tiếp cận:***

* Phân tích dữ liệu cổ phiếu theo từng mã và thời gian.
* Lựa chọn đặc trưng phù hợp như: giá đóng cửa, giá mở cửa, khối lượng giao dịch...
* Áp dụng kỹ thuật trung bình trượt (SMA), hồi quy, hoặc LSTM để dự báo.
* Xây dựng mô hình thành một ứng dụng web Flask, với autocomplete mã cổ phiếu và biểu đồ trực quan bằng Plotly.

***Thách thức:***

* Dữ liệu cổ phiếu không ổn định, có nhiều mã đã ngừng giao dịch, gây lỗi khi dự báo.
* Dự báo thời gian là bài toán phức tạp, đòi hỏi xử lý chuỗi thời gian kỹ lưỡng.
* Lựa chọn độ dài cửa sổ (window) hợp lý khi dùng mô hình chuỗi thời gian.
* Tối ưu hiệu suất web app khi hiển thị biểu đồ động và gọi mô hình dự báo.

# **CHƯƠNG II. CƠ SỞ DỮ LIỆU**

Trong chương trình dự báo giá cổ phiếu, nhóm đã sử dụng nhiều kiến thức và công cụ trong lĩnh vực lập trình Python, khoa học dữ liệu và học máy. Dưới đây là các nội dung chuyên môn chính được áp dụng.

## **2.1 Pandas – Thư viện xử lý dữ liệu**

Pandas là thư viện mạnh mẽ trong Python giúp xử lý và phân tích dữ liệu dạng bảng. Trong chương trình này, Pandas được dùng để:

* ***Đọc và xử lý dữ liệu***: Đọc các file CSV chứa thông tin giá cổ phiếu theo ngày.
* ***Làm sạch dữ liệu***: Xử lý các giá trị bị thiếu, trùng lặp hoặc không hợp lệ.
* ***Tạo bảng dữ liệu***: Dữ liệu được lưu trong các đối tượng DataFrame – cấu trúc bảng có dòng và cột dễ truy xuất.
* ***Tính toán đặc trưng***: Tính các chỉ báo kỹ thuật như trung bình động (SMA, EMA), chỉ số RSI,...
* ***Trích xuất thông tin***: Lọc dữ liệu theo mã cổ phiếu hoặc theo thời gian để phục vụ phân tích và dự báo.

## **2.2. Thư viện ta – Chỉ báo kỹ thuật tài chính**

Đây là thư viện tính toán các chỉ báo kỹ thuật phổ biến trong tài chính, được dùng để tạo các đặc trưng đầu vào cho mô hình dự báo. Cụ thể:

* **SMA (Simple Moving Average) –** Trung bình giá cổ phiếu trong một khoảng thời gian.
* **EMA (Exponential Moving Average) –** Trung bình có trọng số theo thời gian.
* **RSI (Relative Strength Index) –** Đánh giá mức độ quá mua hoặc quá bán**.**

Những đặc trưng này giúp mô hình học được xu hướng tăng/giảm và độ mạnh của thị trường.

## **2.3. XGBoost – Mô hình học máy**

XGBoost là mô hình học máy mạnh, tối ưu cho bài toán hồi quy, được dùng trong chương trình để dự đoán giá cổ phiếu ngày kế tiếp.

* **Đầu vào**: Các đặc trưng kỹ thuật tính từ dữ liệu lịch sử giá và khối lượng.
* **Đầu ra**: Giá cổ phiếu dự đoán trong phiên tiếp theo.
* **Ưu điểm**: Hiệu quả cao, kiểm soát quá khớp tốt, dễ điều chỉnh tham số.

## **2.4. Plotly – Trực quan hóa dữ liệu**

Plotly là thư viện tạo biểu đồ tương tác, dùng để hiển thị dữ liệu tài chính theo thời gian:

* **Biểu đồ đường (Line chart)**: Hiển thị giá cổ phiếu theo ngày.
* **Biểu đồ nến (Candlestick)**: Biểu diễn chi tiết giá mở, cao, thấp, đóng.
* **Ghi chú giá dự báo**: Biểu tượng đặc biệt đánh dấu giá dự báo để người dùng dễ theo dõi.

Các biểu đồ giúp người dùng hiểu rõ xu hướng và nhận diện các mốc giá quan trọng.

