TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN** 2

**Tìm hiểu các phương pháp dự đoán chỉ số VN30 cho thị trường chứng khoán phái sinh Việt Nam**

*Người hướng dẫn*: **Thầy NGUYỄN CHÍ THIỆN**

*Người thực hiện*: **NGUYỄN DUY HÀN LÂM– 51403229**

**PHẠM TRẦN THỦY TIÊN – 51403342**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2017**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN** 2

**Tìm hiểu các phương pháp dự đoán chỉ số VN30 cho thị trường chứng khoán phái sinh Việt Nam**

*Người hướng dẫn*: **Thầy NGUYỄN CHÍ THIỆN**

*Người thực hiện*: **NGUYỄN DUY HÀN LÂM– 51403229**

**PHẠM TRẦN THỦY TIÊN – 51403342**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2017**

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng tôi và được sự hướng dẫn của TS Nguyễn Chí Thiện. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày tháng năm*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Nguyễn Duy Hàn Lâm*

*Phạm Trần Thủy Tiên*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TÓM TẮT

Trình bày tóm tắt vấn đề nghiên cứu, các hướng tiếp cận, cách giải quyết vấn đề và một số kết quả đạt được, những phát hiện cơ bản trong vòng 1 -2 trang.

MỤC LỤC

[LỜI CẢM ƠN ii](#_Toc387692905)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN ii](#_Toc387692906)

[TÓM TẮT ii](#_Toc387692907)

[MỤC LỤC 2](#_Toc387692908)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ 2](#_Toc387692909)

[CHƯƠNG 1 – MỞ ĐẦU 2](#_Toc387692910)

[1.1 Tiểu mục cấp 1 2](#_Toc387692911)

[1.1.1 Tiểu mục cấp 2 2](#_Toc387692912)

[1.1.1.1 Tiểu mục cấp 3 2](#_Toc387692913)

[1.1.1.2 Tiểu mục cấp 3 tiếp theo. 2](#_Toc387692914)

[1.1.2 Tiểu mục cấp 2 tiếp theo 2](#_Toc387692915)

[1.2 Nội dung của chương này 2](#_Toc387692916)

[CHƯƠNG 2 – TỔNG QUAN 2](#_Toc387692917)

[1.1 Trình bày công thức toán học 2](#_Toc387692918)

[1.2 Trình bày một hình vẽ, sơ đồ 2](#_Toc387692919)

[CHƯƠNG 3 – CƠ SỞ LÝ THUYẾT / NGHIÊN CỨU THỰC NGHIỆM 2](#_Toc387692920)

[3.1 Chèn bảng: 2](#_Toc387692921)

[3.2 Viết tắt 2](#_Toc387692922)

[3.3 Trích dẫn 2](#_Toc387692923)

[3.3.1 Tài liệu tham khảo và cách trích dẫn 2](#_Toc387692924)

[3.3.2 Qui định của Khoa Công nghệ thông tin 2](#_Toc387692925)

**DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT**

**CÁC KÝ HIỆU**

*f Tần số của dòng điện và điện áp (Hz)*

*p Mật độ điện tích khối (C/m3)*

**CÁC CHỮ VIẾT TẮT**

CSTD Công suất tác dụng

MF Máy phát điện

BER Tỷ lệ bít lỗi

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

**DANH MỤC HÌNH**

[Hình 2.1: Kiến trúc FTP 2](#_Toc387689394)

**DANH MỤC BẢNG**

[Bảng 3.1 Ví dụ cho chèn bảng 2](#_Toc387689363)

CHƯƠNG 1 – TỔNG QUAN VỀ HỌC MÁY

* 1. Định nghĩa

Là một lĩnh vực nhỏ của khoa học máy tính, giúp máy tính có khả năng tự học hỏi dựa trên dữ liệu đưa vào mà không cần lập trình cụ thể, tạo ra các chương trình máy tính bằng việc phân tích các tập dữ liệu.[1]

Học máy có liên quan lớn đến thống kê, tập trung vào sự phức tạp của các giải thuật trong việc thực thi tính toán. Một phần của học máy là nghiên cứu sự phát triển các giải thuật suy luận xấp xỉ có thể xử lí được.

Có hai phương pháp học máy chính:

* Phương pháp quy nạp: phân biệt các khái niệm dựa trên dữ liệu đã thu thập được trước đó, cho phép tận dụng được nguồn dữ liệu rất nhiều và sẵn có.
* Phương pháp suy diễn: phân biệt các khái niệm dựa vào các luật, cho phép tận dụng được các kiến thức chuyên ngành để hỗ trợ máy tính.
  1. Các ngành khoa học liên quan

Thống kê: cho phép ước lượng sai số các phương pháp học máy. Các kết quả trong xác suất thống kê là tiền đề cho rất nhiều phương pháp học máy.

Toán ứng dụng: hầu hết các thuật toán học máy thường sử dụng các tính toán số thực hoặc số nguyên trên dữ liệu rất lớn. Điển hình là các bài toán tối ưu có ràng buộc hoặc không có ràng buộc, giải phương trình tuyến tính,.. được sử dụng rất phổ biến.

Khoa học máy tính: là cơ sở để thiết kế các thuật toán, đồng thời đánh giá thời gian chạy, bộ nhớ của các thuật toán học máy.

* 1. Các phương pháp dùng trong học máy

Mô hình hóa các hàm mật độ xác suất điều kiện:

* Mạng nơ-ron
* Cây quyết định
* Lập trình biểu thức gen
* Lập trình di truyền
* Hồi quy quá trình Gauss
* Phân tích biệt thức tuyến tính
* K láng giềng gần nhất
* Độ dài thông điệp tối thiểu
* Cảm tri nguyên
* Hàm cơ sở xuyên tâm
* Máy hỗ trợ vector

Mô hình hóa các hàm mật độ xác suất qua các mô hình phát sinh:

* Thuật toán cực đại kì vọng
* Các mô hình đồ họa gồm mạng Bayes và mạng Markov
* Ánh xạ topo phát sinh

Các kỹ thuật suy luận xấp xỉ đúng:

* Chuỗi Markov phương pháp Monte Carlo
* Phương pháp biến thiên

Hầu hết các phương pháp trên đều sử dụng tối ưu hóa hoặc thể hiện của các thuật toán tối ưu hóa.

* 1. Ứng dụng

Học máy được ứng dụng rộng rãi trong các ngành khoa học, sản xuất. Đặc biệt những ngành cần phân tích khối lượng dữ liệu khổng lồ. Một số ứng dụng thường thấy:

* Xử lý ngôn ngữ tự nhiên (Natural Language Processing): xử lý văn bản, giao tiếp người-máy.
* Nhận dạng (Pattern Recognition): khuôn mặt, giọng nói, chữ viết tay, vân tay, thị giác máy.
* Gợi ý bạn bè từ facebook, phân loại spam email từ google mail, dự đoán kết quả trận đấu.
* Chuẩn đoán y khoa: phân tích ảnh X-quang, các hệ chuyên gia chẩn đoán tự động, phân loại các chuẩn DNA.
* Tin sinh học: phân loại chuỗi gen, quá trình hình thành gen/protein.
* Vật lý: phân tích ảnh thiên văn, tác động giữa các hạt,….
* Phát hiện gian lận tài chính (Financial fraud): gian lận thẻ tín dụng.
* Phân tích thị trường chứng khoán (Stock market analysis).
* Chơi trò chơi: tự động chơi cờ, hành động của các nhân vật ảo.
* Robot: tạo nên hệ thần kinh, bộ não của người máy.
  1. Các loại giải thuật

Được chia làm 2 loại chính là Supervised Learning (Học có giám sát) và Unsupervised Learning (Học không giám sát). Ngoài ra còn một vài loại khác như SemiSupervised Learning, Reinforcement Learning, Learning to Learn, Developmental Learning,…

* + 1. Supervised Learning

Là kĩ thuật xây dựng hàm từ dữ liệu có sẵn và output chính xác, đồng thời xác định được mối quan hệ giữa input và output. Cụ thể cho sẵn tập giá trị x và y tương ứng mỗi x, khi đó mối quan hệ x và y được xác định y = f(x). Xác định được hàm f tốt nhất để dự đoán được y khi có x.

Chia thành hai nhóm:

* Regression Problem(Bài toán hồi qui): giá trị của output là giá trị thực và liên tục. Ví dụ: giá cả, cân nặng, chiều cao,…các giải thuật được dùng gồm Linear Regression, Logistic Regression, Random Forest.
* Classification Problem(Bài toán phân loại): giá trị output rời rạc, không có mối quan hệ với nhau. Ví dụ: màu sắc, khối u lành tính hay ác tính,….các giải thuật được dùng gồm Random Forest, Support Vector.
  + 1. Unsupervised Learning

Không tìm một correct output như Supervised Learning mà tìm các structure, relationship ẩn sâu trong data set. Output phụ thuộc vào input ban đầu.

Kết hợp với suy diễn Bayes cho ra xác suất có điều kiện cho bất kì biến ngẫu nhiên khi biết trước các biến khác.

Hữu ích cho việc nén dữ liệu và gom nhóm dữ liệu.

Ứng dụng trong nhiều lĩnh vực đời sống. Phổ biến nhất là gợi ý kết bạn của facebook, đánh giá các nhóm khách hàng tiềm năng cho công ty trong lĩnh vực tài chính ngân hàng.

Chia làm hai nhóm:

* Clustering: hướng đến việc phân nhóm, phân đoạn dữ liệu từ tập dữ liệu ban đầu. Ví dụ: muốn phân nhóm tự động một tập gồm 1 triệu Gen cần dựa vào vòng đời, vị trí, vai trò,…dùng k-means.
* Non-clustering: tìm các structure ẩn trong dữ liệu. Ví dụ: nhận dạng giọng nói trong môi trường tạp âm dùng Cocktail Party Algorithm.
  + 1. Half Supervised Learning

Kĩ thuật sử dụng một lượng nhỏ dữ liệu có gán nhãn cùng với lượng lớn dữ liệu chưa gán nhãn nhằm cải thiện đáng kể độ chính xác.

Dùng cho các dữ liệu đồng huấn luyện, trong đó một hay nhiều bộ học được huấn luyện cùng tập ví dụ nhưng mỗi bộ dùng tập đặc trưng khác nhau, độc lập nhau.

Dùng trong mô hình hóa phân phối xác suất đồng thời của các đặc trưng và nhãn, coi dữ liệu chưa gán nhãn là dữ liệu còn thiếu. Các kĩ thuật xử lý dữ liệu còn thiếu như lấy mẫu Gibbs và tối ưu kỳ vọng để ước lượng tham số.

CHƯƠNG 2 – CHỈ SỐ VN30 VÀ PHƯƠNG PHÁP DỰ ĐOÁN CƠ BẢN

Chương trước chúng ta khái quát một số nội dung về học máy. Ở chương này tài liệu sẽ giải thích các khái niệm về chứng khoán, thị trường chứng khoán phái sinh, chỉ số VN30 và cách dự đoán chỉ số VN30 cơ bản.

