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: Đặng Trung Dũng. . . . . . **. . . . .**

Lớp: K57KMT.01. . . . . . **. . . . . .**

Giáo viên giảng dạy: Nguyễn Văn Huy . . . .

Link GitHub: https://github.com/Trungdung090/BTL_KHDL

THÁI NGUYÊN - 2025

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

Bộ Môn: Công Nghệ Thông Tin

Môn Học: Khoa Học Dữ Liệu

Sinh viên: Đặng Trung Dũng MSSV: K215480106138

Ngành: Kỹ Thuật Máy Tính

Lớp: K57KMT.01 Khoá: 2021-2026

Giáo viên hướng dẫn: Nguyễn Văn Huy

Ngày giao đề: 19/05/2025 Ngày hoàn thành: 30/05/2025

1. Tên đề tài: Web app dự báo giá cổ phiếu.

2. Yêu cầu:

Đầu vào:

- 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 2025

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

MỤC LỤC

DANH MỤC CÁC BẢNG VÀ HÌNH VẼ, ĐỒ THỊ

LỜI NÓI ĐẦU

Trong thời đại công nghệ số phát triển mạnh mẽ, thị trường chứng khoán không chỉ là một sân chơi của các nhà đầu tư chuyên nghiệp mà còn thu hút sự quan tâm ngày càng lớn từ công chúng. Việc phân tích và dự báo giá cổ phiếu đang trở thành một nhu cầu thiết yếu nhằm hỗ trợ ra quyết định đầu tư chính xác và hiệu quả. Tuy nhiên, với lượng dữ liệu khổng lồ và biến động phức tạp của thị trường, việc đưa ra những dự báo chính xác không hề đơn giản, đòi hỏi sự hỗ trợ của các công cụ công nghệ hiện đại và các mô hình học máy.

Xuất phát từ thực tiễn đó, đề tài "*Web app dự báo giá cổ phiếu*" được xây dựng nhằm mục tiêu thiết kế một ứng dụng web cho phép người dùng theo dõi, phân tích và dự báo xu hướng giá cổ phiếu theo thời gian. Ứng dụng sử dụng nguồn dữ liệu từ bộ dữ liệu thị trường chứng khoán trên Kaggle, kết hợp với các mô hình học máy và trực quan hóa dữ liệu để mang lại cái nhìn tổng quan và hỗ trợ người dùng đưa ra quyết định đầu tư.

Bài tiểu luận này sẽ trình bày các bước xây dựng hệ thống, từ tiền xử lý dữ liệu, lựa chọn mô hình dự báo phù hợp, đến thiết kế giao diện người dùng thân thiện. Qua đó, người đọc sẽ hiểu được quy trình phát triển một ứng dụng trí tuệ nhân tạo đơn giản nhưng hiệu quả trong lĩnh vực tài chính.

Mặc dù đã rất cố gắng để hoàn thành công việc, nhưng thời gian có hạn và thiếu kinh nghiệm cũng như kỹ năng chưa cao nên việc phân tích thiết kế còn nhiều thiếu sót, kính mong quý thầy cô và các bạn góp ý, bổ sung để em hoàn thiện bài tập tốt hơn nữa.

Em xin chân thành cảm ơn sự quan tâm, chỉ bảo của thầy Nguyễn Văn Huy cùng toàn thể các thầy cô giáo và các bạn trong trường đã giúp đỡ em hoàn thành bài tiểu luận này.

LỜI CẢM ƠN

Trong suốt quá trình học tập và thực hiện bài tập lớn, em đã nhận được sự giúp đỡ tận tình của thầy Nguyễn Văn Huy thuộc bộ môn Công Nghệ Thông Tin – Khoa Điện tử - Trường Đại học Kỹ thuật Công Nghiệp – Đại học Thái Nguyên. Em bày tỏ lòng biết ơn thầy đã tận tình giúp đỡ, hướng dẫn em trong thời gian thực hiện đề tài này.

Mặc dù đã cố gắng hết sức, song do điều kiện thời gian và kinh nghiệm thực tế còn ít, cho nên đề tài không thể tránh khỏi thiếu sót. Vì vậy, em rất mong nhận được sự đóng góp ý kiến của các thầy giáo, cô giáo và các bạn.

Em xin chân thành cảm ơn!

Sinh viên thực hiện

Đặng Trung Dũng

CHƯƠNG I: GIỚI THIỆU CHUNG

1.1. Giới thiệu đề tài

Trong bối cảnh thị trường tài chính biến động không ngừng, việc dự báo giá cổ phiếu đóng vai trò quan trọng giúp nhà đầu tư đưa ra quyết định đúng đắn. Đề tài này hướng đến việc xây dựng một web app trực quan và dễ sử dụng, hỗ trợ người dùng tải lên dữ liệu lịch sử giá cổ phiếu, huấn luyện mô hình, và đưa ra dự báo về giá cổ phiếu trong tương lai. Ứng dụng này kết hợp các mô hình học máy (Machine Learning) truyền thống với các chỉ số kỹ thuật để cung cấp cái nhìn tổng quan và dự đoán xu hướng giá cổ phiếu.

Mục tiêu chính:

- Thiết kế và phát triển một web app với giao diện thân thiện, cho phép người dùng kiểm tra dự báo giá cổ phiếu theo thời gian thực.
- Áp dụng các thuật toán dự báo trong Data Science để xử lý và phân tích dữ liệu chuỗi thời gian.
- Tích hợp các chức năng trực quan hóa dữ liệu nhằm giúp người dùng dễ dàng nắm bắt xu hướng và đưa ra quyết định đầu tư.

1.2. Tính năng của chương trình

- Tải lên dữ liệu CSV: Người dùng có thể dễ dàng tải lên các tệp dữ liệu giá cổ phiếu định dạng CSV. Ứng dụng sẽ tự động xử lý và chuẩn hóa dữ liệu, bao gồm đổi tên cột (nếu cần) và chuyển đổi định dạng ngày tháng.
- Xử lý và chuẩn hóa dữ liệu:
  - Đảm bảo các cột cần thiết như 'date', 'open', 'high', 'low', 'close', 'volume' được định dạng đúng.
  - Xử lý các giá trị thiếu (NaN) bằng cách sử dụng phương pháp ffill (điền xuôi) và bfill (điền ngược) để đảm bảo tính liên tục của dữ liệu.
  - Kiểm tra số lượng dữ liệu đầu vào, yêu cầu tối thiểu 100 dòng để đảm bảo đủ dữ liệu cho việc tính toán và huấn luyện mô hình.
- Tính toán các chỉ số kỹ thuật: Chương trình tự động tính toán một số chỉ số kỹ thuật phổ biến để làm cơ sở cho việc dự báo:

  - SMA (Simple Moving Average): Trung bình động đơn giản (20 và 50 ngày).
  - EMA (Exponential Moving Average): Trung bình động hàm mũ (20 ngày).
  - RSI (Relative Strength Index): Chỉ số sức mạnh tương đối (14 ngày).
  - Bollinger Bands: Dải Bollinger (đường giữa, cận trên, cận dưới).
  - MACD (Moving Average Convergence Divergence): Đường trung bình động hội tụ phân kỳ.
- Huấn luyện mô hình dự báo:
  - Chương trình sử dụng hai mô hình học máy chính: Linear Regression (Hồi quy tuyến tính) và Random Forest Regressor (Hồi quy rừng ngẫu nhiên).
  - Dữ liệu được chia thành tập huấn luyện và tập kiểm tra (80/20).
  - Dữ liệu đầu vào (features) được chuẩn hóa bằng MinMaxScaler.
  - Sau khi huấn luyện, các chỉ số đánh giá mô hình như MSE (Mean Squared Error), MAE (Mean Absolute Error) và RMSE (Root Mean Squared Error) được hiển thị.
- Dự báo giá cổ phiếu trong tương lai:
  - Người dùng có thể chọn mô hình dự báo (mặc định là Random Forest) và số ngày muốn dự báo (tối đa 90 ngày).
  - Chương trình sử dụng dữ liệu lịch sử gần nhất để dự báo các ngày tiếp theo một cách tuần tự (rolling prediction).
- Trực quan hóa dữ liệu và dự báo:
  - Biểu đồ tương tác được tạo bằng thư viện Plotly, hiển thị giá cổ phiếu thực tế, các chỉ số kỹ thuật như SMA, và đường giá dự báo.
  - Biểu đồ được thiết kế với giao diện thân thiện, dễ đọc, cho phép người dùng xem chi tiết giá trị tại từng điểm dữ liệu.
- Thống kê tổng quan: Cung cấp các số liệu thống kê cơ bản về dữ liệu cổ phiếu đã tải lên, bao gồm tổng số ngày, giá trung bình, giá cao nhất, giá thấp nhất, khối lượng giao dịch trung bình, và phần trăm thay đổi giá.

1.3. Thách thức và giải pháp

1.3.1. Thách thức

Xử lý dữ liệu đa dạng: Dữ liệu CSV về cổ phiếu có thể có nhiều định dạng khác nhau về tên cột, kiểu dữ liệu, và giá trị thiếu. Việc chuẩn hóa và xử lý lỗi dữ liệu là rất quan trọng để đảm bảo tính đúng đắn cho mô hình. Thách thức này được giải quyết bằng cách kiểm tra và đổi tên cột, xử lý giá trị NaN và đảm bảo các cột cần thiết tồn tại.

Thiếu dữ liệu: Mô hình học máy yêu cầu một lượng dữ liệu đủ lớn để có thể huấn luyện hiệu quả. Nếu tệp CSV được tải lên có quá ít dữ liệu (dưới 100 dòng), ứng dụng sẽ thông báo lỗi.

Lựa chọn và tối ưu mô hình: Việc lựa chọn mô hình phù hợp và tối ưu hóa các siêu tham số là một thách thức. Random Forest Regressor thường hoạt động tốt với dữ liệu chuỗi thời gian do khả năng nắm bắt các mối quan hệ phi tuyến tính. Tuy nhiên, việc đánh giá hiệu suất của các mô hình khác nhau (Linear Regression) là cần thiết để đưa ra lựa chọn tốt nhất.

Dự báo chuỗi thời gian: Dự báo giá cổ phiếu là một bài toán chuỗi thời gian, nơi giá trị dự báo cho ngày tiếp theo phụ thuộc vào dữ liệu của các ngày trước đó. Việc xây dựng features từ dữ liệu lịch sử (ví dụ: 30 ngày trước đó) và các chỉ số kỹ thuật đòi hỏi sự cẩn trọng để đảm bảo tính chính xác.

Trực quan hóa phức tạp: Biểu đồ giá cổ phiếu với nhiều đường dữ liệu (giá thực tế, SMA, dự báo) đòi hỏi một thư viện trực quan hóa mạnh mẽ như Plotly để tạo ra các biểu đồ tương tác, dễ hiểu.

Tối ưu hiệu suất: Với việc xử lý dữ liệu lớn và các mô hình tính toán phức tạp, hiệu suất của ứng dụng cần được tối ưu để đảm bảo thời gian phản hồi nhanh chóng. Việc sử dụng *n_jobs=-1* trong *RandomForestRegressor* giúp tận dụng đa luồng xử lý.

1.3.2. Giải pháp

- **Thu thập và tiền xử lý dữ liệu**:
  - Đọc và xử lý tệp CSV bằng thư viện Pandas.
  - Chuyển đổi kiểu dữ liệu (ví dụ: pd.to_datetime).
  - Xử lý giá trị thiếu (NaN) và ngoại lệ.
  - Chuẩn hóa tên cột.
- **Kỹ thuật đặc trưng:**

- Tạo ra các chỉ số kỹ thuật (SMA, EMA, RSI, Bollinger Bands, MACD) từ dữ liệu giá và khối lượng để cung cấp thông tin hữu ích cho mô hình.
- Xây dựng các đặc trưng (features) bằng cách sử dụng dữ liệu lịch sử của n ngày trước đó (ví dụ: 30 ngày).

- **Học máy (Machine Learning):**
  - Hồi quy (Regression): Bài toán dự báo giá cổ phiếu là một bài toán hồi quy, dự đoán một giá trị liên tục.
  - Mô hình Linear Regression: Một mô hình cơ bản nhưng hiệu quả để tìm ra mối quan hệ tuyến tính giữa các biến.
  - Mô hình Random Forest Regressor: Một mô hình ensemble mạnh mẽ, ít nhạy cảm với overfitting và có khả năng xử lý các mối quan hệ phi tuyến tính, thường cho kết quả tốt trong dự báo chuỗi thời gian.
  - Đánh giá mô hình: Sử dụng các chỉ số như MSE, MAE, RMSE để định lượng hiệu suất của các mô hình dự báo.
  - Chia tập dữ liệu (Train/Test): Phân chia dữ liệu thành tập huấn luyện và tập kiểm tra để đánh giá khả năng tổng quát hóa của mô hình.
  - Chuẩn hóa dữ liệu: Sử dụng MinMaxScaler để đưa các đặc trưng về cùng một phạm vi giá trị.
- **Trực quan hóa dữ liệu:**
  - Sử dụng thư viện Plotly để tạo biểu đồ đường tương tác, hiển thị xu hướng giá và các điểm dự báo.
  - Tùy chỉnh giao diện biểu đồ để nâng cao trải nghiệm người dùng.
- **Phát triển Web (Web Development):**
  - Sử dụng Flask framework để xây dựng backend của ứng dụng web, xử lý các yêu cầu HTTP (upload file, predict, get stats).
  - Sử dụng Jinja2 để render template HTML cho giao diện người dùng.
  - Sử dụng AJAX (Fetch API) để gửi và nhận dữ liệu giữa frontend và backend một cách không đồng bộ.

CHƯƠNG II: CƠ SỞ LÝ THUYẾT

Chương này sẽ trình bày các nội dung chuyên môn và các thư viện, công cụ được sử dụng để xây dựng ứng dụng web app dự báo giá cổ phiếu. Các kiến thức này là nền tảng để xử lý dữ liệu, xây dựng mô hình và trực quan hóa kết quả.

2.1. Ngôn ngữ lập trình và Framework

Hình 2.1.1. Ngôn ngữ lập trình Python

- **Python**: Là ngôn ngữ lập trình chính được sử dụng cho toàn bộ ứng dụng. Python được lựa chọn nhờ sự phong phú của các thư viện hỗ trợ khoa học dữ liệu, học máy và phát triển web.

Hình 2.1.2. Flask

- **Flask**: Là một micro-framework cho Python, được sử dụng để xây dựng phần backend của ứng dụng web. Flask nhẹ, linh hoạt và dễ dàng mở rộng, phù hợp cho việc phát triển các ứng dụng web nhỏ và vừa. Nó xử lý các yêu cầu HTTP (GET, POST), định tuyến URL và trả về các phản hồi JSON hoặc render các template HTML.