## **2.5 Flask – Xây dựng Web App**

Flask là một framework web nhẹ, được dùng để triển khai chương trình dự báo lên giao diện người dùng.

* **Nhận input từ người dùng**: Mã cổ phiếu cần dự báo.
* **Hiển thị kết quả**: Biểu đồ và giá dự báo.
* **Chức năng hỗ trợ**: Autocomplete mã cổ phiếu, kiểm tra mã đã dừng giao dịch.

Flask giúp kết nối mô hình học máy với người dùng thông qua trình duyệt web.

## **2.6 Kiểu dữ liệu Python – List, Dict**

Các kiểu dữ liệu cơ bản trong Python cũng được sử dụng thường xuyên:

* **List**: Lưu danh sách các mã cổ phiếu, danh sách giá theo ngày.
* **Dict (Dictionary)**: Dùng để tra cứu nhanh các thông tin như giá hiện tại, thông tin giao dịch, kết quả dự báo,...

## **2.7 Dữ liệu sử dụng**

Chương trình sử dụng dữ liệu lịch sử giá cổ phiếu từ bộ dữ liệu "Stock Market Dataset" trên Kaggle. Dữ liệu bao gồm:

* Mã cổ phiếu
* Ngày giao dịch
* Giá mở, cao, thấp, đóng
* Khối lượng giao dịch

Tập dữ liệu được chia theo từng công ty, mỗi file CSV tương ứng với một mã cổ phiếu.

# **CHƯƠNG III. THIẾT KẾ XÂY DỰNG CHƯƠNG TRÌNH**

## **3.1. Sơ đồ khối hệ thống**

Hệ thống được chia thành các module chính như sau:

* ***Module giao diện người dùng (UI)****:* Sử dụng Flask để xây dựng giao diện web, cho phép người dùng nhập mã cổ phiếu cần dự báo và nhận kết quả trực quan.
* ***Module xử lý dữ liệu***:
  + Đọc dữ liệu từ tệp .csv (dữ liệu giá cổ phiếu).
  + Làm sạch dữ liệu, xử lý thời gian, chuẩn hóa ngày tháng.
* ***Module tính toán chỉ báo kỹ thuật***: Sử dụng thư viện ta để tính RSI, SMA, EMA, sự thay đổi giá và khối lượng.
* ***Module huấn luyện mô hình ML***: Sử dụng XGBoost để huấn luyện trên tập dữ liệu đặc trưng kỹ thuật đã xử lý.
* ***Module dự báo và hiển thị***:
  + Thực hiện dự báo giá cổ phiếu trong ngày tiếp theo.
  + Hiển thị biểu đồ bằng Plotly (giá thực tế và vùng dự báo).

**Biểu đồ phân cấp chức năng**:

A diagram of a diagram

AI-generated content may be incorrect.



## **3.2 Sơ đồ khố các thuật toán chính**

Dưới đây là sơ đồ khối thể hiện các thuật toán chính trong chương trình và mối quan hệ :

A diagram of a diagram

AI-generated content may be incorrect.

**Mô tả các thuật toán chính**:

*a. Thuật toán xử lý dữ liệu:*

* Đầu vào: File CSV mã cổ phiếu.
* Xử lý:
  + Chuyển cột Date sang định dạng thời gian chuẩn.
  + Sắp xếp theo ngày tăng dần.
  + Loại bỏ các dòng thiếu thông tin (NaN).
* Đầu ra: DataFrame sạch và có thể dùng.

*b. Thuật toán tính chỉ báo kỹ thuật:*

* Tính toán các chỉ báo kỹ thuật như:
  + SMA 5 ngày.
  + EMA 12 ngày.
  + RSI 14 ngày.
  + Thay đổi giá và khối lượng.
* Đầu vào: DataFrame đã làm sạch.
* Đầu ra: DataFrame có thêm các đặc trưng kỹ thuật.

*c. Thuật toán huấn luyện mô hình:*

* Dùng **XGBoost Regressor**.
* Huấn luyện dựa trên đặc trưng kỹ thuật để dự báo giá ngày tiếp theo.
* Đầu vào: Đặc trưng (X) và nhãn (y = Close dịch chuyển -1).
* Đầu ra: Mô hình đã huấn luyện.

*d. Thuật toán dự báo:*

* Lấy dòng đặc trưng mới nhất từ DataFrame.
* Dự đoán giá tiếp theo bằng mô hình XGBoost đã huấn luyện.

