

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO CUỐI KỲ
MÔN PHÂN TÍCH XÁC SUẤT VÀ GIẢI THUẬT NGẪU NHIÊN

ĐỀ TÀI: PREDICTING THE DIRECTION
OF STOCK MARKET PRICES USING
RANDOM FOREST

Người hướng dẫn: TS. NGUYỄN CHÍ THIỆN

Người thực hiện: NGUYỄN VŨ KHOA – MSHV: 51503245

VÕ ĐĂNG KHOA – MSHV: 186005032

Lớp: 18600531

Khoá: 2018

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2019

LỜI CẢM ƠN

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều và dù trực tiếp hay gián tiếp của người khác. Trong suốt thời gian từ khi bắt đầu học tập ở giảng đường của trường đến nay, nhóm em đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý Thầy cô, gia đình và bạn bè.

Với lòng biết ơn sâu sắc nhất, nhóm em xin gửi đến quý Thầy cô ở Khoa Công Nghệ Thông Tin – Trường Đại Học Tôn Đức Thắng đã cùng với tri thức và tâm huyết của mình để truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian học tập tại trường. Và đặc biệt, trong học kì này, Khoa đã tổ chức cho nhóm em được tiếp cận tìm hiểu và nghiên cứu đề tài Predicting the direction of stock market prices using random forest (Dự đoán hướng của giá thị trường chứng khoán sử dụng rừng ngẫu nhiên) rất hữu ích đối với sinh viên ngành Công Nghệ Thông Tin cũng như tất cả các sinh viên thuộc các chuyên ngành Tin Học khác. Và trong quá trình học tập tìm hiểu, em nhận được nhiều sự giúp đỡ của các cá nhân và tập thể trong lớp cũng như là bên phía thầy cô trường Đại học Tôn Đức Thắng nói chung và thầy cô khoa Công Nghệ Thông Tin nói riêng.

Xin cảm ơn nhà trường đã tạo điều kiện cho chúng em có cơ hội được học tập, nghiên cứu, bổ sung kiến thức và kiểm chứng thêm những kiến thức đã được học ở trường.

Và cuối cùng chúng em xin chân thành cảm ơn thầy Nguyễn Chí Thiện đã tận tâm hướng dẫn nhóm qua từng bước một để hoàn thành bài báo cáo này. Nếu không có những lời hướng dẫn, dạy bảo của thầy thì em nghĩ bài báo cáo này của nhóm em rất khó có thể hoàn thiện được. Một lần nữa, chúng em xin chân thành cảm ơn thầy.

TP. Hồ Chí Minh, ngày tháng năm

Nguyễn Vũ Khoa

Võ Đăng Khoa

ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Tôi xin cam đoan đây là sản phẩm đồ án của chúng tôi và được sự hướng dẫn của Thầy 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 Vũ Khoa

Võ Đăng Khoa

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

Trong năm 2019, ta có thể chứng kiến một phong trào sôi nổi liên quan đến đề tài “Predicting the direction of stock market prices using random forest”(Dự đoán hướng của giá thị trường chứng khoán sử dụng rừng ngẫu nhiên) trong cộng đồng . Hàng loạt package mới được phát triển nhằm diễn giải cho từng algorithm chuyên biệt như hồi quy tuyến tính (GLM), Random Forest (RF) và Extreme Gradient boosting (XGB).

Nghiên cứu dự đoán hướng của giá thị trường chứng khoán sử dụng rừng ngẫu nhiên cũng đã và đang được các nước trên thế giới thực hiện rất nhiều năm qua và cũng đã có những thành công nhất định.

Ở Việt Nam cũng có nhiều công trình nghiên cứu và thử nghiệm, tuy nhiên, các kết quả vẫn còn hạn chế và cần có nhiều nghiên cứu nữa trong vấn đề này. Nhằm tìm hiểu phương pháp Random Forest. Nhóm đã nghiên cứu phần mềm ứng dụng về đề tài Dự đoán hướng của giá thị trường chứng khoán sử dụng rừng ngẫu nhiên.