2.2. Xử lý và phân tích dữ liệu

- **Pandas**: Thư viện Pandas là công cụ không thể thiếu để làm việc với dữ liệu dạng bảng (tabular data). Trong chương trình, Pandas được sử dụng để:
  - Đọc và tải dữ liệu: Đọc các tệp CSV chứa dữ liệu giá cổ phiếu lịch sử.
  - Xử lý dữ liệu: Chuẩn hóa tên các cột ('Date' thành 'date', 'Close' thành 'close', v.v.).
  - Chuyển đổi kiểu dữ liệu: Chuyển đổi cột ngày tháng sang định dạng *datetime* để dễ dàng thao tác.
  - Xử lý dữ liệu thiếu: Sử dụng các phương pháp như *dropna()* để loại bỏ các hàng có giá trị NaN trong các cột quan trọng và *fillna(method='ffill').fillna(method='bfill')* để điền giá trị thiếu, đảm bảo tính liên tục của dữ liệu.
  - Sắp xếp dữ liệu: Sắp xếp dữ liệu theo ngày để đảm bảo đúng thứ tự thời gian.
  - Tính toán thống kê: Thực hiện các phép tính thống kê cơ bản như giá trung bình, giá cao nhất, giá thấp nhất, khối lượng trung bình và phần trăm thay đổi giá.
- **Numpy**: Thư viện Numpy cung cấp khả năng xử lý mảng và ma trận hiệu quả, là nền tảng cho nhiều phép toán số học trong khoa học dữ liệu. Trong chương trình, Numpy được sử dụng để:
  - Chuyển đổi dữ liệu: Chuyển đổi các danh sách Python thành mảng Numpy (*np.array*) để đưa vào mô hình học máy.
  - Tính toán toán học: Hỗ trợ các phép toán như căn bậc hai (*np.sqrt*) để tính RMSE.

2.3. Các chỉ số kỹ thuật

Các chỉ số kỹ thuật được tính toán từ dữ liệu giá và khối lượng lịch sử, giúp nhận diện xu hướng và tín hiệu giao dịch. Chúng được sử dụng làm các đặc trưng đầu vào cho mô hình học máy.

- **Simple Moving Average** (SMA - Trung bình động đơn giản): Tính giá trung bình của cổ phiếu trong một khoảng thời gian nhất định (ví dụ: 20 ngày, 50 ngày). SMA giúp làm mượt dữ liệu giá và xác định xu hướng.

Công thức:

$$SMA_n = \frac{P_1 + P_2 + \ldots + P_n}{n}$$

- **Exponential Moving Average** (EMA - Trung bình động hàm mũ): Tương tự SMA nhưng ưu tiên hơn các dữ liệu gần đây hơn. EMA phản ứng nhanh hơn với sự thay đổi giá.

Công thức:

$$EMA_n = \left(Close - EMA_{\text{prev}}\right) \times \text{Multiplier} + EMA_{\text{prev}}$$

- **Relative Strength Index** (RSI - Chỉ số sức mạnh tương đối): Đo lường tốc độ và mức độ thay đổi giá để đánh giá điều kiện mua quá mức (overbought) hoặc bán quá mức (oversold). Giá trị RSI thường nằm trong khoảng từ 0 đến 100.

Công thức:

$$RSI = 100 - \frac{100}{1 + RS}, \text{ trong đó } RS = \frac{\text{AverageGain}}{\text{AverageLoss}}$$

- **Bollinger Bands:** Gồm ba đường: đường trung bình động (đường giữa), và hai dải trên và dưới cách đường giữa một số độ lệch chuẩn. Bollinger Bands giúp đánh giá biến động giá và các điểm đảo chiều tiềm năng.

Công thức:

Middle Band = 20-day SMA

Upper Band = Middle Band + (2 x 20-day Standard Deviation)

Lower Band = Middle Band - (2 x 20-day Standard Deviation)[1]

- **Moving Average Convergence Divergence** (MACD - Đường trung bình động hội tụ phân kỳ): Là một chỉ báo động lượng theo xu hướng, cho thấy mối quan hệ giữa hai đường trung bình động của giá. Nó bao gồm đường MACD, đường tín hiệu và biểu đồ cột (histogram).

Công thức:

MACD Line = (12-day EMA of Close) - (26-day EMA of Close)

Signal Line = 9-day EMA of MACD Line

2.4. Các mô hình học máy

- **Hồi quy tuyến tính (Linear Regression):** Là một mô hình học máy cơ bản được sử dụng để mô hình hóa mối quan hệ tuyến tính giữa biến phụ thuộc (giá cổ phiếu) và một hoặc nhiều biến độc lập (các đặc trưng). Mặc dù đơn giản, nó cung cấp một đường cơ sở để đánh giá các mô hình phức tạp hơn.
- **Rừng ngẫu nhiên (Random Forest Regressor):** Là một mô hình học máy ensemble, xây dựng nhiều cây quyết định và lấy giá trị trung bình của các dự đoán từ các cây đó. Random Forest mạnh mẽ, ít bị quá khớp (overfitting) và có khả năng xử lý các mối quan hệ phi tuyến tính, làm cho nó trở thành lựa chọn tốt cho dự báo chuỗi thời gian.
    - Tham số *n_estimators=100* xác định số lượng cây trong rừng.
    - Tham số *n_jobs=-1* cho phép sử dụng tất cả các lõi CPU có sẵn để tăng tốc độ huấn luyện.
- **Chuẩn hóa dữ liệu** - MinMaxScaler: Được sử dụng để chuẩn hóa các đặc trưng đầu vào về cùng một phạm vi giá trị (thường từ 0 đến 1). Điều này giúp các thuật toán học máy hội tụ nhanh hơn và hoạt động ổn định hơn, đặc biệt với các thuật toán dựa trên khoảng cách hoặc độ dốc.

2.5. Đánh giá mô hình

Để đánh giá hiệu suất của các mô hình dự báo, các chỉ số sau được sử dụng:

- **Mean Squared Error (MSE - Sai số bình phương trung bình):** Đo lường trung bình của bình phương các sai số giữa giá trị dự đoán và giá trị thực tế. MSE càng nhỏ càng tốt.

  Công thức:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2$$

- **Mean Absolute Error (MAE - Sai số tuyệt đối trung bình):** Đo lường trung bình của các sai số tuyệt đối giữa giá trị dự đoán và giá trị thực tế. MAE ít nhạy cảm với các giá trị ngoại lệ hơn MSE.

  Công thức:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Y_i - \hat{Y}_i\right|$$

- **Root Mean Squared Error (RMSE - Căn bậc hai của sai số bình phương trung bình)**: Là căn bậc hai của MSE, có cùng đơn vị với biến phụ thuộc, giúp dễ dàng diễn giải hơn. RMSE càng nhỏ càng tốt.

  Công thức:

$$RMSE = \sqrt{MSE}$$

2.6. Trực quan hóa dữ liệu

- **Plotly**: Thư viện Plotly được sử dụng để tạo các biểu đồ tương tác, giàu thông tin.
  - Tạo biểu đồ đường hiển thị giá cổ phiếu thực tế, các đường SMA và đường giá dự báo.
  - Cung cấp khả năng tương tác như phóng to, thu nhỏ, di chuột để xem thông tin chi tiết từng điểm dữ liệu.
  - Xuất biểu đồ dưới dạng JSON để truyền tải và hiển thị trên giao diện web.