*e. Thuật toán hiển thị biểu đồ:*

* Vẽ biểu đồ đường giá Close.
* Thêm vùng giá dự báo (±1%).
* Thêm điểm dự báo trên biểu đồ.

## **3.3 Cấu trúc dữ liệu**

Chương trình sử dụng cấu trúc dữ liệu chính là DataFrame từ thư viện Pandas để lưu trữ và xử lý dữ liệu giá cổ phiếu theo từng ngày. Dữ liệu đầu vào được lấy từ tập dữ liệu chứa thông tin giá cổ phiếu theo thời gian của nhiều công ty khác nhau. Cấu trúc dữ liệu trong chương trình có thể được chia thành ba phần chính như sau:

### **3.3.1. DataFrame chứa dữ liệu gốc về giá cổ phiếu**

Dữ liệu gốc được đọc từ một file CSV hoặc từ một nguồn dữ liệu có cấu trúc tương tự, với mỗi dòng đại diện cho một ngày giao dịch. Các cột điển hình bao gồm:

* **Name**: Mã cổ phiếu (ví dụ: AAPL, MSFT, TSLA).
* **Date**: Ngày giao dịch (dạng chuỗi hoặc datetime).
* **Open**: Giá mở cửa.
* **High**: Giá cao nhất trong ngày.
* **Low**: Giá thấp nhất trong ngày.
* **Close**: Giá đóng cửa (dùng làm mục tiêu dự báo).
* **Volume**: Khối lượng giao dịch.

Sau khi đọc vào, dữ liệu sẽ được tiền xử lý bằng cách:

* Chuyển đổi Date sang định dạng ngày tháng.
* Sắp xếp dữ liệu theo thời gian.
* Lọc ra các cổ phiếu còn hoạt động.
* Xử lý các giá trị bị thiếu hoặc bất thường.

### **3.3.2. Dữ liệu đầu vào cho mô hình dự báo**

Chương trình sử dụng một mô hình đơn giản là trung bình trượt (Simple Moving Average) để dự báo giá cổ phiếu trong tương lai. Đầu vào cho mô hình là một chuỗi giá đóng cửa (Close) theo thời gian của một mã cổ phiếu nhất định.

Ví dụ:

* **Ngày**: [2024-01-01, 2024-01-02, ..., 2024-01-30]
* **Giá đóng cửa**: [125.0, 126.3, 127.2, ..., 135.4]

Từ đó, mô hình sẽ tính trung bình của một số ngày gần nhất (ví dụ: 5 ngày, 10 ngày) để dự đoán giá của ngày tiếp theo.

### **3.3.3. Dữ liệu đầu ra từ mô hình dự báo**

Sau khi thực hiện dự báo, chương trình trả về:

* **Ngày được dự báo**: Ví dụ: [2024-01-31]
* **Giá cổ phiếu dự báo**: Ví dụ: [136.2]

Ngoài ra, chương trình còn xuất ra:

* Biểu đồ giá cổ phiếu thực tế kết hợp với giá dự báo.
* Bảng dữ liệu bao gồm cả giá thật và giá dự báo.
* Cảnh báo nếu cổ phiếu không còn giao dịch (dựa vào ngày cuối cùng trong dữ liệu).

## **3.4. Chương trình**

Chương trình dự báo giá cổ phiếu được xây dựng bằng Python với Flask làm backend, sử dụng mô hình XGBoost để dự báo giá trong ngắn hạn. Chương trình gồm các phần chính như sau:

### ***3.4.1. Hàm get\_tickers()***

* **Chức năng**: Trích xuất danh sách các mã cổ phiếu từ thư mục chứa file CSV.
* **Đầu vào**: Không có.
* **Đầu ra**: Danh sách các mã cổ phiếu hiện có trong dữ liệu.
* **Chi tiết**: Tên mã được lấy từ tên file CSV trong thư mục nasdaq/csv.

### ***3.4.2. Hàm load\_data(ticker)***

* **Chức năng**: Tải dữ liệu lịch sử giá cổ phiếu từ file CSV tương ứng.
* **Đầu vào**: Mã cổ phiếu (ticker).
* **Đầu ra**: DataFrame chứa dữ liệu với cột Date, Close, Volume,...
* **Xử lý**: Chuyển đổi định dạng ngày, sắp xếp theo thời gian tăng dần.