1. 1. Các khái niệm
      1. Chứng khoán

Chứng khoán là bằng chứng xác nhận quyền và lợi ích hợp pháp của người sở hữu đối với tài sản hoặc phần vốn của tổ chức phát hành.[2]

Chứng khoán được thể hiện bằng hình thức chứng chỉ, bút toán ghi sổ hoặc dữ liệu điện tử.

Chứng khoán bao gồm các loại: [cổ phiếu](https://vi.wikipedia.org/wiki/C%E1%BB%95_phi%E1%BA%BFu), [trái phiếu](https://vi.wikipedia.org/wiki/Tr%C3%A1i_phi%E1%BA%BFu), [chứng chỉ quỹ đầu tư](https://vi.wikipedia.org/w/index.php?title=Ch%E1%BB%A9ng_ch%E1%BB%89_qu%E1%BB%B9_%C4%91%E1%BA%A7u_t%C6%B0&action=edit&redlink=1), [chứng khoán phái sinh](https://vi.wikipedia.org/wiki/Ph%C3%A1i_sinh_t%C3%A0i_ch%C3%ADnh). Thực chất chứng khoán là một loại [hàng hóa](https://vi.wikipedia.org/wiki/H%C3%A0ng_h%C3%B3a) đặc biệt.

Chứng khoán là một phương tiện hàng hóa trừu tượng có thể thỏa thuận và có thể thay thế được, đại diện cho một giá trị tài chính. Chứng khoán gồm các loại: [chứng khoán cổ phần](https://vi.wikipedia.org/w/index.php?title=Ch%E1%BB%A9ng_kho%C3%A1n_c%E1%BB%95_ph%E1%BA%A7n&action=edit&redlink=1) (ví dụ cổ phiếu phổ thông của một công ty), [chứng khoán nợ](https://vi.wikipedia.org/w/index.php?title=Ch%E1%BB%A9ng_kho%C3%A1n_n%E1%BB%A3&action=edit&redlink=1) (như trái phiếu nhà nước, trái phiếu công ty...) và các [chứng khoán phái sinh](https://vi.wikipedia.org/wiki/Ph%C3%A1i_sinh_t%C3%A0i_ch%C3%ADnh) (như các quyền chọn, hợp đồng quy đổi - Swap, hợp đồng tương lai, Hợp đồng kỳ hạn).

* + 1. Chứng khoán phái sinh

Chứng khoán phái sinh là các công cụ tài mà giá trị của chúng phụ thuộc vào một tài sản cơ sở.[3] Chứng khoán phái sinh cũng là một loại của chứng khoán vì vậy nội dung của chứng khoán phái sinh trong đó sẽ quy định quyền lợi, nghĩa vụ của các bên tham gia vào các hình thức của nó (hợp đồng, chuyển giao,…) trong các hoạt động thanh toán, trao đổi,… với một mức giá được thỏa thuận ban đầu vào một thời điểm nhất định trong tương lai.

Tài sản cơ sở của chứng khoán phái sinh có thể là hàng hóa cụ thể như nông sản, kim loại,… hoặc các công cụ tài chính trừu tượng như trái phiếu, cổ phiếu, lãi suất,…



Hình 2.1 Minh họa về chứng khoán phái sinh[3]

* + 1. Thị trường chứng khoán

Thị trường chứng khoán là một bộ phận quan trọng của Thị trường vốn, hoạt động của nó nhằm huy động những nguồn vốn tiết kiệm nhỏ trong xã hội tập trung thành nguồn vốn lớn tài trợ cho doanh nghiệp, các tổ chức kinh tế và Chính phủ để phát triển sản xuất, tăng trưởng kinh tế hay cho các dự án đầu tư.

Thị trường chứng khoán là nơi diễn ra các hoạt động giao dịch mua bán các loại chứng khoán. Việc mua bán được tiến hành ở hai thị trường sơ cấp và thứ cấp. do vậy thị trường chứng khoán là nơi chứng khoán được phát hành và trao đổi.

* [Thị trường sơ cấp](https://vi.wikipedia.org/wiki/Th%E1%BB%8B_tr%C6%B0%E1%BB%9Dng_ch%E1%BB%A9ng_kho%C3%A1n_s%C6%A1_c%E1%BA%A5p): người mua mua được chứng khoán lần đầu từ những người phát hành;
* [Thị trường thứ cấp](https://vi.wikipedia.org/wiki/Th%E1%BB%8B_tr%C6%B0%E1%BB%9Dng_ch%E1%BB%A9ng_kho%C3%A1n_th%E1%BB%A9_c%E1%BA%A5p): nơi diễn ra sự mua đi bán lại các chứng khoán đã được phát hành ở Thị trường sơ cấp.

Căn cứ vào tính chất chứng khoán giao dịch:

* [Thị trường cổ phiếu](https://vi.wikipedia.org/wiki/C%E1%BB%95_phi%E1%BA%BFu);
* [Thị trường trái phiếu](https://vi.wikipedia.org/wiki/Tr%C3%A1i_phi%E1%BA%BFu);
* [Thị trường chứng khoán phái sinh](https://vi.wikipedia.org/wiki/Ch%E1%BB%A9ng_kho%C3%A1n_ph%C3%A1i_sinh), bao gồm thị trường hợp đồng tương lai, hợp đồng lựa chọn,… - Đây là thị trường cấp cao mua bán chuyển giao các công cụ tài chính cấp cao; Do đó thị trường này chỉ xuất hiện ở các nước có thị trường chứng khoán phát triển ở trình độ cao.
  + 1. Chỉ số VN30

Ngày 06/02/2012, Sở Giao Dịch Chứng Khoán TPHCM (HOSE) chính thức áp dụng chỉ số VN30 trên sàn giao dịch chứng khoán, thay thế cho chỉ số VN Index trước đây.

Chỉ số VN30 là chỉ số do HOSE công bố, chỉ số này được tính dựa trên ba tiêu chí: giá trị vốn hoá, tỷ lệ free-float và giá trị giao dịch; bao gồm 30 cổ phiếu của công ty niêm yết trên HOSE có vốn hoá thị trường và tính thanh khoản cao nhất (chiếm khoảng 80% tổng giá trị vốn hóa và 60% tổng giá trị giao dịch toàn thị trường).

VN30 sẽ giúp NĐT đánh giá thị trường chính xác hơn so với VN Index, tránh được hiện tượng VN Index bị bóp méo do các mã vốn hóa lớn: Trong hơn 300 cổ phiếu niêm yết trên HOSE được phân làm 11 ngành chính thì riêng VN30 đã có đại diện của 9 ngành, chiếm khoảng 80% tổng giá trị vốn hóa và 60% thanh khoản của thị trường. Như vậy, tính đại diện của nó rất rõ, vừa đại diện ngành, vốn hóa và thanh khoản, thật sự trở thành công cụ hữu ích cho các quỹ đầu tư chỉ số...

Thành phần chỉ số VN30 sẽ được hội đồng tư vấn chỉ số của HOSE xem xét lại 6 tháng/lần vào tháng 7 trong năm và tháng 1 của năm kế tiếp. Dữ liệu dùng trong kỳ xem xét được lấy sau khi đóng cửa phiên giao dịch cuối cùng của tháng 6 và tháng 12.

**Phương pháp tính:**

VN30 được tính dựa trên ba tiêu chí: giá trị vốn hoá, tỷ lệ free-float và giá trị giao dịch; bao gồm 30 cổ phiếu của công ty niêm yết trên HOSE có vốn hoá thị trường và tính thanh khoản cao nhất (chiếm khoảng 80% tổng giá trị vốn hóa và 60% tổng giá trị giao dịch toàn thị trường)

Chỉ số VN30 có tần suất tính toán 1 phút/lần và sẽ được xem xét định kỳ 6 tháng/lần vào tháng 7 và tháng 1 hằng năm. Tỷ trọng các cấu phần trong chỉ số không quá 10%. Chỉ số được tính toán dựa trên phương pháp giá trị vốn hóa thị trường có điều chỉnh tỷ lệ cổ phiếu tự do chuyển nhượng. Hiện HOSE chưa xác định ngày cơ sở và giá trị chỉ số cơ sở.

**Công thức tính VN30:**

Trong đó, CMV là giá trị vốn hóa thị trường hiện tại (Current Market Value) và BMV là giá trị vốn hóa thị trường cơ sở hay còn gọi là hệ số chia cơ bản (Base Market Value).

BMV được điều chỉnh nhằm loại trừ những thay đổi về khối lượng và giá cổ phiếu ảnh hưởng đến chỉ số.

Trong (1), CMV được tính theo công thức sau:

Với:

* n: số cổ phiếu trong rổ chỉ số.
* pi: giá của cổ phiếu i trong rổ chỉ số tại thời điểm tính toán.
* si: Khối lượng đang lưu hành của cổ phiếu i trong rổ chỉ số tại thời điểm tính toán.
* fi: tỷ lệ free-float của cổ phiếu i trong rổ chỉ số tại thời điểm tính toán.
* ci: giới hạn tỷ trọng vốn hóa của cổ phiếu i trong rổ chỉ số tại thời điểm tính toán.

Trong (2), tỷ lệ free-float (f) được tính theo công thức:

Chú thích: KLCP: khối lượng cổ phiếu.

Trong đó, thay đổi tỷ lệ free-float (tỷ lệ cổ phiếu tự do chuyển nhượng) của các cổ phiếu thành phần sẽ được điều chỉnh 6 tháng/lần cùng với thời điểm xem xét lại chỉ số.

Đối với chỉ số trong ngày, giá để tính toán chỉ số là giá khớp gần nhất của các cổ phiếu trong rổ tính đến thời điểm chỉ số được tính. Trường hợp không có giá khớp lệnh thì sẽ lấy giá đóng cửa gần nhất của cổ phiếu đó hoặc giá đóng cửa điều chỉnh đối với cổ phiếu có xảy ra sự kiện doanh nghiệp.

Đối với chỉ số cuối ngày, giá để tính toán chỉ số là giá đóng cửa trong ngày giao dịch của các cổ phiếu trong rổ. Trường hợp trong ngày giao dịch, cổ phiếu không có khớp lệnh thì sẽ lấy giá đóng cửa gần nhất của cổ phiếu hoặc giá đóng cửa điều chỉnh đối với cổ phiếu có xảy ra sự kiện doanh nghiệp.

Vì tính chất phức tạp của công thức tính chỉ số VN30 nên ta chỉ thường để ý đến kết quả của VN30. Do đó việc dự đoán gần đúng hoặc chỉ số VN30 có thể giúp nhà đầu tư có thể rút ngắn thời gian, công sức, có cái nhìn bao quát tình hình thị trường giúp nhà đầu tư có những lựa chọn sáng suốt.

* 1. Phương pháp dự đoán chỉ số VN30 cơ bản

Trong tài liệu này, chúng ta sẽ làm việc thông qua một loạt chuỗi các dự báo từ đầu đến cuối, quy trình được bắt đầu từ việc nhận tập tài liệu dữ liệu chứa các kết quả chỉ số VN30 được tính theo từng ngày sau đó là quy trình xác định vấn đề để đào tạo ra một mô hình cuối cùng giúp ta có cái nhìn tổng quan về VN30 và cuối cùng từ mô hình đó ta sẽ đưa ra dự đoán.