2.7. Các thư viện khác

- **Datetime và timedelta**: Được sử dụng để xử lý các đối tượng ngày tháng và tính toán các ngày trong tương lai cho dự báo.
- **Os và Werkzeug.utils.secure_filename**: Được sử dụng để xử lý việc tải lên tệp, tạo thư mục lưu trữ và bảo mật tên tệp để ngăn chặn các cuộc tấn công liên quan đến đường dẫn.
- **Json**: Để chuyển đổi dữ liệu Python sang định dạng JSON và ngược lại, phục vụ cho việc giao tiếp giữa backend và frontend của ứng dụng web.

CHƯƠNG III:THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH

Chương này trình bày chi tiết về kiến trúc, các module, thuật toán và cấu trúc dữ liệu được sử dụng để xây dựng ứng dụng web dự báo giá cổ phiếu.

3.1. Sơ đồ khối hệ thống

Hệ thống được thiết kế theo mô hình client-server, trong đó frontend (giao diện người dùng) tương tác với backend (server Flask) để xử lý dữ liệu và thực hiện các tác vụ dự báo.

3.1.1. Các modul chính trong chương trình

Giao diện người dùng: Là phần tương tác trực tiếp với người dùng, được xây dựng bằng HTML, CSS (Tailwind CSS) và JavaScript. Nó chịu trách nhiệm hiển thị form tải lên tệp, các tùy chọn dự báo, kết quả thống kê, kết quả huấn luyện mô hình và biểu đồ dự báo.

- **Chức năng chính:**
  - Cho phép người dùng tải lên tệp CSV.
  - Hiển thị trạng thái tải lên và thông báo lỗi.
  - Cung cấp các tùy chọn để chọn mô hình dự báo và số ngày dự báo.
  - Hiển thị biểu đồ giá cổ phiếu và dự báo tương tác.
  - Trình bày các số liệu thống kê về dữ liệu đã tải lên và kết quả đánh giá mô hình.

Server Flask: Là trái tim của ứng dụng, được xây dựng bằng Flask (Python). Nó đóng vai trò là API trung gian, nhận yêu cầu từ frontend, xử lý logic nghiệp vụ và trả về phản hồi cho frontend.

- **Chức năng chính:**
  - Xử lý yêu cầu HTTP (POST để tải lên tệp, POST để dự báo, GET để lấy thống kê).
  - Quản lý việc lưu trữ tạm thời các tệp CSV được tải lên.
  - Tương tác với *Module StockPredictor* để thực hiện các tác vụ xử lý dữ liệu, huấn luyện mô hình và dự báo.

- Chuyển đổi dữ liệu Python sang JSON để gửi về frontend.

Lớp StockPredictor: Đây là module chính chứa toàn bộ logic xử lý dữ liệu, tính toán các chỉ số kỹ thuật, huấn luyện mô hình học máy và thực hiện dự báo. Nó được thiết kế dưới dạng một lớp để quản lý trạng thái (dữ liệu, scaler, mô hình).

- **Chức năng chính:**
  - Tải và tiền xử lý dữ liệu cổ phiếu từ tệp CSV.
  - Tính toán các chỉ số kỹ thuật (SMA, EMA, RSI, Bollinger Bands, MACD).
  - Chuẩn bị dữ liệu cho huấn luyện mô hình (tạo đặc trưng, chia tập train/test, chuẩn hóa).
  - Huấn luyện các mô hình Linear Regression và Random Forest Regressor.
  - Đánh giá hiệu suất mô hình (MSE, MAE, RMSE).
  - Thực hiện dự báo giá cổ phiếu trong tương lai.
  - Tạo biểu đồ tương tác bằng Plotly.
  - Cung cấp các số liệu thống kê tổng quan về dữ liệu.

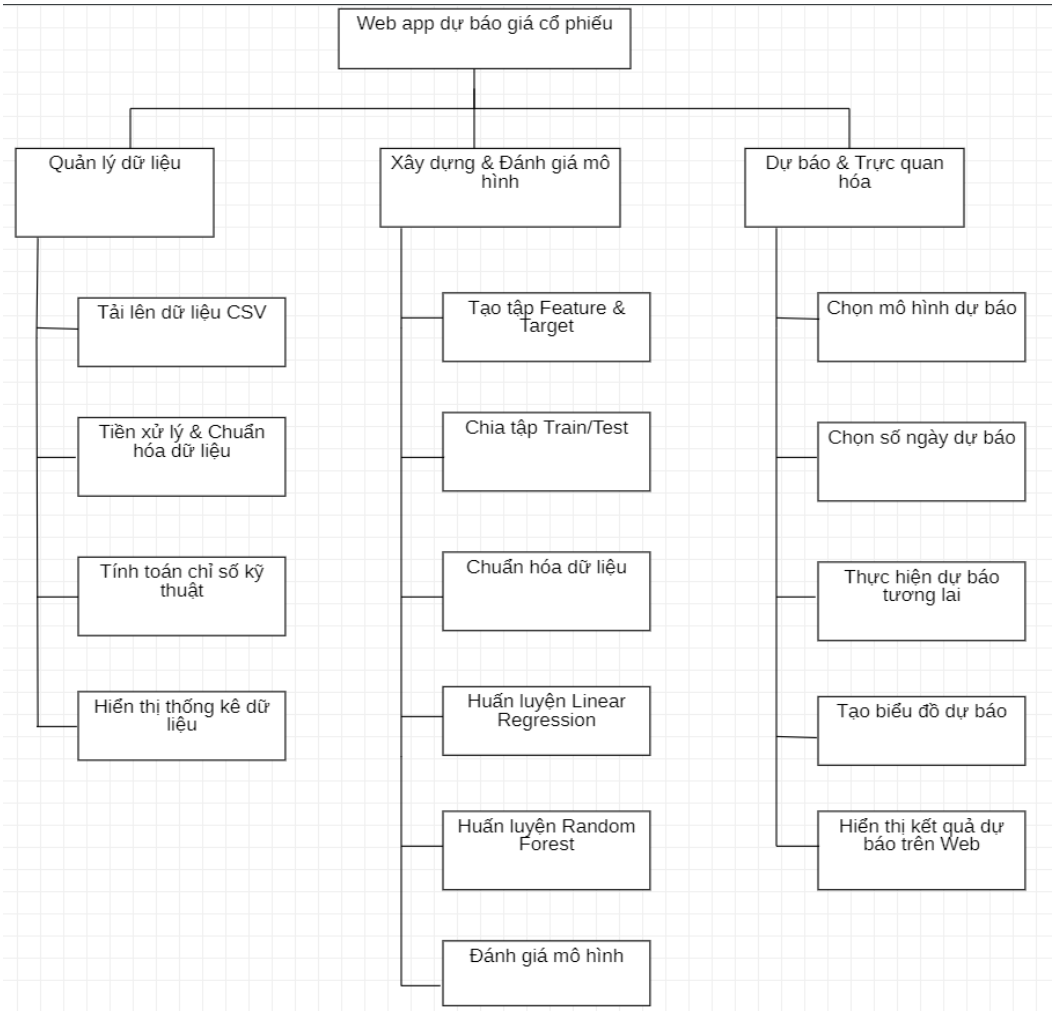
Thư mục Uploads: Là một thư mục trên server để lưu trữ tạm thời các tệp CSV do người dùng tải lên trước khi chúng được xử lý bởi *StockPredictor*.