### ***3.4.3. Hàm check\_stock\_active(df)***

* **Chức năng**: Kiểm tra xem cổ phiếu còn đang giao dịch hay đã ngừng.
* **Đầu vào**: DataFrame chứa lịch sử cổ phiếu.
* **Đầu ra**: True/False cùng ngày cuối cùng cổ phiếu có biến động giá.
* **Chi tiết**: Nếu giá không thay đổi kể từ trước ngày 01/06/2022 thì được coi là đã ngừng giao dịch.

### ***3.4.4. Hàm add\_technical\_indicators(df)***

* **Chức năng**: Tính toán các chỉ báo kỹ thuật cho mô hình học máy.
* **Đầu vào**: DataFrame gốc.
* **Đầu ra**: DataFrame mới có thêm các cột:
  + **SMA\_5**: Trung bình trượt 5 phiên.
  + **EMA\_12**: Trung bình luỹ thừa 12 phiên.
  + **RSI\_14**: Chỉ số sức mạnh tương đối 14 phiên.
  + **Price\_Change**: Mức thay đổi giá theo phần trăm.
  + **Volume\_Change**: Mức thay đổi khối lượng theo phần trăm.

### ***3.4.5. Hàm train\_xgboost(df)***

* **Chức năng**: Huấn luyện mô hình XGBoost để dự báo giá đóng cửa phiên tiếp theo.
* **Đầu vào**: DataFrame dữ liệu lịch sử đã xử lý.
* **Đầu ra**: Mô hình đã huấn luyện cùng danh sách các đặc trưng.
* **Chi tiết**: Mục tiêu là giá đóng cửa ngày kế tiếp (Close.shift(-1)).

### ***3.4.6. Hàm forecast\_xgboost(df, model, features)***

* **Chức năng**: Sử dụng mô hình XGBoost đã huấn luyện để dự báo giá phiên tiếp theo.
* **Đầu vào**: Dữ liệu, mô hình và danh sách đặc trưng.
* **Đầu ra**: Giá dự báo cho phiên tiếp theo.

### ***3.4.7. Hàm index() (Flask route /)***

* **Chức năng**: Là endpoint chính của trang web. Cho phép người dùng nhập mã cổ phiếu để xem dự báo.
* **Đầu vào**: Dữ liệu từ form HTML (symbol).
* **Xử lý**:
  + Tải dữ liệu cổ phiếu.
  + Kiểm tra tình trạng giao dịch.
  + Huấn luyện mô hình và dự báo giá.
  + Tạo biểu đồ Plotly hiển thị giá cổ phiếu và vùng dự báo.
* **Đầu ra**: Giao diện web hiển thị biểu đồ và thông tin dự báo.

### ***3.4.8. Hàm autocomplete() (Flask route /autocomplete)***

* **Chức năng**: Hỗ trợ tính năng gợi ý mã cổ phiếu khi người dùng nhập.
* **Đầu vào**: Từ khóa tìm kiếm (q) từ AJAX request.
* **Đầu ra**: JSON chứa 10 mã cổ phiếu khớp với từ khóa.

### ***3.4.9 Biểu đồ hiển thị:***

Biểu đồ sử dụng thư viện **Plotly** để minh hoạ:

* Giá đóng cửa qua thời gian (Close).
* Điểm dự báo phiên tiếp theo (marker).
* Vùng dự báo (±1% xung quanh giá dự báo) thể hiện mức độ dao động.

# **CHƯƠNG IV: THỰC NGHIỆM VÀ KẾT LUẬN**

## **4.1. Thực nghiệm**

### ***a) Môi trường thực nghiệm***

* Ngôn ngữ lập trình: Python 3.10
* Thư viện sử dụng:
  + Flask (Web framework)
  + pandas, numpy (Xử lý dữ liệu)
  + XGBoost (Mô hình dự báo)
  + ta (Chỉ báo kỹ thuật)
  + plotly (Vẽ biểu đồ)
* Bộ dữ liệu: Stock Market Dataset (NASDAQ) lấy từ Kaggle, gồm các file CSV tương ứng từng mã cổ phiếu.