Tài liệu này có thể sẽ không đầy đủ, nhưng sẽ cung cấp một phương pháp dự đoán cơ bản cho vấn đề dự báo chỉ số VN30 một cách nhanh chóng thông qua phương pháp chuỗi dự đoán một cách có hệ thống.

Các bước của phương pháp bao gồm:

* Kiểm tra môi trường lập trình.
* Mô tả của bài toán.
* Các khai thác kiểm tra.
* Tính độ sai lệch.
* Phân tích dữ liệu.
* Lập các mô hình ARIMA.
* Mô hình chuẩn.

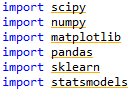
2.1.1 Môi trường lập trình

Các thư viện / môi trường mà tài liệu sử dụng bao gồm:

* SciPy
* NumPy
* Matplotlib
* Pandas
* scikit-learn
* statsmodels

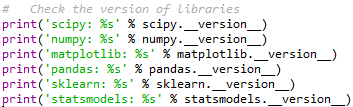
Về vấn đề cài đặt Python và các thư viện trên, ta có thể tải xuống và cài đặt mã nguồn mở của Anacoda.

Đầu tiên, ta sẽ import các thư viện:



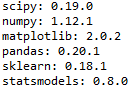
Hình 2.2 Import các thư viện cần dùng

Tập lệnh Python sau sẽ giúp chúng ta có thể kiểm tra phiên bản đã cài đặt của các thư viện này:



Hình 2.3 Kiểm tra phiên bản của các thư viện đã import

Chương trình sẽ trả ra phiên bản của các thư viện tương ứng:



Hình 2.4 Kết quả phiên bản của các thư viện

2.1.2 Mô tả của bài toán

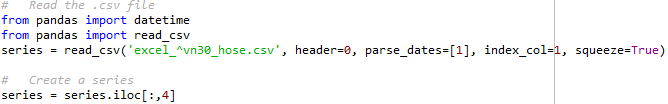
Bài toán ở đây đó là dự đoán chỉ số VN30 hàng ngày của thị trường phái sinh Việt Nam.

Tập dữ liệu cung cấp là tập dữ liệu các chỉ số VN30 được tính theo ngày từ ngày 06/02/2012 cho đến ngày 09/03/2012 (sẽ có một số ngày không cung cấp giá trị VN30).

Tập dữ liệu có tổng cộng là 25 quan sát (25 bản ghi/dòng). Mỗi dòng là dữ liệu chỉ số VN30 của một ngày duy nhất.

Tập dữ liệu được đính kèm cùng với tài liệu này có tên file là “excel\_^vn30\_hose.csv”.

Ta sẽ đọc file .csv này vào chương trình bằng đoạn python sau:



Hình 2.5 Đoạn code python đọc file .csv

Thực hiện đoạn code trên là ta đã nạp xong dữ liệu vào chương trình. Ta có thể bắt đầu tiến hành khảo sát dữ liệu.

2.1.3 Cách khai thác kiểm tra

Việc đầu tiên từ tập dữ liệu ban đầu ta sẽ tạo thành bộ dữ liệu kiểm thử để điều tra về dữ liệu và đánh giá các mô hình.

Giai đoạn này bao gồm hai bước:

* Định nghĩa tập dữ liệu Validation dataset.
* Xây dựng một tập dữ liệu để làm phương pháp để xây dựng các mô hình.

2.1.3.1 Tập Validation dataset

Các dữ liệu ban đầu không phải là các dữ liệu của thời điểm hiện tại. Điều này có nghĩa là chúng ta không có dữ liệu hiện tại để xác thực (xây dựng) các mô hình.

Vì vậy, chúng ta sẽ phải giả vờ rằng một thời gian nào đó trong tập dữ liệu là thời gian hiện tại và phần còn lại của dữ liệu sẽ được dùng để phân tích và lựa chọn mô hình.

Tập dữ liệu từ thời điểm ta giả định trở về trước sẽ được sử dụng để xây dựng mô hình cuối cùng ta gọi là dataset. Còn validation test (phần còn lại) sẽ được dùng để đánh giá mô hình mà ta xây dựng xong.

Tập dữ liệu validation dataset sẽ chiếm khoảng 40% số liệu ban đầu của tập dữ liệu gốc. Nên ta sẽ có dòng code sau:



Hình 2.6 Tính số lượng dữ liệu của tập validation dataset sẽ có

Qua dòng code trên ta sẽ tính được validation dataset sẽ có 10 dòng dữ liệu và phần còn lại sẽ được lưu vô tập dữ liệu với tên khác có số lượng là 15 dòng dữ liệu.

Lưu ý rằng các tập dữ liệu đã lưu không có dòng tiêu đề, do đó chúng ta không cần phải phục vụ cho việc này khi làm việc với các tệp này sau đó.

Đoạn code python dưới đây sẽ chia tập dữ liệu thành hai phần, một phần sẽ dùng cho việc phát triển mô hình (dataset.csv) và một phần sẽ dùng cho việc xác nhận tính hợp lệ của mô hình (validation.csv):



Hình 2.7 Chia tập dữ liệu gốc thành 2 phần

Trong đó tập validation dataset sẽ có tên là validation và phần còn lại để đánh giá các mô hình là dataset.

Như vậy file dataset.csv sẽ chứa dữ liệu từ ngày 02/06/2012 cho đến ngày 24/02/2012. Và file validation.csv sẽ chứa dữ liệu từ ngày 27/02/2012 cho đến ngày 09/03/2012.

2.1.3.2 Đánh giá các mô hình

Đánh giá mô hình sẽ chỉ được thực hiện trên tập dữ liệu (dataset) được lưu trong file dataset.csv được tạo ra trong phần trước.

Đánh giá mô hình bao gồm hai yếu tố:

* Đo lường hiệu suất.
* Chiến lược kiểm tra.

**Đo lường hiệu suất:**

Dữ liệu quan sát là chỉ số VN30 theo từng ngày.

Chúng ta sẽ đánh giá hiệu suất (độ sai số) của dự đoán bằng cách sử dụng căn bậc hai của Sai số toàn phương trung bình (RMSE – Root Mean Squared Error). Điều này sẽ đánh giá khá chính xác về độ sai số của các dự đoán và sẽ giá trị này sẽ cùng kiểu với dữ liệu ban đầu (giá trị RMSE sẽ là giá trị số thực 32 bit).

Mọi thay đổi về dữ liệu cần phải được thực hiện trước khi RMSE được tính toán và báo cáo để ta có thể tính được RMSE một cách chính xác.

Chúng ta có thể tính toán RMSE bằng cách sử dụng hàm từ thư viện mean\_squared\_error() để tìm ra giá trị sai số toàn phương trung bình (MSE) giữa một tập danh sách các giá trị mong đợi (tập kiểm tra) và tập danh sách các dự đoán (cách làm sẽ được nói ở mục 2.1.4). Sau đó ta sẽ lấy căn bậc hai của giá trị này để ta thu được giá trị RMSE.

Về cơ bản, ta có công thức của RMSE như sau:

Trong đó MSE được tính bởi công thức:

Với:

* là một tập vector các giá trị quan sát được
* là một tập vector các giá trị dự báo

**Chiến lược kiểm tra:**

Các mô hình sẽ được xây dựng bằng phương pháp walk-forward validation.

Lý do ta sử dụng walk-forward validation có nguyên nhân xuất phát từ định nghĩa của các phương pháp chuỗi thời gian bên kinh tế lượng. Cứ mỗi bước của dự báo đều cần phải dựa vào tất cả các dữ liệu được cung cấp (cụ thể trong ví dụ này là dữ liệu trong file “dataset.csv”).

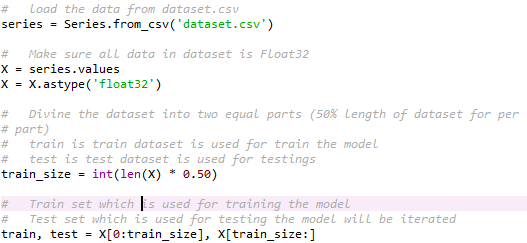
Việc walk-forward validation sẽ hoạt động như sau:

* 50% dữ liệu đầu tiên sẽ được giữ lại để đào tạo ra mô hình.
* 50% dữ liệu còn lại sẽ được lặp lại và thử nghiệm mô hình.
* Đối với mỗi bước trong tập dữ liệu kiểm tra/thử nghiệm (“dataset.csv”) sẽ có các công việc sau:
* Đào tạo mô hình ở mỗi bước.
* Thực hiện dự đoán và dự đoán này được lưu trữ để dùng cho việc đánh giá sau này.
* Các quan sát thực tế từ tập dữ liệu thử nghiệm sẽ được thêm vào tập dữ liệu huấn luyện cho lần lặp tiếp theo.
* Các dự đoán sẽ được thực hiện trong quá trình lặp của bộ dữ liệu đồng thời các dự đoán này sẽ được đánh giá độ sai số/sai lệch bằng giá trị RMSE.

Với kích thước nhỏ của dữ liệu, việc đào tào lại mô hình với tất cả các dữ liệu trước mỗi dự đoán (trong mỗi vòng lặp) là điều có thể cho phép.

Chúng ta có thể làm điều này bằng các câu lệnh Python và các câu lệnh sử dụng thư viện NumPy viết trên Python.

Điều đầu tiên chúng ta cần làm đó là chia tập dữ liệu (lưu trong file “dataset.csv”) thành hai phần bằng nhau, trong đó một phần sẽ được dùng để đào tạo mô hình và phần còn lại sẽ được dùng để kiểm tra trực tiếp. Trong một vài trường hợp, dữ liệu trong tập dữ liệu có thể khiến chương trình không thể xử lý được (như là dữ liệu dạng chuỗi, dạng số nguyên,…). Vì vậy để cẩn thận, ta sẽ chuyển tất cả dữ liệu về dạng số thực 32 bit (đồng bộ toàn bộ dữ liệu thành dạng số thực 32 bit). Ta sẽ có đoạn lệnh Python sau:



Hình 2.8 Công tác chuẩn bị về mặt dữ liệu

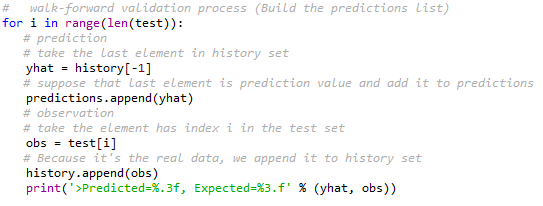
Như vậy tập train sẽ chứa các giá trị VN30 từ ngày 06/02/2012 cho đến ngày 14/02/2012. Và tập test sẽ chứa các giá trị VN30 từ ngày 15/02/2012 đến ngày 24/02/2012.