Mô hình ML (LR, RF): Các đối tượng mô hình học máy (*LinearRegression, RandomForestRegressor*) sau khi được huấn luyện sẽ được lưu trữ trong *Module StockPredictor* để sẵn sàng cho việc dự báo.

Plotly: Thư viện dùng để tạo ra các đối tượng biểu đồ (graph objects) ở phía backend (Python) và sau đó được chuyển đổi sang định dạng JSON để frontend (JavaScript) render.

- **Chức năng chính:** Tạo biểu đồ line chart, candlestick chart (nếu mở rộng), với các trục, chú thích, dải chỉ số kỹ thuật và đường dự báo.

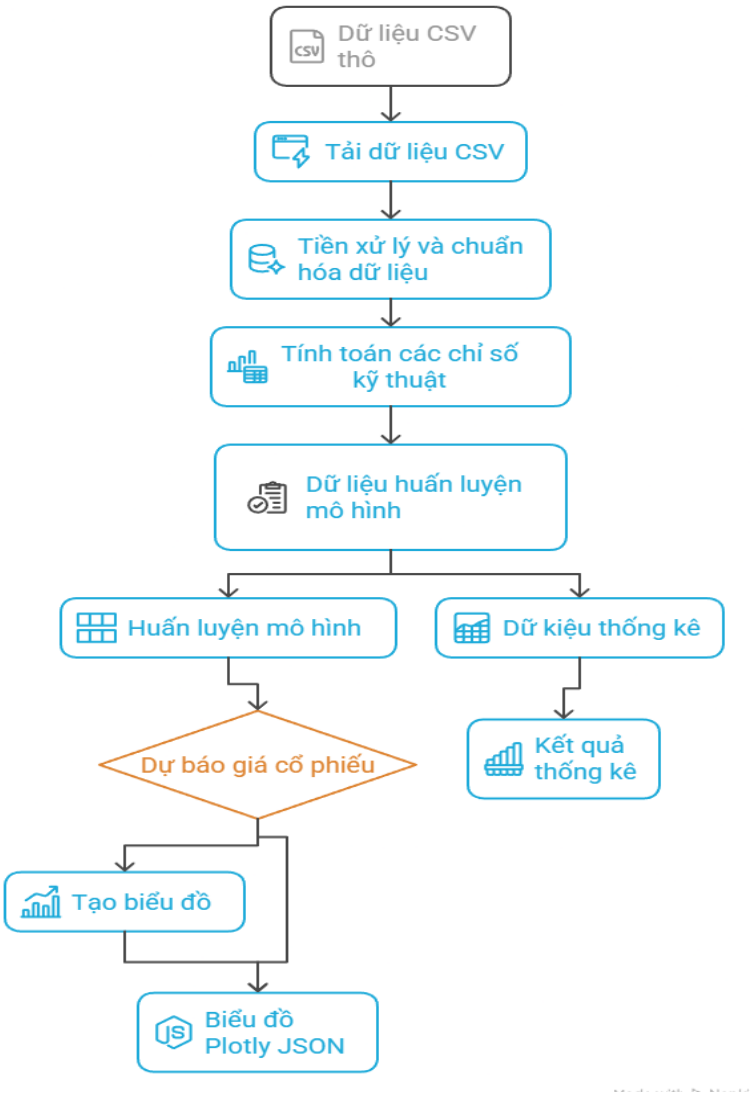
3.1.2. Biểu đồ phân cấp chức năng



Hình 3.1.1. Biểu đồ phân cấp chức năng

3.2. Sơ đồ khối các thuật toán chính

Dưới đây là mô tả chi tiết các khối thuật toán chính trong lớp *StockPredictor*, thể hiện quan hệ đầu vào/đầu ra và chức năng của từng khối.



Hình 3.2.1. Sơ đồ khối thuật toán chính

Mô tả các khối thuật toán chính:

1. **Tải dữ liệu CSV:**
   - **Chức năng:** Tải dữ liệu từ tệp CSV được chỉ định bởi file_path vào một DataFrame của Pandas.
   - **Đầu vào:** file_path (đường dẫn tới tệp CSV).
   - **Đầu ra:** DataFrame Pandas chứa dữ liệu thô.
   - **Thách thức:** Xử lý lỗi đọc file, kiểm tra định dạng file.
2. **Tiền xử lý và chuẩn hóa dữ liệu:**
   - **Chức năng:** Tiền xử lý và chuẩn hóa dữ liệu DataFrame. Bao gồm đổi tên cột (Date -> date, Close -> close, v.v.), chuyển đổi kiểu dữ liệu

(date sang datetime), xử lý giá trị thiếu (NaN) bằng cách điền xuôi/ngược (ffill, bfill), và kiểm tra số lượng dòng dữ liệu tối thiểu.

- **Đầu vào:** DataFrame Pandas thô.
- **Đầu ra:** DataFrame Pandas đã tiền xử lý.
- **Thách thức:** Đảm bảo dữ liệu liên tục, xử lý các trường hợp dữ liệu không đủ.

3. Tính toán các chỉ số kỹ thuật

- **Chức năng:** Tính toán các chỉ số kỹ thuật từ dữ liệu self.data (đã tiền xử lý). Các chỉ số này bao gồm SMA, EMA, RSI, Bollinger Bands, MACD.
- **Đầu vào:** self.data (DataFrame đã tiền xử lý).
- **Đầu ra:** Cập nhật self.data với các cột chỉ số kỹ thuật mới.
- **Thách thức:** Đảm bảo tính toán chính xác theo công thức, xử lý các trường hợp cần dữ liệu đủ dài (ví dụ: RSI 14 ngày, SMA 20/50 ngày).

4. Dữ liệu huấn luyện mô hình

- **Chức năng:** Chuẩn bị dữ liệu cho quá trình huấn luyện mô hình.
    - Tạo các đặc trưng (features) từ dữ liệu giá lịch sử (ví dụ: giá đóng cửa của 30 ngày trước đó) và các chỉ số kỹ thuật.
    - Xác định biến mục tiêu (target) là giá đóng cửa của ngày hiện tại.
    - Loại bỏ các hàng có giá trị NaN do việc tạo đặc trưng.
    - Chia dữ liệu thành tập huấn luyện (80%) và tập kiểm tra (20%).
    - Chuẩn hóa dữ liệu features bằng MinMaxScaler.
- **Đầu vào:** self.data (DataFrame với các chỉ số kỹ thuật).
- **Đầu ra:** X_train, X_test, y_train, y_test (dữ liệu đã chuẩn hóa và chia tập), và self.scaler đã được fit.
- **Thách thức:** Xác định số lượng ngày look_back_days phù hợp, đảm bảo không có dữ liệu rò rỉ (data leakage) giữa tập huấn luyện và kiểm tra.

5. Huấn luyện mô hình học máy

- **Chức năng:** Huấn luyện các mô hình học máy đã định nghĩa (LinearRegression, RandomForestRegressor) trên dữ liệu huấn luyện. Sau khi huấn luyện, đánh giá hiệu suất của từng mô hình trên tập kiểm tra bằng MSE, MAE, RMSE.
- **Đầu vào:** X_train, y_train, X_test, y_test từ prepare_for_training.
- **Đầu ra:** self.models (các mô hình đã huấn luyện) và training_results (kết quả đánh giá mô hình).
- **Thách thức:** Đảm bảo mô hình được huấn luyện đúng cách và đánh giá chính xác.