### ***b) Kịch bản thử nghiệm***

Tiến hành thử nghiệm trên một số mã cổ phiếu phổ biến như:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mã cổ phiếu | Ngày cuối cùng | Giá đóng cửa | Giá dự báo | Giao dịch còn hoạt động không |
| AAPL | 29-12-2023 | 193.97 | 195.12 | Có |
| GOOG | 29-12-2023 | 140.93 | 142.04 | Có |
| PYCZ | 01-07-2021 | 12.03 |  | Không |

### ***c) Giao diện người dùng***

Trang web được xây dựng với giao diện đơn giản, cho phép người dùng:

* Nhập mã cổ phiếu (có hỗ trợ gợi ý).
* Nhấn nút “Dự báo” để nhận kết quả.
* Xem biểu đồ minh họa:
  + Đường giá cổ phiếu theo thời gian.
  + Điểm dự báo tiếp theo (biểu tượng tam giác màu xanh).
  + Vùng dự báo ±1% thể hiện độ không chắc chắn.

### ***d) Biểu đồ minh họa***

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.1.1. Giao diện Web

A screen shot of a graph

AI-generated content may be incorrect.

Hình 4.1.2. Bảng kết quả hiển thị dữ liệu và dự đoán về một cổ phiếu V.

A screen shot of a graph

AI-generated content may be incorrect.

Hình 4.1.3. Bảng kết quả hiển thị dữ liệu của một cố phiếu đã dùng giao dịch

## **4.2. Kết luận**

### ***a) Đánh giá kết quả***

* Hệ thống có thể thực hiện dự báo ngắn hạn giá cổ phiếu dựa trên các đặc trưng kỹ thuật cơ bản.
* Mô hình XGBoost tỏ ra hiệu quả trong việc học quan hệ phi tuyến giữa chỉ báo kỹ thuật và giá.
* Giao diện thân thiện, dễ sử dụng, hỗ trợ gợi ý mã và cảnh báo cổ phiếu ngừng giao dịch.

### ***b) Hạn chế***

* Chưa sử dụng dữ liệu vĩ mô hoặc tin tức ảnh hưởng đến giá cổ phiếu.
* Mô hình chỉ dự báo giá cho phiên tiếp theo, chưa hỗ trợ chuỗi ngày hoặc dự báo dài hạn.
* Vùng dự báo cố định ±1% là đơn giản, chưa phản ánh đúng độ bất định của mô hình.

### ***c) Định hướng phát triển***

* Áp dụng các mô hình nâng cao hơn như LSTM, Transformer cho chuỗi thời gian.
* Bổ sung phân tích rủi ro, confidence interval hoặc các mô hình dự báo phân phối.
* Tích hợp API cập nhật dữ liệu theo thời gian thực (ví dụ từ Yahoo Finance, Alpha Vantage).
* Mở rộng giao diện thành dashboard quản lý danh mục cổ phiếu cá nhân.

# **LỜI CẢM ƠN**

Trong quá trình hoàn thành đề tài này, em nhận thấy bản thân mình đã học hỏi được rất nhiều điều quý giá, không chỉ về kiến thức chuyên môn mà còn về sự kiên trì và trách nhiệm trong công việc. Trước hết, em xin gửi lời cảm ơn sâu sắc nhất đến Thầy Nguyễn Văn Huy – người thầy đã luôn tận tình hướng dẫn, chỉ bảo và truyền cảm hứng cho em. Thầy không chỉ giúp em hiểu sâu hơn về các kiến thức liên quan đến đề tài mà còn động viên em vượt qua những thời điểm khó khăn, thử thách trong suốt quá trình làm việc. Những góp ý chân thành và sự hỗ trợ nhiệt tình của Thầy chính là động lực lớn giúp em hoàn thiện báo cáo này một cách tốt nhất.

Bên cạnh đó, em cũng xin gửi lời cảm ơn đến các bạn bè và đồng nghiệp đã luôn đồng hành, chia sẻ kinh nghiệm và giúp đỡ em mỗi khi gặp khó khăn trong việc nghiên cứu và lập trình. Sự hỗ trợ quý báu từ mọi người đã tạo nên một môi trường học tập và làm việc tích cực, giúp em có thể hoàn thành dự án này suôn sẻ và hiệu quả hơn.

Em hy vọng rằng những gì mình đã học được và hoàn thành trong đề tài này sẽ là bước đệm quan trọng cho những bước tiến tiếp theo trên con đường học vấn và nghề nghiệp của bản thân.