Ở bước tiếp theo, chúng ta sẽ thực hiện các bước lặp xử lý dữ liệu trong dataset. Ta sẽ tạo một danh sách có biến là history sẽ chứa các phần tử của tập train dataset (trong hình 2.8). Sau thao tác này ta sẽ có được 2 danh sách trung gian là history và predictions để xử lý dữ liệu. Ta có đoạn code sau:



Hình 2.9 Tạo 2 danh sách là history và predictions

Tiếp theo ta sẽ có đoạn code Python sau:



Hình 2.10 Sử dụng các bước lặp để tạo nền tảng cho việc tính RMSE

Các dự đoán được thực hiện bởi mô hình được gọi là yhat (hình 2.10) (bởi vì biến là chữ Y có gắn thêm dấu mũ nên gọi là yhat).

Việc ta làm ở đây là tính độ sai lệch của các dự đoán giá trị VN30 ở thời điểm tương lai do đó ta sẽ không thể tính được giá trị theo công thức (3). Vì vậy ta cần có một cơ sở nào đó để ta có dựa vào. Do đó nên ta sẽ dựa vào tập dữ liệu được cung cấp để thực hiện công việc này. Ta sẽ giả sử rằng giá trị cuối cùng của tập train (hình 2.8) hay nói cách khác là phần tử cuối cùng của list history (hình 2.9) (giá trị VN30 của ngày 14/02/2012) sẽ là giá trị dự đoán cơ sở cho giá trị VN30 của ngày 15/02/2012 hay nói cách khác là . Và giá trị VN30 của ngày 15/02/2012 là giá trị mong đợi/giá trị thật của giá trị VN30 của ngày 15/02/2012 hay nói cách khác là . Vòng lặp ta cũng áp dụng tương tự lấy giá trị ngày 15/02/2012 sẽ là giá trị dự đoán cho giá trị VN30 của ngày 16/02/2012 () và giá trị VN30 của ngày 16/02/2012 sẽ là giá trị thật/mong đợi (). Ta sẽ lặp lại liên tục như vậy cho đến phần tử cuối cùng của danh sách test.

Như vậy qua đoạn code ở hình 2.10 ta được 2 danh sách là test (hình 2.8) là danh sách chứa các giá trị thật và predictions là danh sách chứa các giá trị dự báo ứng với hai biến là và trong công thức (3).

Các giá trị dự báo và các giá trị sự thật sẽ được in trong mỗi vòng lặp để có thể kiểm tra tính hợp lý của các dự đoán để phòng hờ các trường hợp các mô hình vấn đề.

2.1.4 Tính độ sai lệch

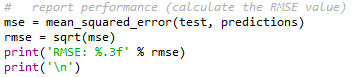
Bước đầu tiên trước khi phân tích dữ liệu và mô hình hóa là tính toán nền tảng hiệu suất của dự đoán (ngưỡng sai số của phương pháp).

Là một giá trị mẫu để đánh giá các mô hình bằng cách sử dụng bộ kiểm thử được xây dựng được nêu ra ở mục 2.1.3 và một thước đo hiệu suất để theo đó tất cả các mô hình tiên đoán phức tạp hơn có thể được so sánh.

Nền tảng dự báo cơ bản cho dự báo hàng loạt thời gian theo phương pháp này được gọi là dự báo cơ bản, hay còn gọi là dự báo kiên trì.

Các thao tác được làm ở mục 2.1.3 sẽ được sử dụng để làm dự báo cho các quan sát ở các bước tiếp theo.

Chúng ta tính giá trị RMSE theo đoạn code Python sau:

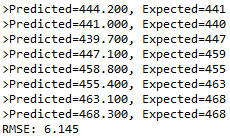


Hình 2.11 Tính toán hiệu suất của phương pháp

Chạy thử nghiệm các đoạn code ở mục 2.1.3 và 2.1.4.

Chương trình sẽ in ra giá trị RMSE của phương pháp.

Trong trường hợp này, chúng ta có thể thấy rằng mô hình bền bỉ đạt được một RMSE có giá trị là 9.105. Điều này có nghĩa là trung bình mô hình sẽ có giá trị sai số là 9.105 cho mỗi lần dự đoán được thực hiện.



Hình 2.12 Kết quả của chương trình sau khi chạy

Chúng ta đã chuẩn bị xong nền tảng cho phương pháp và tính được hiệu suất; vì vậy bây giờ chúng ta có thể bắt đầu đào sâu vào dữ liệu.

2.1.5 Phân tích dữ liệu

Chúng ta có thể sử dụng số liệu thống kê tóm tắt và các mảng dữ liệu để có thể tìm hiểu thêm một cách nhanh chóng về cấu trúc của dữ liệu mà ta sẽ đi dự đoán (cung cấp cái nhìn chung/tổng quan về dữ liệu).

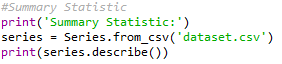
Trong phần này, chúng ta sẽ xem xét dữ liệu từ năm góc độ:

* Các thống kê tóm tắt về dữ liệu.
* Các biểu đồ đường.
* Các biểu đồ mật độ (biểu đồ đường, cột).

2.1.5.1 Các thống kê tóm tắt về dữ liệu

Công việc tóm tắt thống kê cung cấp một cái nhìn tổng quan về các giới hạn của các dữ liệu sẽ được quan sát. Nó có thể giúp ta có được một ý tưởng nhanh về những gì chúng ta đang làm việc.

Đoạn code Python dưới đây sẽ tính toán và in số liệu thống kê tóm tắt của dữ liệu.

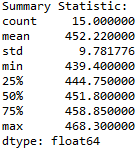


Hình 2.13 Thực thống kế dữ tóm tắt về dữ liệu

Cho chương trình chạy đoạn code Python trên.

Một số nhận xét từ các số liệu thống kê được chương trình đưa ra bao gồm:

* Số lượng dữ liệu quan sát có khớp so với dữ liệu được cung cấp không, có nghĩa là chúng ta đang xử lý dữ liệu chính xác và đầy đủ về mặt số lượng dữ liệu.
* Giá trị trung bình cộng là khoảng 452.22
* Độ lệch chuẩn (std) là 9.781776
* Các phân vi (min, 25%, 50%, 75%, max) cùng với độ lệch tiêu chuẩn (std) cho thấy độ trải dài của dữ liệu (khoảng cách giữa các giá trị của dữ liệu).



Hình 2.14 Kết quả thống kê tóm tắt về dữ liệu

Trong đó độ lệch chuẩn được tính bởi công thức:

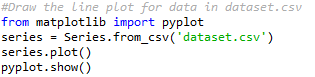
Với:

* là std, như vậy sẽ là phương sai.
* là số lượng dữ liệu: count
* là mean
* là giá trị VN30 ở vị trí thứ trong tập series (tập dữ liệu)

2.1.5.2 Biểu đồ đường

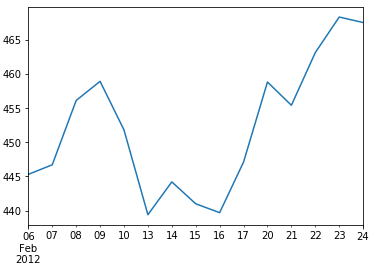
Biểu đồ đường có thể cung cấp cho chúng ta cái nhìn tổng quan nhất về dữ liệu.

Đoạn code Python sau sẽ vẽ giúp ta biểu đồ đường thể hiện các giá trị của tập dữ liệu:



Hình 2.15 Vẽ biểu đồ đường thể hiện các giá trị của tập dữ liệu

Chạy chương trình và quan sát cấu trúc, đặc điểm của các giá trị của tập dữ liệu. Ta được kết quả sau:



Hình 2.16 Biểu đồ đường thể hiện giá trị VN30 từ ngày 06/02/2012 đến ngày 24/02/2012

Một số quan sát/rút ra được từ biểu đồ bao gồm:

* Giá trị VN30 có xu hướng tăng theo thời gian. Giá trị VN30 đầu tiên là 445.3 (của ngày 06/02/2012) và giá trị VN30 kết thúc là 467.5 (của ngày 24/02/2012). Từ ngày 06/02/2012 đến ngày 07/03/2012 đã tăng 22.2. Xu hướng (trend) khá rõ rệt.
* Các giá trị không tuân theo bất cứ một loại chu kỳ nào (không có tính chu kỳ).
* Đường biểu diễn thể hiện các giá trị tăng không liên tục (giá trị có xu hướng giảm từ ngày 09/02/2012 nhưng lại có xu hướng tăng lại từ ngày 16/02/2012)
* Từ biểu đồ ta thấy rằng chuỗi giá trị VN30 của tập dữ liệu là không ổn định.

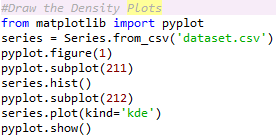
Do dữ liệu không có tính chu kỳ nên ta không cần phải vẽ biểu đồ chu kỳ cho dữ liệu.

Xu hướng gia tăng không liên tục của dữ liệu khiến chúng ta khó có thể đưa các dự đoán gần chính xác.

2.1.5.3 Các biểu đồ thể hiện về mật độ

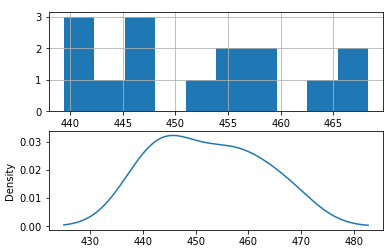
Xem xét các đồ thị mật độ có thể cung cấp cái nhìn sâu hơn về cấu trúc của dữ liệu.

Đoạn code Python sẽ giúp ta vẽ các biểu đồ thể hiện mật độ cũng như phân phối xác suất của các giá trị:



Hình 2.17 Vẽ biểu đồ thể hiện mật độ của dữ liệu

Chạy chương trình và xem kết quả:



Hình 2.18 Các biểu đồ kết quả

Một số nhận xét từ các biểu đồ bao gồm:

* Sự phân bố của dữ liệu không phải là phân phối Gaussian.
* Hình dạng đuôi dài phía bên phải có thể gợi ý đây là phân phối mũ

Các biểu đồ này có thể hỗ trợ cho chúng ta khám phá và hiểu nhiều hơn về dữ liệu trước khi chúng ta mô hình hoá.

2.1.6 Xây dựng các mô hình ARIMA

Trong phần này, chúng ta sẽ phát triển mô hình dự báo Autoregressive Integrated Moving Average, hay còn gọi là ARIMA.

Chúng ta sẽ xây dựng việc mô hình hóa bằng cách định hình thủ công và định hình tự động mô hình ARIMA. Việc này sẽ thực hiện trên 3 bước điều tra các lỗi còn sót lại của việc chọn mô hình.

Như vậy, phần này sẽ được chia thành 3 bước:

* Ước lượng và điều chỉnh dữ liệu.
* Xây dựng mô hình ARIMA.
* Xem xét lại toàn bộ quá trình xây dựng mô hình.