6. Dự báo giá cổ phiếu

- **Chức năng:** Thực hiện dự báo giá cổ phiếu cho days ngày trong tương lai bằng cách sử dụng mô hình được chọn (model_name). Quá trình dự báo diễn ra theo kiểu "rolling prediction" (từng bước), nghĩa là mỗi dự báo cho ngày tiếp theo sẽ dựa trên dữ liệu thực tế và dự báo của các ngày trước đó.
- **Đầu vào:** model_name (tên mô hình), days (số ngày muốn dự báo).
- **Đầu ra:** Một danh sách các từ điển chứa ngày dự báo và giá dự đoán.
- **Thách thức:** Xây dựng logic dự báo tuần tự phức tạp, đảm bảo các đặc trưng cho ngày dự báo được tạo đúng cách và chuẩn hóa trước khi đưa vào mô hình.

7. Tạo biểu đồ

- **Chức năng:** Tạo biểu đồ tương tác bằng thư viện Plotly, hiển thị giá cổ phiếu lịch sử (giá đóng cửa), các đường chỉ số kỹ thuật (SMA) và đường giá dự báo.
- **Đầu vào:** predictions (danh sách các dự báo từ predict_future).
- **Đầu ra:** Một đối tượng biểu đồ Plotly được chuyển đổi sang định dạng JSON.
- **Thách thức:** Thiết kế biểu đồ rõ ràng, dễ đọc, với đầy đủ thông tin và tính năng tương tác.

8. Thống kê dữ liệu

- **Chức năng:** Trích xuất và trả về các số liệu thống kê cơ bản về dữ liệu cổ phiếu đã tải lên, bao gồm số ngày, giá trung bình, giá cao nhất, giá thấp nhất, khối lượng trung bình và phần trăm thay đổi giá.
- **Đầu vào:** self.data (DataFrame đã xử lý).
- **Đầu ra:** Một từ điển chứa các số liệu thống kê.

3.3. Cấu trúc dữ liệu

Dữ liệu chính được sử dụng và lưu trữ trong chương trình là dữ liệu giá cổ phiếu lịch sử, thường có cấu trúc dạng bảng (DataFrame của Pandas).

STT	Trường thông tin	Mô tả	Kiểu dữ liệu	Ví dụ
1	date	Ngày giao dịch của cổ phiếu	datetime	2023-01-01
2	open	Giá mở cửa của cổ phiếu trong ngày	float64	100.50
3	high	Giá cao nhất của cổ phiếu trong ngày	float64	102.75
4	low	Giá thấp nhất của cổ phiếu trong ngày	float64	99.20
5	close	Giá đóng cửa của cổ phiếu trong ngày	float64	101.30
6	volume	Khối lượng giao dịch của cổ phiếu trong ngày	int64	1200000
7	SMA_20	Trung bình động đơn giản 20 ngày	float64	99.80
8	SMA_50	Trung bình động đơn giản 50 ngày	float64	98.55
9	EMA_20	Trung bình động hàm mũ 20 ngày	float64	100.10
10	RSI_14	Chỉ số sức mạnh tương đối 14 ngày	float64	65.23
11	BOLU	Dải Bollinger trên	float64	103.15
12	BOLD	Dải Bollinger dưới	float64	97.45
13	MACD	Đường MACD	float64	0.87

14	Signal_Line	Đường tín hiệu của MACD	float64	0.72
15	hist	Biểu đồ cột của MACD	float64	0.15
16	close_lag_1	Giá đóng cửa của ngày trước đó (lag 1)	float64	100.80
17	close_lag_2	Giá đóng cửa của 2 ngày trước đó (lag 2)	float64	100.20
18	...	...	...	...
19	close_lag_N	Giá đóng cửa của N ngày trước đó	float64	98.90
20	close	Ngày giao dịch của cổ phiếu	datetime	2023-01-01

Bảng 3.3.1. Cấu trúc dữ liệu chính (Pandas DataFrame)

Lưu ý: Các cột SMA_20 đến close_lag_N là các cột được thêm vào trong quá trình tiền xử lý và kỹ thuật đặc trưng. Số lượng cột close_lag_X sẽ phụ thuộc vào giá trị look_back_days được cấu hình.

3.4. Chương trình

Dưới đây là trình bày các hàm chính trong chương trình, tập trung vào cấu trúc và chức năng của chúng, đặc biệt là các hàm liên quan đến lớp *StockPredictor* và các route của Flask.

- *__init__(self):*

Chức năng: Khởi tạo đối tượng StockPredictor, thiết lập các thuộc tính ban đầu như self.data, self.scaler, và self.models (chứa các mô hình Linear Regression và Random Forest Regressor).

- *load_data(self, file_path):*

Chức năng: Tải và tiền xử lý dữ liệu từ tệp CSV. Đổi tên cột, chuyển đổi định dạng ngày, xử lý giá trị NaN.

- *calculate_features(self):*

Chức năng: Tính toán và thêm các cột chỉ số kỹ thuật (SMA, EMA, RSI, Bollinger Bands, MACD) vào DataFrame self.data.

- *prepare_for_training(self):*

Chức năng: Tạo tập dữ liệu đầu vào (features) và đầu ra (target) cho mô hình. Chia dữ liệu thành tập huấn luyện và kiểm tra, và chuẩn hóa các đặc trưng.

- *train_models(self):*

Chức năng: Huấn luyện các mô hình Linear Regression và Random Forest Regressor trên tập huấn luyện. Đánh giá và lưu trữ kết quả MSE, MAE, RMSE trên tập kiểm tra.

- *predict_future(self, model_name, days):*

Chức năng: Dự báo giá cổ phiếu cho days ngày tới bằng mô hình được chọn. Thực hiện dự báo lặp (rolling prediction).

- *create_chart(self, predictions):*

Chức năng: Tạo đối tượng biểu đồ Plotly tương tác bao gồm giá lịch sử, các chỉ số và đường dự báo, sau đó chuyển đổi thành JSON.

- *get_statistics(self):*

Chức năng: Trích xuất và trả về các số liệu thống kê tổng quan của dữ liệu cổ phiếu.

Các hàm Route của Flask trong chương trình chính:

- *@app.route('/')*

Chức năng: Định tuyến cho trang chủ của ứng dụng. Render tệp index.html để hiển thị giao diện người dùng ban đầu.

- *@app.route('/upload', methods=['POST'])*

Chức năng: Xử lý yêu cầu tải lên tệp CSV từ frontend.

  - Input: Tệp CSV qua request.files.
  - Output: JSON phản hồi (success: True hoặc error) cùng với các kết quả huấn luyện mô hình và thông tin biểu đồ ban đầu nếu thành công.
  - Xử lý:
    - Kiểm tra sự tồn tại của tệp và định dạng (.csv).
    - Lưu tệp tạm thời vào thư mục UPLOAD_FOLDER.
    - Gọi *predictor.load_data(), predictor.calculate_features(), predictor.prepare_for_training(),và predictor.train_models().*
    - Tạo biểu đồ ban đầu *(predictor.create_chart()).*
    - Trả về kết quả dưới dạng JSON.

- *@app.route('/predict', methods=['POST'])*

Chức năng: Xử lý yêu cầu dự báo giá cổ phiếu trong tương lai.

- Input: JSON chứa model (tên mô hình) và days (số ngày dự báo).
- Output: JSON phản hồi (success: True hoặc error) cùng với danh sách dự báo và biểu đồ cập nhật.
- Xử lý:
  - Lấy model_name và days từ request JSON.
  - Gọi predictor.predict_future() để nhận dự báo.
  - Cập nhật biểu đồ với dữ liệu dự báo bằng predictor.create_chart().
  - Trả về kết quả dưới dạng JSON.