2.1.6.1 Mô hình ARIMA

Trong số liệu thống kê và kinh tế lượng, và đặc biệt trong phân tích chuỗi thời gian, mô hình tích hợp tự hồi quy (ARIMA) là mô hình tổng quát của mô hình ARMA. Cả hai mô hình này đều sử dụng dữ liệu chuỗi thời gian để giúp ta hiểu rõ hơn về dữ liệu hoặc để dự đoán các điểm trong tương lai trong chuỗi dữ liệu thời gian (dự báo một giá trị trong tương lai). Các mô hình ARIMA được áp dụng trong một số trường hợp khi dữ liệu không ổn định, trong đó việc sai phân (tương ứng với phần "tích hợp" của mô hình) có thể được áp dụng một lần hoặc nhiều lần để loại bỏ tính không ổn định của chuỗi dữ liệu.

Phần AR của ARIMA chỉ ra rằng biến số ở phần tử thứ (t) sẽ được hồi quy về giá trị phía trước nó, tức là phần tử thứ (t – 1) của nó. Phần MA tượng trưng cho các lỗi hồi quy, các giá trị này thực ra là một tổ hợp tuyến tính của các giá trị xảy ra đồng thời ở những thời điểm khác nhau trong quá khứ. Chữ I (Integrated – tích hợp) trong ARIMA tượng trưng cho các giá trị dữ liệu đã được thay thế bằng các giá trị sai phân của nó với các giá trị trước đó của nó (và quá trình sai phân này có thể được thực hiện nhiều hơn một lần). Mục đích của việc làm này là làm cho mô hình có thể phù hợp với dữ liệu nhất có thể.

Mô hình ARIMA không có tính mùa (non-seasonal – không có tính chu kỳ) thường được ký hiệu là ARIMA(p, d, q), trong đó p, d, và q là các số nguyên không âm, p là bậc (độ trễ của thời gian) của mô hình tự hồi quy, d là bậc sai phân (số lần dữ liệu đã bị vi phân), và q là bậc của mô hình trung bình động (Moving-Average Model). Mô hình có tính mùa vụ (chu kỳ) ARIMA thường được ký hiệu là ARIMA(p, d, q)(P, D, Q)m, trong đó m là số chu kỳ trong mỗi mùa, và các chữ hoa P, D, Q lần lượt đặc trưng cho các tính tự hồi quy, sai phân, và giai đoạn trung bình động cho các thành phần mùa trong mô hình ARIMA.

Khi hai trong ba tham số trên (p, d, q) bằng 0, mô hình sẽ dựa vào tham số còn lại khác không, lúc này mô hình ARIMA sẽ thành 1 trong ba mô hình sau AR(p), I(q), hoặc MA(d). Ví dụ, mô hình ARIMA(1,0,0) tương đương với mô hình AR(1), tương tự ARIMA(0,1,0) sẽ tương đương với I(1), và ARIMA(0,0,1) sẽ tương đương với MA(1).

Các mô hình ARIMA có thể được ước tính theo phương thức Box-Jenkins.

Cho dãy dữ liệu thời gian của dữ liệu Xt trong đó t là số nguyên và Xt là số thực, một mô hình ARMA (p, q) được định nghĩa bởi(4):

hoặc tương đương bởi (tóm gọi lại là):

Trong đó là toán tử lag, là các tham số của phần tự hồi quy của mô hình, là các tham số của phần trung bình động và là các error term (sai số). Các error term thường được gọi là các biến độc lập hay là các biến phân phối mẫu được hình thành từ một phân bố chuẩn với giá trị mean (trung bình cộng) là bằng 0.

Toán tử được định nghĩa như sau:

Hoặc:

Lưu ý rằng toán tử lag có quyền được lũy thừa tùy ý do đó ta có trường hợp sau:

Do đó ta tổng quát thành:

Giả sử rằng đa thức có một gốc đơn vị (một nhân tố ) lũy thừa d. Có thể được biểu diễn như sau:

Một quá trình ARIMA(p, d, q) được biểu diễn bởi đa thức thức nhân tử (4) kèm theo với đa thức ta có , và được viết thành:

Công thức (5) có thể được nghĩ như là một trường hợp cụ thể của một quá trình ARMA (p + d, q) có đa thức tự hồi quy với các unit root là d. (Vì lý do này, không có mô hình ARIMA với d> 0 sẽ thỏa mãn trạng thái dừng rộng.)

Phương trình (5) có thể khái quát lại như sau:

Theo đó định nghĩa một quá trình ARIMA(p, d, q) là phép

2.1.6.2 Cấu hình thủ công mô hình ARIMA

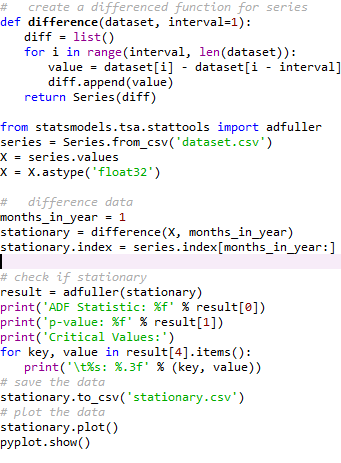
Như đã giới thiệu ở trên mô hình ARIMA(p, d, q) yêu cầu ba tham số và sẽ được chúng ta cấu hình theo cách thủ công ở mục này.

Việc phân tích số liệu chuỗi thời gian yêu cầu chúng ta phải đang làm việc trên một chuỗi thời gian tĩnh (có tính ổn định – ít phụ thuộc vào hàm thời gian).

Các chuỗi thời gian mà ta xét gần như chắc chắn là không ổn định. Chúng ta có thể làm cho nó dừng lại (ổn định - stationary) bằng cách sai phân chuỗi thời gian và sử dụng một phương pháp kiểm tra thống kê (ở đây tài liệu xin sử dụng phương pháp Dickey-Fuller tăng cường) để xác nhận rằng kết quả ta sai phân xong là chuỗi thời gian tĩnh.

Chúng ta khó có thể phát hiện tính mùa vụ trong chuỗi gian, trong trường hợp này ta sẽ dùng phép sai phân giữa ngày sau với ngày trước. Do đó ta sẽ mất đi dữ liệu của ngày 06/02/2012.

Đoạn code sau sẽ làm giảm tính không ổn định (non-stationary) của chuỗi thời gian bằng cách dùng phương pháp sai phân bậc 1 và lưu nó vào tệp tin stationary.csv.



Hình 2.19 Thao tác làm giảm tính không ổn định của chuỗi thời gian

Chạy chương trình, ta được kết quả đầu ra là kết quả của phương pháp kiểm tra thống kê cho ta thấy chuỗi thời gian đã chuyển sang trạng thái tĩnh. Cụ thể, phương pháp kiểm tra ở đây chúng ta sử dụng là phương pháp Dickey-Fuller tăng cường/mở rộng (Agumented Dickey – Fuller (ADF) test).

Kiểm định nghiệm đơn vị là một kiểm định được sử dụng khá phổ biến để kiểm định một chuỗi thời gian là dừng hay không dừng. Dickey và Fuller (1981) đã đưa ra **kiểm định Dickey và Fuller (DF) và kiểm định Dickey và Fuller mở rộng (ADF)**. Nghiên cứu này sử dụng kiểm định ADF để thực hiện kiểm định nghiệm đơn vị nên chỉ tập trung vào lý thuyết của mô hình này. Cụ thể, theo Dickey và Fuller (1981) mô hình kiểm định nghiệm đơn vị mở rộng ADF có dạng:

Trong đó:

* là dữ liệu trong chuỗi thời gian
* chiều dài của độ trễ
* là nhiễu trắng

Mô hình (6) khác với mô hình (7) là có thêm biến xu hướng về thời gian t. Biến xu hướng là một biến có giá trị từ 1 đến n, trong đó 1 đại diện cho quan sát đầu tiên trong dữ liệu và n đại diện cho quan sát cuối cùng trong chuỗi dữ liệu.

Nhiễu trắng là số hạng là giá trị sai số ngẫu nhiên xuất phát từ các giả định cổ điển rằng nó có giá trị trung bình cộng bằng 0 (giá trị mean), phương sai (bình phương độ lệch chuẩn (bình phương giá trị std) là hằng số và không tự tương quan.

Kết quả kiểm định ADF thường rất nhạy cảm với sự lựa chọn chiều dài độ trễ k nên tiêu chuẩn thông tin AIC (Akaike’s Information Criterion) của Akaike (1973) được sử dụng để chọn lựa k tối ưu cho mô hình ADF. Cụ thể, giá trị k được lựa chọn sao cho AIC nhỏ nhất.

Kiểm định thống kê làm cho các giả định/dự đoán chúng ta về dữ liệu trở nên mạnh mẽ/chắc chắn hơn. Các kết quả kiểm định được sử dụng để thông báo cho chúng ta biết về mức độ mà một giả thuyết không (null hypothesis) có thể được chấp nhận hay không. Kết quả thu được sẽ phản ánh được ý nghĩa của dữ liệu.

Hơn nữa, qua các bài kiểm định thống kê Dickey – Fuller mở rộng, ta có thể biết được rằng chuỗi thời gian chúng ta đang xét là chuỗi ổn định hay là chuỗi không ổn định

Chuỗi thời gian càng ổn định thì các kết quả ta dự đoán càng chính xác, bởi vì các giá trị chúng ta mà chúng ta dự đoán sẽ ít phụ thuộc vào thời gian (t).

Thử nghiệm Augmented Dickey-Fuller là một loại thử nghiệm thống kê được gọi là phương pháp unit root test.

Về tổng quát, cách tiếp cận của phương pháp unit root test giả định rằng chuỗi thời gian ta đang xét có dạng có thể được viết thành như sau:

Trong đó:

* là thành phần xác định (xu hướng, thành phần theo mùa, v.v.)
* là thành phần stochastic (thống kê ngẫu nhiên).
* là quá trình lỗi cố định.

Nhiệm vụ của phương pháp đó là xác định xem thành phần stochastic có chứa unit root hay không.

Trong lý thuyết xác suất và thống kê, unit root là điểm đặc trưng của một số quá trình stochastic có thể gây ra các vấn đề trong quá trình suy luận thống kê liên quan đến các mô hình chuỗi thời gian. Một quá trình tuyến tính ngẫu nhiên có một unit root nếu 1 là gốc của phương trình đặc trưng của chuỗi thời gian. Một chuỗi như vậy được gọi là không ổn định (không có tính dừng) và không phải lúc nào chuỗi này cũng có một xu hướng cụ thể (tăng lên hoặc giảm xuống một cách rõ nét).

Mục đích của việc quan sát kết quả của một bài kiểm định unit root test đó là xác định mức độ mạnh yếu của một chuỗi thời gian thông qua chiều xu hướng của nó.

Unit root test và Augmented Dickey-Fuller (ADF) là một trong những phương pháp hay được sử dụng để kiểm tra tính dừng (ổn định) của chuỗi thời gian. Thông qua một mô hình tự hồi quy ta có thể thấy được chuỗi thời gian đã dừng hay chưa và ta có thể tối ưu hóa chuỗi đó thông qua nhiều giá trị sai phân độ trễ khác nhau.

Việc chấp nhận hay bác bỏ giả thuyết không (null hypothesis) có thể được xác định thông qua bài kiểm tra có thể nói cho chúng ta biết chuỗi thời gian có thể chứa một unit root hay là không , nếu có (tồn tại (chấp nhận) null hypothesis) ta có thể kết luận rằng chuỗi thời gian có tính không ổn định (vì nó có chứa một số cấu trúc phụ thuộc vào thời gian). Việc ta chấp nhận giả thiết thay thế (từ chối giả thuyết không) có thể giúp ta kết rằng chuỗi thời gian mà ta đang xét là ổn định (có tính dừng).

* Null hypothesis (H0): Nếu được chấp nhận, điều này cho thấy chuỗi thời gian có một unit root, nghĩa là chuỗi có tính không ổn định. Chuỗi có một số cấu trúc phụ thuộc vào biến thời gian.
* Giả thiết thay thế (alternate hypothesis) (H1): null hypothesis bị từ chối; điều này có nghĩa là chuỗi thời gian không có unit root, nghĩa là chuỗi thời gian có tính dừng. Và nó không có cấu trúc phụ thuộc vào thời gian.

Trị số p (mức ý nghĩa) (mà tiếng Anh gọi là p-value) là một con số xác suất, tức là viết tắt chữ “probability value”. Chúng ta thường gặp những phát biểu được kèm theo con số, chẳng hạn như “Kết quả phân tích cho thấy tỉ lệ gãy xương trong nhóm bệnh nhân được điều trị bằng thuốc Alendronate là 2%, thấp hơn tỉ lệ trong nhóm bệnh nhân không được chữa trị (5%), và mức độ khác biệt này có ý nghĩa thống kê (p = 0.01)”, hay một phát biểu như “Sau 3 tháng điều trị, mức độ giảm áp suất máu trong nhóm bệnh nhân là 10% (p < 0.05)”. Trong văn cảnh trên đây, đại đa số nhà khoa học hiểu rằng trị số p phản ánh xác suất sự hiệu nghiệm của thuốc Alendronate hay một thuật điều trị.

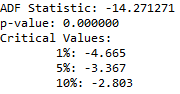
Việc chấp nhận hay bác bỏ null hypothesis ta sẽ thực hiện thông qua việc sử dụng giá trị p từ bài kiểm tra. Giá trị p ở ngưỡng dưới (từ khoảng [1%, 5%] hoặc thấp hơn) cho thấy chúng ta từ chối giả thuyết không (tức là chuỗi của ta tĩnh hay nói cách khác là có tính dừng), ngược lại nếu giá trị p ở ngưỡng trên cho thấy chúng ta chấp nhận giả thuyết không (tức là chuỗi của ta không ổn định hay nói cách khác là không có tính dừng).

* p-value > 0,05: Chấp nhận giả thuyết không (H0), dữ liệu có một unit root và không ổn định.
* p-value <= 0,05: Từ chối null hypothesis (H0), dữ liệu không có unit root và có tính dừng.

Chạy chương trình, chương trình sẽ in giá trị thử nghiệm thống kê (ADF Statistic) là -14.271271. Giá trị này càng âm, chúng ta càng có khả năng từ chối giả thuyết không (tức là chúng ta có thể có một tập dữ liệu ổn định).

Kết quả cho thấy giá trị thử nghiệm thống kê là -14.271271 nhỏ hơn giá trị tới hạn (critical value) tại 1% là -4.665. Điều này cho thấy chúng ta có thể bác bỏ giả thuyết không có mức ý nghĩa ít hơn 1% (nghĩa là xác suất thấp bài toán mà ta đang kiểm định, thống kê ở đây không phải là một giá trị thống kê ngẫu nhiên (statistical fluke)).

Loại bỏ giả thuyết không có nghĩa là chuỗi mà ta xét không có unit root, và lần lượt ta kết luận rằng chuỗi thời gian mà ta đang xét là chuỗi tĩnh hoặc không có cấu trúc phụ thuộc thời gian.



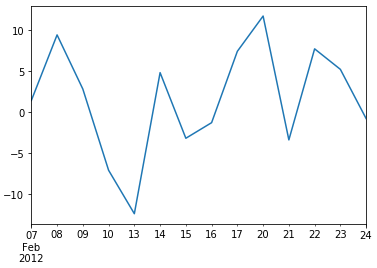
Hình 2.20 Kết quả của phương pháp kiểm tra Dickey – Fuller tăng cường.

Để tiện tham chiếu, ta có thể được đảo ngược quá trình sai phân mà ta thực hiện ở trên bằng cách cộng bù lại giá trị mà ta sai phân cho phần tử thử i trong chuỗi thời gian. Công đoạn này sẽ được thực hiện trong quá trình ta thực hiện các dự đoán từ mô hình được hình thành từ các dữ liệu mà ta đã sai phân (trả lại giá trị dữ liệu ban đầu cho chuỗi). Hàm đảo ngược sai phân sẽ có đoạn code python như sau:



Hình 2.21 Hàm đảo ngược sai phân

Kết quả đồ thị của dữ liệu sau khi được sai phân:



Hình 2.22 Đồ thị kết quả sai phân của dữ liệu

Đồ thị không cho thấy bất kỳ tính mùa vụ hay xu hướng rõ ràng nào tồn tại trong chuỗi thời gian, đây là một sự khởi đầu tốt cho việc xây dựng mô hình.

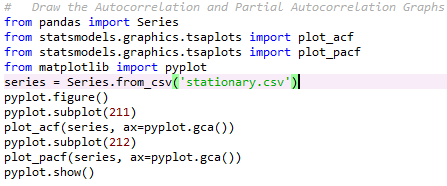
Chúng ta sẽ sử dụng tập dữ liệu này làm đầu vào cho mô hình ARIMA. Bài kiểm tra cũng như đồ thị cho ta biết rằng chúng ta không cần phải tiếp tục các bậc sai phân cao hơn để làm ổn định chuỗi, vì vậy tham số d của mô hình ta có thể gán cho nó giá trị 0.

Bước tiếp theo thứ hai là chọn các giá trị lag cho các tham số p và q tương ứng với 2 mô hình còn lại là Autoregression (AR) và Moving Average (MA).

Ta có thể chọn 2 giá trị p và q này bằng cách dùng hàm Autocorrelation Function (ACF) và hàm Partial Autocorrelation Function (PACF).

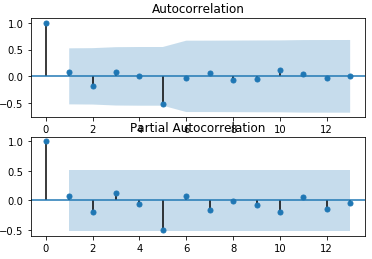
Chúng ta sẽ sử dụng tập dữ liệu đã được sai phân được lưu trong file stationary.csv.

Ta sẽ vẽ biểu đồ sử dụng hai hàm ACF và PACF bằng đoạn code python sau:



Hình 2.23 Thực hiện vẽ 2 biểu đồ Autocorrelation và Partial Autocorrelation

Chạy chương trình và xem xét các biểu đồ mà chương trình vẽ ra từ đó ta sẽ đưa ra một số nhận xét và chọn giá trị cho các biến p và q của mô hình ARIMA.



Hình 2.24 Hai biểu đồ độ trễ Autocorrelation (AC) và Partical Autocorrelation (PAC) có được sau khi thực hiện chương trình

Một số nhận xét được đúc kết từ hai biểu đồ:

* ACF cho thấy một giá trị độ trễ lớn: giá trị đó là 1.
* Tương tự PACF cũng cho thấy giá trị độ trễ lớn là 1, một giá trị độ trễ tương đối khác vào khoảng 5 ngày sau của biểu đồ.
* Cả ACF và PACF đều có một giá trị như nhau tại cùng một thời điểm, có thể gợi ý rằng có một sự kết hợp giữa AR và MA.

Như vậy giá trị ban đầu tốt cho các giá trị p và d là 1 (vì chỉ có 1 là giá trị có độ trỡ lớn trong biểu đồ).

Biểu đồ PAC cho ta thấy chuỗi thời gian sau khi sai phân vẫn còn một số chỗ có tính mùa vụ (ví dụ như giá trị ở chỗ sau 2 ngày và sau 10 ngày có tính mùa vụ).

Chúng ta có thể chọn lựa mô hình tốt hơn để giải quyết tính mùa vụ này, bằng cách có thể mô hình hóa trực tiếp và loại bỏ các tính mùa vụ này một cách rõ ràng ra khỏi mô hình bằng cách sai phân từng phần tử.

Sau quá trình phân tích nhanh này, ta thấy rằng lựa chọn sử dụng mô hình ARIMA (1,0,1) cho dữ liệu là một điểm khởi đầu tốt.

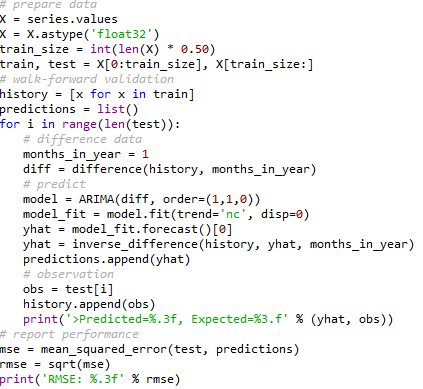
Các quan sát lịch sử sẽ được phân biệt theo mùa vụ trước khi phù hợp với từng mô hình ARIMA. Sự khác biệt sẽ được đảo ngược cho tất cả các dự đoán được thực hiện để làm cho chúng trực tiếp so sánh với sự quan sát dự kiến trong các đơn vị đếm số lần bán ban đầu.

Chúng ta sẽ sai phân từng phần tử một trước khi các phần tử này sẽ được đem đi chỉnh cho khớp với từng mô hình ARIMA ở mỗi bước lặp. Sau khi được điều chỉnh cho khớp với mô hình ARIMA ta sẽ đảo ngược quá trình sai phân để khôi phục lại dữ liệu ban đầu cho từng phần tử

Thử nghiệm cho thấy cấu hình ARIMA này không hội tụ và kết quả là lỗi do các thư viện cơ bản. Ở bước tiếp theo ta sẽ sai phân ở từng vòng lặp vào dữ liệu mặc dù nó đã ổn định để làm cho mô hình càng ổn định hơn. Do đó mô hình của chúng ta sẽ được mở rộng thành ARIMA (1,1,1).

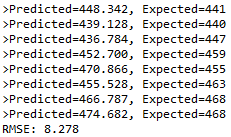
Chúng ta cũng sẽ vô hiệu hóa việc tự động bổ sung xu hướng (trend) từ mô hình bằng cách thiết lập các tham số 'trend' thành 'nc' để không có hằng số trong chuỗi trước khi ta gọi hàm khớp dữ liệu với mô hình (hàm fit()). Từ thử nghiệm này, ta sẽ nhận thấy rằng điều này có thể dẫn đến hiệu suất dự báo tốt hơn nếu trong chuỗi có một số vấn đề còn tồn tại.

Ta có đoạn code python minh họa cho quy trình trên như sau:



Hình 2.25 Thực hiện sai phân trong từng vòng lặp

Ta được kết quả của chương trình là:



Hình 2.26 Kết quả của chương trình sau khi chạy đoạn code hình 2.25

2.1.6.3 Xây dựng mô hình ARIMA

Các biểu đồ ACF và PACF cho thấy mô hình ARIMA (1,0,1) hoặc các mô hình tương tự (trong đoạn code là ARIMA(1,1,0)) là những mô hình tốt mà chúng ta có thể áp dụng.