- *@app.route('/stats')*

Chức năng: Cung cấp API để lấy các số liệu thống kê về dữ liệu đang được xử lý.

- Input: Không có.
- Output: JSON chứa các số liệu thống kê từ predictor.get_statistics().
- Xử lý: Đơn giản là gọi predictor.get_statistics() và trả về kết quả.

- *@app.route('/<path:filename>')*
  - Chức năng: Phục vụ các tệp tĩnh (như CSS, JS) trong trường hợp không sử dụng web server chuyên dụng hoặc cho mục đích phát triển.
  - Input: Đường dẫn tệp.
  - Output: Tệp tĩnh được yêu cầu.

CHƯƠNG IV: THỰC NGHIỆM VÀ KẾT LUẬN

4.1. Thực nghiệm

Để đánh giá chất lượng và chức năng của ứng dụng web dự báo giá cổ phiếu, em đã tiến hành các bài kiểm thử trên các tính năng chính của sản phẩm. Một tệp dữ liệu cổ phiếu mẫu (ví dụ: AAPL.csv hoặc một tệp CSV tương tự có đủ dữ liệu lịch sử) sẽ được sử dụng để thực hiện các bài kiểm thử.

4.1.1. Môi trường thử nghiệm

Cấu hình hệ thống:

- Hệ điều hành: Windows/macOS/Linux
- Backend: Python 3.9+, Flask
- Frontend: HTML, CSS (Tailwind CSS), JavaScript (Plotly.js)
- Các thư viện: pandas, numpy, scikit-learn, plotly
- Trình duyệt: Chrome/Firefox/Safari (hỗ trợ HTML5)

Dữ liệu thử nghiệm:

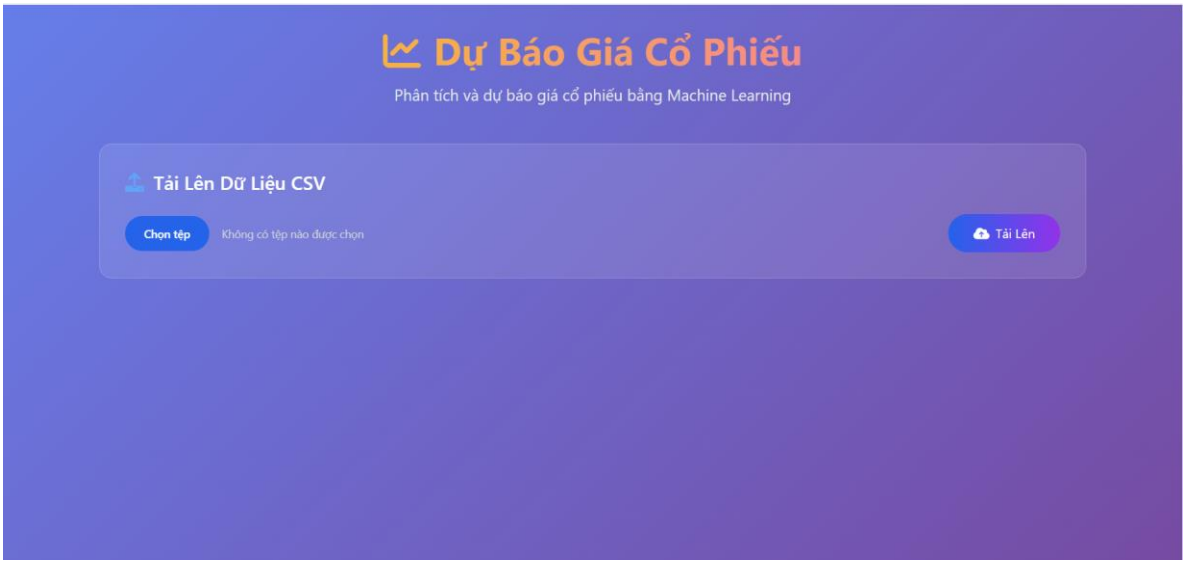
- File CSV chứa dữ liệu lịch sử giá cổ phiếu
- Các cột bắt buộc: Date, Open, High, Low, Close, Volume
- Kích thước dữ liệu: tối thiểu 100 dòng để đảm bảo độ chính xác

4.1.2. Bài kiểm thử tính năng tải lên và xử lý dữ liệu

- **Mục tiêu:** Kiểm tra khả năng tải lên tệp CSV, xử lý dữ liệu, tính toán các chỉ số kỹ thuật và hiển thị thống kê cơ bản.
- **Các bước thực hiện:**
  1. Truy cập vào địa chỉ web
  2. Nhấp vào nút "Chọn tệp" và chọn một tệp CSV chứa dữ liệu giá cổ phiếu hợp lệ.
  3. Nhấp vào nút "Tải lên và Phân tích".
- **Kết quả mong đợi:**
  - Ứng dụng hiển thị thông báo "Tải lên thành công!".

- Phần "Thống kê dữ liệu" được cập nhật với các thông tin tổng quan về dữ liệu như tổng số ngày, giá trung bình, cao nhất, thấp nhất, khối lượng trung bình và phần trăm thay đổi giá.
- Biểu đồ giá cổ phiếu lịch sử (Historical Price Chart) xuất hiện, hiển thị giá đóng cửa và các đường SMA (20, 50), EMA (20).
- Kết quả đánh giá mô hình (MSE, MAE, RMSE) cho cả Linear Regression và Random Forest được hiển thị.

- **Ảnh minh họa:**



Hình 4.1.1: Giao diện trước khi tải lên dữ liệu

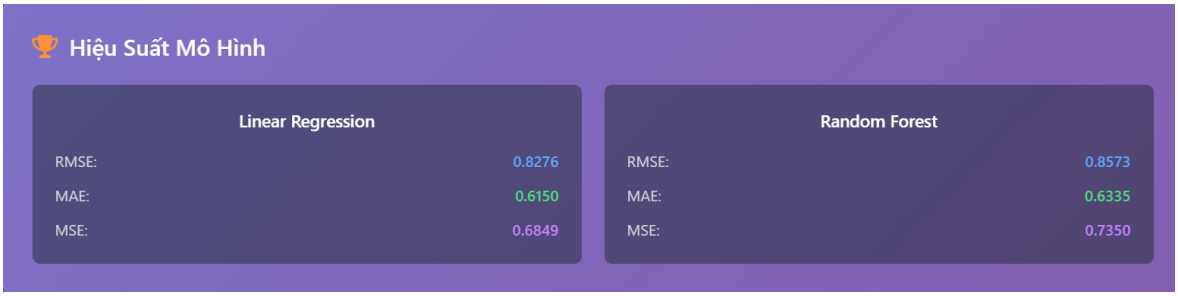
Hình 4.1.2: Màn hình sau khi tải lên dữ liệu thành công