Bước tiếp theo, chúng ta sẽ tìm một bộ siêu tham số sao cho xây dựng một mô hình tốt nhất có thể ARIMA và kiểm tra rằng không có mô hình kết quả quả nào tốt hơn mô hình mà ta đã tìm được.

Trong phần này, chúng ta sẽ tìm kiếm các giá trị của p, d, và q (bỏ qua những giá trị không hội tụ), và tìm ra bộ q, d, q sao cho có thể dẫn đến hiệu suất RMSE tốt nhất trên tập kiểm tra. Chúng ta sẽ sử dụng một dạng tìm kiếm lưới để tìm ra tất cả các bộ giá trị p, d, q trong một tập con các giá trị số nguyên mà ta định sẵn.

Cụ thể, chúng ta sẽ tìm kiếm tất cả các bộ giá trị p, d, q được kết hợp từ các miền giá trị sau:

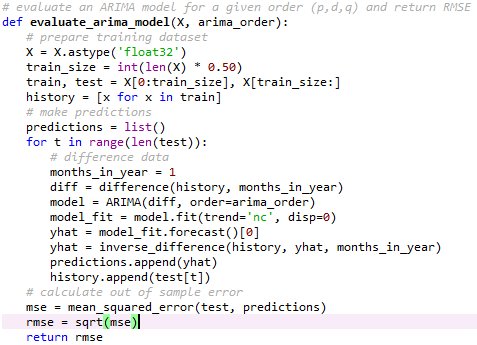
p: 0 đến 6.

d: 0 đến 2.

q: 0 đến 6.

Tổng cộng ta có không gian mẫu là (7 \* 3 \* 7), hoặc nói cách khác là ta có 147 bộ giá trị.

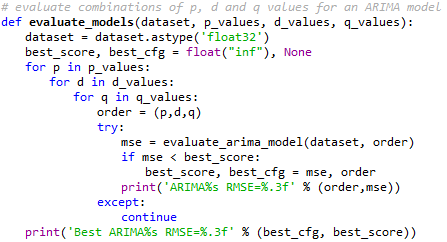
Ta có đoạn code sau:



Hình 2.27 Hàm tính giá trị RMSE cho mỗi mô hình

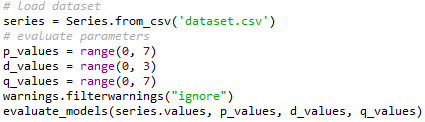
Hàm evaluate\_arima\_model này có tham số X là chuỗi thời gian và arima\_order là bộ ba giá trị (p, d, q), kết quả của hàm là giá trị RMSE tính được của mô hình ARIMA(p, d, q) (ứng với giá trị arima\_order được truyền vào) của chuỗi thời gian X.

Nguyên lý làm việc của hàm này dựa vào các quá trình mà ta đã thực hiện ở phần trước là: sai phân chuỗi và gọi hàm ARIMA và fit() để xây dựng mô hình. Trong từng vòng lặp ta sẽ sai phân để cho chuỗi được ổn định sau đó ta gọi hàm ARIMA để xây dựng mô hình, tiếp tục ta gọi hàm fit() để hoàn chỉnh mô hình và cuối cùng ta sẽ gọi hàm mean\_squared\_error để tính giá trị MSE. Để kết thúc hàm ta sẽ lấy căn MSE và trả về giá trị RMSE.



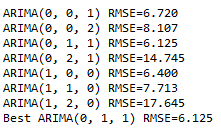
Hình 2.28 Hàm tìm ra mô hình ARIMA tốt nhất có thể

Hàm evaluate\_models dùng để giúp ta tìm ra mô hình ARIMA có giá trị RMSE nhỏ nhất có thể tương đương tìm ra với mô hình dự đoán sai số thấp nhất có thể, giúp ta có được kết quả dự đoán tốt hơn một cách tự động mà không cần phải cấu hình thủ công như phần trước.



Hình 2.29 Công đoạn xử lý tìm ra mô hình ARIMA tốt nhất

Chạy chương trình. Chương trình sẽ in ra từng mô hình ARIMA và giá trị RMSE tương ứng với từng mô hình. Kết thúc chương trình, chương trình sẽ in ra cho chúng ta biết mô hình ARIMA tốt nhất có thể mà nó tìm được.



Hình 2.30 Mô hình ARIMA tốt nhất tìm được

Kết quả cho thấy cấu hình tốt nhất được phát hiện là ARIMA(0, 1, 1) với RMSE là 6.125, thấp hơn một chút so với ARIMA(1,1,0) được ta cấu hình thủ công ở phần trước. Sự sai phân mà ta thực hiện có thể có hoặc không có ý nghĩa về mặt thống kê.

Như vậy mô hình ARIMA tốt nhất mà ta tìm được là ARIMA(0, 1, 1)

2.1.6.4 Xem xét lại mô hình ARIMA

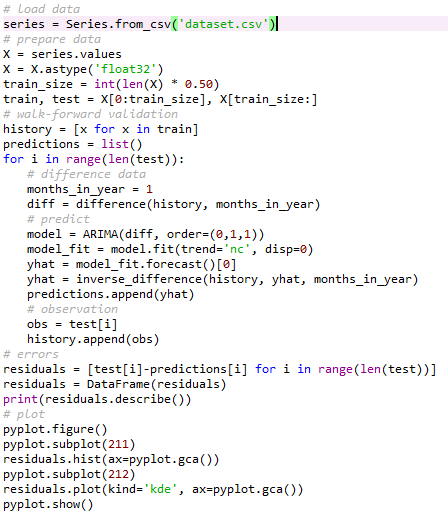
Một bài kiểm tra cuối cùng về tính tốt (tối ưu) của một mô hình là xem lại các lỗi dự báo còn sót lại (các sai số còn sót lại).

Ở đây ta sẽ xem lại sự mật độ phân phối của chuỗi, hình dạng phân phối nên là dạng phân phối Gaussian với giá trị mean (giá trị trung bình) bằng 0 và phương sai () hoặc độ lệch chuẩn (std ()) bằng 1.

Chúng ta sẽ kiểm tra lại xem mô hình ARIMA(0,1,1) còn những lỗi dư thừa (sai số) bằng cách tạo chuỗi có tên là residuals được tạo ra bằng cách lấy tập test (nửa tập sau của dataset.csv) trừ đi tập predictions (tập giả sử giá trị dự đoán (xem lại ở phần trước))

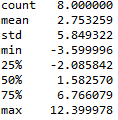
Sau đó, Chúng ta sẽ kiểm tra các giá trị liên quan về chuỗi residuals như (std, mean,…) và các biểu đồ để kiểm tra các lỗi còn dư thừa từ mô hình ARIMA(0, 1, 1) mà ta đã tìm được ở trên.

Sau đây là đoạn code Python mà ta sẽ thực hiện.

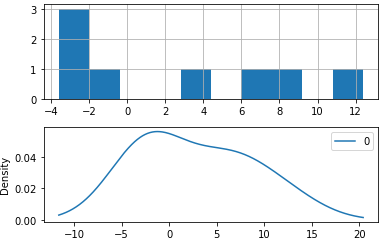


Hình 2.31 Đoạn code kiểm tra mô hình

Ta cho chạy chương trình. Và đây sẽ là kết quả chương trình trả về.



Hình 2.32 Kết quả tóm tắt các giá trị thống kê của chương trình

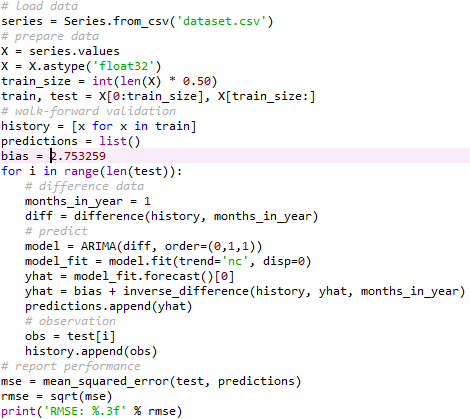


Hình 2.33 Kết quả biểu đồ của bài kiểm tra

Từ kết quả, chúng ta có thể thấy rằng sự phân bố mật độ bị lệch nhiều về phía bên trái và giá trị mean là 2.753259 (khác 0) và độ lệch chuẩn là 5.849322. Chưa đạt được mục đích là đưa phân phối về dạng gaussian. Vì vậy chúng ta sẽ tiếp tục đưa phân phối về dạng gần gaussian nhất có thể.

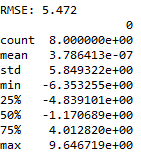
Cho ta thấy được mô hình có tính thiên vị (bias).

Chúng ta sẽ sửa chữa sự sai lệch của phân phối bằng cách thêm vào giá trị sai số dư trung bình là 2.753259 cho mỗi dự báo được thực hiện. Ta sẽ miêu tả quá trình này qua đoạn code sau:



Hình 2.34 Cộng bù giá trị bias để đưa phân phối về guassian

Thống kê các giá trị quan trọng của chuỗi sau khi ta cộng bù:

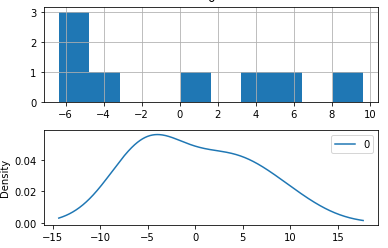


Hình 2.35 Kết quả của chuỗi sau khi cộng bù.

Ta thấy rằng hiệu suất của các dự đoán (RMSE) được cải thiện từ 6.125 xuống 5.472.

Bên cạnh đó ta cũng thấy được giá trị mean được dịch chuyển rất gần với không (giá trị mean tính được là 3.786413e-007 = 0.0000003786413).

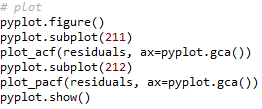
Và đây là kết quả đồ thị mật độ phân phối của chuỗi:



Hình 2.36 Đồ thị mật độ phân phối của chuỗi

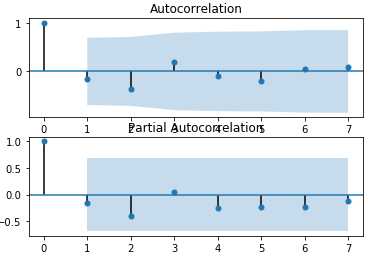
Ta thấy rằng 2 bên của đồ thị đã được dịch chuyển gần đối xứng và giá trị mean đã rất gần với 0 trong nỗ lực đưa hàm phân phối về dạng gaussian. Tuy nhiên độ lệch chuẩn vẫn còn rất lớn (xấp xỉ là 5.85) và mật độ phân phối vẫn còn nhiều chỗ lồi lớn. Mặc dù kết quả không được như mong đợi nhưng ta đã nỗ lực trong việc đưa chuỗi về dạng gaussian và kết quả thu được là ta đã nỗ lực giảm được giá trị RMSE từ 6.125 xuống 5.472.

Cuối cùng ta sẽ xem lại 2 biểu đồ ACF và PACF để xem lại cấu trúc của chuỗi thời gian mà ta test mô hình như thế nào. Ta sẽ thực hiện các dòng code sau:



Hình 2.37 Đoạn code vẽ biểu đồ ACF và PACF

Và đây là kết quả của chương trình:



Hình 2.38 Biểu đồ ACF và PACF của chuỗi dữ liệu residuals

2.1.7 Mô hình chuẩn cuối cùng

Ở phần trước ta đã thực hiện các bước khảo sát, xây dựng các mô hình, tìm ra mô hình tốt nhất có thể và kiểm thử, kiểm tra mô hình này. Ở phần này, ta sẽ xây dựng và kiểm thử và dự đoán trên mô hình cuối cùng mà ta đã tìm ra ở phần trước.

Sau khi các mô hình đã được phát triển và ta sẽ lựa chọn một mô hình cuối cùng, mô hình này phải được xác nhận và hoàn thành.

Việc xác nhận mô hình là một công việc không bắt buộc của quá trình, nhưng nó là thao tác quan trọng vì nó giúp ta tránh được sai sót, và giúp ta hiểu rõ hơn về dữ liệu.

Phần này bao gồm các bước sau:

* Hoàn thiện mô hình: đào tạo và lưu mô hình cuối cùng.
* Tạo các dự đoán: Nạp mô hình đã hoàn tất và thực hiện dự đoán.
* Validate Model: Nạp và xác nhận mô hình cuối cùng.

2.1.7.1 Hoàn thiện mô hình

Thông thường, việc hoàn thiện mô hình ARIMA sẽ thực hiện trên tập dữ liệu gốc, trong trường hợp này ta thực hiện trên một phiên bản biến đổi của toàn bộ tập dữ liệu.

Sau khi đã phù hợp, mô hình có thể được lưu vào tệp tin để sử dụng sau này.

Ví dụ dưới đây sẽ huấn luyện mô hình ARIMA(0,1,1) trên tập dữ liệu và lưu toàn bộ đối tượng phù hợp với mô hình và giá trị thiên vị của mô hình vào tập tin.

Tại thời điểm hiện tại, có lỗi trong phiên bản ổn định hiện tại của thư viện statsmodels (v0.6.1) dẫn đến lỗi trong quá trình tải mô hình ARIMA đã lưu từ tệp. Lỗi báo cáo sẽ có dạng:

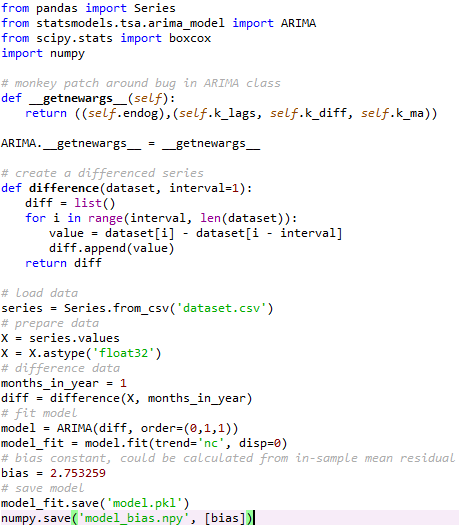


Hình 2.39 Thông báo lỗi thư viện

Chúng ta có thể sửa lỗi này bằng cách gọi hàm \_\_getnewargs \_\_() vào lớp ARIMA trước khi lưu.

Ta có đoạn code python sau sẽ tạo ra 2 file:

* model.pkl Đây là đối tượng ARIMAResult được trả về từ hàm ARIMA.fit(). Đối tượng này bao gồm các hệ số và tất cả các dữ liệu liên quan khác đến việc xây dựng mô hình, tóm gọn là file này chứa dữ liệu là mô hình ARIMA mà ta đã xây dựng.
* model\_bias.npy Trong đây sẽ có 1 bảng có cấu trúc 1 dòng 1 cột, ô này chứa giá trị bias của mô hình.



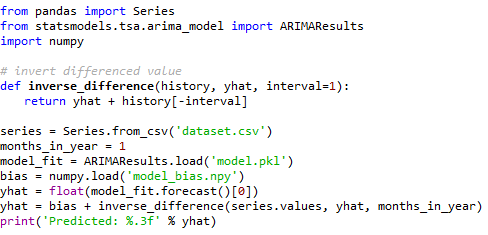
Hình 2.40 Đoạn code xây dựng mô hình ARIMA cuối cùng

2.1.7.2 Tạo các dự đoán

Trên thực tế vẫn có trường hợp ta dùng mô hình chỉ để dự đoán 1 giá trị duy nhất nào đó trong tương lai (1 ngày trước đó, 1 tháng trước đó,…).

Việc thực hiện các đơn dự đoán này rất đơn giản bao gồm khôi phục lại mô hình và giá trị bias mà ta đã lưu vào 2 file trong phần trước và gọi phương thức forecast(). Ta sẽ đảo ngược tính sai phân của tập dữ liệu history (giữ nguyên giá trị gốc của tập nguồn).

Đoạn code dưới đây sẽ mô tả thao tác đơn dự đoán:



Hình 2.41 Đoạn code thực hiện việc đơn dự đoán

Kết quả dự đoán 1 ngày sau được trả về là:



Hình 2.42 Kết quả dự đoán của 1 ngày sau

Nếu chúng ta kiểm tra bên trong file validation.csv, chúng ta có thể thấy rằng giá trị 472.022 này sẽ gần bằng với giá trị thật (ở dòng đầu tiên của file validation.csv) là 472.2

2.1.7.3 Xác nhận mô hình

Ở phần trước ta đã thử nghiệm mô hình bằng cách thực hiện đơn dự đoán (dự đoán giá trị của 1 ngày sau của ngày cuối cùng trong tập dataset.csv).

Trong phần này chúng ta sẽ xem xét và kiểm tra lại mô hình, chúng ta sẽ làm tương tự bằng cách so sánh các giá trị dự đoán của mô hình với các giá trị thực được lưu trong file validation.csv.

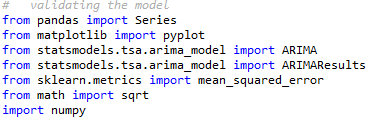
Có hai cách chúng ta có thể tiến hành kiểm tra mô hình:

* Cách 1: Tải mô hình từ file và sử dụng nó để dự đoán giá trị của 10 ngày tiếp theo (tương ứng với 10 ngày được lưu trong validation.csv). Xu hướng dự báo có thể sẽ bị lệch từ 2 đến 3 ngày.
* Cách 2: Tải mô hình và sử dụng nó theo cách dự báo, cập nhật biến đổi và lập mô hình cho từng bước thời gian. Đây là phương pháp được ưa thích vì nó là cách để người ta sử dụng mô hình này trong thực tế vì nó sẽ đạt được hiệu suất tốt nhất.

Cũng giống như đánh giá mô hình trong các phần trước, chúng ta sẽ đưa ra dự đoán theo cách dự báo.

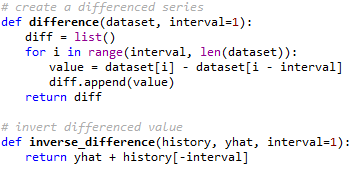
Thao tác này cũng tương tự như các phần trước, bao gồm:

Bước 1: Khai báo các thư viện:



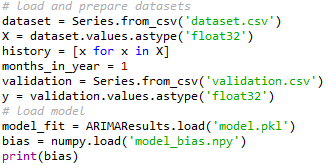
Hình 2.43 Thông báo thư viện

Bước 2: Viết hàm sai phân và đảo ngược của nó:



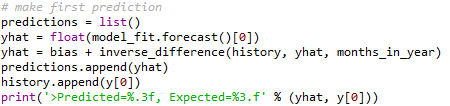
Hình 2.44 Hàm sai phân và đảo sai phân

Bước 3: Load các tập dữ liệu, giá trị bias và mô hình:



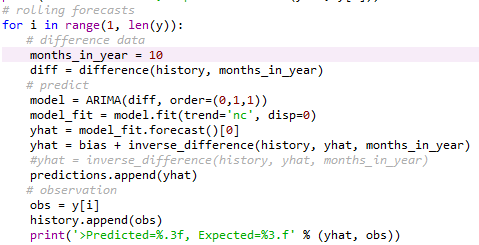
Hình 2.45 Load dữ liệu, bias và mô hình

Bước 4: Thực hiện việc đơn dự đoán



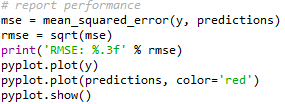
Hình 2.46 Đơn dự đoán

Bước 5: Thực hiện chuỗi dự đoán (dự đoán 10 ngày liên tiếp)



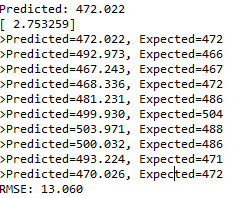
Hình 2.47 Thực hiện chuỗi dự đoán

Bước 6: Báo cáo kết quả và vẽ đồ thị



Hình 2.48 Báo cáo và vẽ biểu đồ

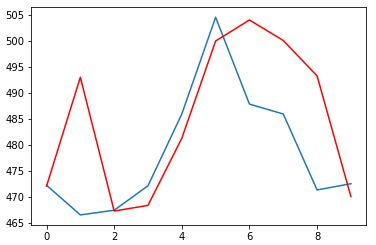
Cho chạy 6 bước trên, ta sẽ được chương trình in ra kết quả sau:



Hình 2.49 Kết quả của chương trình

Kết quả hiệu suất của mô hình (RMSE) là 13.060

Và cuối cùng là biểu đồ của kết quả cung cấp cho ta cái nhìn tổng quan về kết quả dự đoán của mô hình (đường màu đỏ) so với dữ liệu gốc (đường màu xanh):



Hình 2.50 Biểu đồ kết quả dự đoán của mô hình

**TÀI LIỆU THAM KHẢO**

**Tiếng Việt**

1. <https://machinelearningcoban.com/2016/12/26/introduce/>
2. <https://vi.wikipedia.org/wiki/Ch%E1%BB%A9ng_kho%C3%A1n>
3. <https://www.vndirect.com.vn/kien-thuc-co-ban-ve-chung-khoan-phai-sinh/>
4. <https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average>

**Tiếng Anh**

**PHỤ LỤC**

Phần này bao gồm những nội dung cần thiết nhằm minh họa hoặc hỗ trợ cho nội dung luận văn như số liệu, biểu mẫu, tranh ảnh. . . . nếu sử dụng những câu trả lời cho một *bảng câu hỏi thì bảng câu hỏi mẫu này phải được đưa vào phần Phụ lục ở dạng nguyên bản* đã dùng để điều tra, thăm dò ý kiến; **không được tóm tắt hoặc sửa đổi**. Các tính toán mẫu trình bày tóm tắt trong các biểu mẫu cũng cần nêu trong Phụ lục của luận văn. Phụ lục không được dày hơn phần chính của luận văn