4.1.3. Bài kiểm thử tính năng dự báo giá cổ phiếu

- **Mục tiêu:** Kiểm tra khả năng của mô hình để dự báo giá cổ phiếu trong tương lai và hiển thị kết quả trên biểu đồ.
- **Các bước thực hiện:**
  1. Thực hiện thành công bước 4.1.2.
  2. Trong phần "Dự báo giá cổ phiếu", chọn "Random Forest" (hoặc Linear Regression) từ dropdown "Chọn mô hình".
  3. Nhập số ngày muốn dự báo (ví dụ: 30 ngày) vào ô "Số ngày muốn dự báo".
  4. Nhấp vào nút "Dự báo".
- **Kết quả mong đợi:**
  - Biểu đồ lịch sử được cập nhật, thêm một đường màu khác biểu diễn giá dự báo cho các ngày tiếp theo.
  - Phần "Giá dự báo chi tiết" (Predictions for next days) hiển thị danh sách các ngày và giá dự báo tương ứng.
  - Thông báo "Dự báo thành công!" xuất hiện.
- **Đánh giá chất lượng:**
  - **Độ chính xác của dự báo:** Dựa trên các chỉ số MSE, MAE, RMSE được hiển thị sau khi tải dữ liệu, mô hình Random Forest thường cho kết quả tốt hơn Linear Regression đối với dữ liệu chuỗi thời gian do khả năng nắm bắt các mối quan hệ phi tuyến tính.
  - **Tính ổn định của dự báo:** Quan sát đường dự báo trên biểu đồ. Một đường dự báo hợp lý sẽ có xu hướng nhất quán với dữ liệu lịch sử gần nhất và không có những biến động quá bất thường.
- **Ảnh minh họa:**

Hình 4.1.3: Màn hình hiển thị kết quả dự báo 30 ngày tới với đường dự báo trên biểu đồ

Hình 4.1.4: Danh sách giá dự báo chi tiết theo từng ngày



Hình 4.1.5: Hiệu suất mô hình

4.2. Kết luận

4.2.1. Kết quả đạt được

Ứng dụng web dự báo giá cổ phiếu đã hoàn thành các mục tiêu đề ra và chứng minh khả năng hoạt động tốt với các tính năng cốt lõi:

- **Xử lý dữ liệu mạnh mẽ:** Có khả năng đọc, tiền xử lý và chuẩn hóa dữ liệu từ tệp CSV với các cơ chế xử lý lỗi cơ bản (thiếu cột, NaN, số lượng dòng).
- **Tích hợp chỉ số kỹ thuật:** Tự động tính toán và đưa các chỉ số kỹ thuật phổ biến vào làm đặc trưng cho mô hình, nâng cao khả năng phân tích.
- **Hỗ trợ đa mô hình Machine Learning:** Triển khai thành công hai mô hình Linear Regression và Random Forest Regressor để dự báo, cung cấp sự lựa chọn cho người dùng.
- **Trực quan hóa dữ liệu hiệu quả:** Sử dụng Plotly để tạo biểu đồ tương tác, hiển thị rõ ràng giá lịch sử, chỉ số kỹ thuật và đường dự báo, giúp người dùng dễ dàng theo dõi xu hướng.
- **Giao diện thân thiện:** Giao diện người dùng được thiết kế đơn giản, trực quan, dễ sử dụng cho mọi đối tượng.
- **Dự báo linh hoạt:** Cho phép người dùng lựa chọn số ngày dự báo và mô hình sử dụng, tăng tính tùy biến.

4.2.2. Rút ra kinh nghiệm

Trong quá trình thực hiện đề tài, em đã tích lũy được nhiều kiến thức và kinh nghiệm quý báu:

- **Kiến thức chuyên sâu về Data Science:** Nắm vững hơn các bước trong quy trình Data Science từ thu thập, tiền xử lý dữ liệu, kỹ thuật đặc trưng, huấn luyện mô hình, đánh giá và trực quan hóa.
- **Kỹ năng xử lý chuỗi thời gian:** Hiểu rõ hơn về các đặc thù của dữ liệu chuỗi thời gian và cách áp dụng các mô hình hồi quy để dự báo. Đặc biệt là cách xây dựng các đặc trưng (lagged features) và thực hiện dự báo cuốn chiếu (rolling prediction).
- **Sử dụng thư viện và công cụ:** Thành thạo hơn trong việc sử dụng Pandas, Numpy cho xử lý dữ liệu; Scikit-learn cho học máy; Plotly cho trực quan hóa; và Flask cho phát triển ứng dụng web.

- **Phát triển ứng dụng Full-stack cơ bản:** Có được cái nhìn tổng thể về việc kết nối frontend (HTML/CSS/JS) với backend (Flask Python) thông qua API RESTful (AJAX).
- **Xử lý lỗi và tối ưu hiệu suất:** Học cách xử lý các trường hợp ngoại lệ, kiểm tra dữ liệu đầu vào và tối ưu hóa một số phần của mã (ví dụ: *n_jobs=-1* cho Random Forest).

4.2.3. Hướng phát triển

Để nâng cao chất lượng và khả năng ứng dụng của sản phẩm, chúng tôi đề xuất một số cải tiến trong tương lai:

- **Mở rộng mô hình dự báo:**
  - Tích hợp các mô hình học sâu (Deep Learning) như LSTM (Long Short-Term Memory) hoặc GRU (Gated Recurrent Unit), vốn rất mạnh mẽ trong việc xử lý chuỗi thời gian phức tạp và có thể mang lại độ chính xác cao hơn.
  - Xem xét các mô hình chuyên biệt cho chuỗi thời gian như Prophet của Facebook.
- **Thêm các chỉ số kỹ thuật và phân tích nâng cao:**
  - Bổ sung thêm các chỉ báo kỹ thuật khác như Volume Weighted Average Price (VWAP), Average True Range (ATR), Ichimoku Cloud, v.v.
  - Tích hợp phân tích cảm xúc (Sentiment Analysis) từ tin tức tài chính hoặc mạng xã hội để đưa ra cái nhìn toàn diện hơn về thị trường.
- **Dữ liệu đa dạng và tự động:**
  - Cho phép người dùng lựa chọn mã cổ phiếu trực tiếp từ các API dữ liệu tài chính (ví dụ: Yahoo Finance API, Alpha Vantage API) thay vì phải tải lên tệp CSV thủ công.
  - Tự động cập nhật dữ liệu hàng ngày để dự báo được liên tục và chính xác hơn.
- **Cải thiện giao diện và trải nghiệm người dùng:**

- Thêm chức năng lưu/tải dữ liệu và cài đặt của người dùng.
- Cung cấp các tùy chỉnh biểu đồ nâng cao hơn.
- Phản hồi trực quan tốt hơn cho quá trình xử lý và dự báo.
- Thêm tính năng hiển thị nến Nhật (Candlestick chart) bên cạnh biểu đồ đường.

- **Đánh giá mô hình chi tiết hơn:**
  - Tích hợp các kỹ thuật đánh giá chuỗi thời gian như Time Series Cross-Validation.
  - Hiển thị biểu đồ so sánh giữa giá thực tế và giá dự báo trên tập kiểm tra để người dùng dễ hình dung hiệu suất mô hình.
- **Triển khai (Deployment):**
  - Đưa ứng dụng lên các nền tảng đám mây (ví dụ: Heroku, AWS, Google Cloud) để có thể truy cập công khai và sử dụng rộng rãi.

TÀI LIỆU THAM KHẢO

[1] VanderPlas, J. (2016). *Python Data Science Handbook: Essential Tools for Working with Data*. O'Reilly Media. (Các chương về Pandas, NumPy, Matplotlib/Plotly và Scikit-Learn)

[2] Murphy, J. J. (1999). *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications*. New York Institute of Finance.

[3] Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems* (2nd ed.). O'Reilly Media. (Các chương về Regressions và Ensemble Learning).

[4] Kaggle: https://www.kaggle.com/ (Nền tảng với nhiều bộ dữ liệu, notebooks và cuộc thi về dự báo giá cổ phiếu)