Tuy chỉ là một ứng dụng nhỏ nhưng một phần nào nói lên bước phát triển của ứng dụng. Và tương tự đó có thể phát triển và tiến xa hơn nữa trong tương lai như các ứng dụng trong dự đoán thiên nhiên, thiên văn, ... có thể ứng dụng dự đoán trong y khoa. Lấy ví dụ với ngành Ngân hàng, chúng ta có hai bài toán phổ biến cho Random Forest, là tìm kiếm khách hàng tiềm năng và khách hàng lừa đảo, dự đoán giá cổ phiếu....

MỤC LỤC

LỜI CẢM ƠN	i
PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN	iii
TÓM TẮT	iv
MỤC LỤC.....	1
DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT	3
DANH MỤC BẢNG BIỂU	4
DANH MỤC HÌNH ẢNH	5
CHƯƠNG 1 – CÁC KIẾN THỨC CƠ BẢN	6
1.1 RANDOM FOREST.....	6
1.1.1 Giới Thiệu về RANDOM FOREST:	6
1.1.2 Hoạt Động Của Thuật Toán Random Forest.....	7
1.1.3 Giải Thuật RF Cho Phân Lớp Được Diễn Giải Như Sau	8
1.1.4 Đặc Điểm của Random Forest.....	9
1.1.5 Ứng Dụng của Random Forest	9
1.1.6 Ưu Điểm Và Nhược Điểm Của Random Forest	10
1.2 Tìm Hiểu Về Thị Trường Chứng Khoáng:	11
1.3 Bootstrap	13
1.3.1 Bootstrap là gì?.....	13
1.3.1 Nội Dung Phương Pháp Bootstrap	13
1.4 Bagging	14
1.5 Khai Phá Dữ Liệu.....	14
1.5.1 Khai Phá Dữ Liệu Là Gì ?.....	14
1.5.2 Chức Năng.....	16
CHƯƠNG 2: XÂY DỰNG PHẦN MỀM	18

1.1	Chuẩn Bị:	18
1.2	Cài Đặt:	19
1.2.1	Cài Đặt Python:	19
1.2.2	Cài Đặt Package Pandas:	19
1.2.3	Cài Đặt Numpy	19
1.2.4	Cài Đặt Random Forest.....	20
CHƯƠNG 3 – DEMO VÀ ĐÁNH GIÁ, NHẬN XÉT PHẦN MỀM ỨNG		
DỤNG		22
1.1	Chạy Demo.....	22
1.2	Đánh Giá và Nhận Xét:	35
TÀI LIỆU THAM KHẢO.....		36

DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

CÁC CHỮ VIẾT TẮT

Từ viết tắt	Tên tiếng Anh	Tên tiếng Việt
RF	Random Forest	Rừng ngẫu nhiên
KDD	Knowledge Discovery in Database	Khám phá kiến thức trong cơ sở dữ liệu
OOB	out-of-bag	Ước lượng lỗi tạo ra từ việc kết hợp các kết quả từ các cây tổng hợp trong random

DANH MỤC BẢNG BIỂU

Bảng 2.1 chạy Pandas và các Package hỗ trợ	19
Bảng 2.2 chạy Package Numpy	20
Bảng 2.3: Thêm tất cả các tính năng	20
Bảng 2.4: Thêm các tính năng riêng lẻ.....	21
Bảng 2.5: Lệnh chạy	21

DANH MỤC HÌNH ẢNH

Hình 1.1: Random Forest.....	7
Hình 1.2: Hoạt động của thuật toán Random Forest.....	8
Hình 1.3: Các Ứng dụng của Random Forest.....	10
Hình 1.4: Sơ đồ mô phỏng phân phối bootstrap	13
Hình 1.5: Các bước trong Data Mining & KDD.....	16
Hình 2.1: Mẫu data từ trang web NASDAQ.....	18
Hình 3.1: Input data đầu vào.....	23
Hình 3.2: Công thức Exponential Smoothing.....	24
Hình 3.3: Câu lệnh Exponential Smoothing.....	24
Hình 3.4: Kết quả data sau khi Exponential Smoothing	25
Hình 3.5: Câu lệnh thực hiện các tính toán các đặc trưng.....	26
Hình 3.6: Kết quả tính RSI	27
Hình 3.7: Kết quả tính Stochastic Oscillator %K	28
Hình 3.8: Kết quả tính Williams %R	29
Hình 3.9: Kết quả tính MACD_Signal	30
Hình 3.10: Kết quả tính OBV	31
Hình 3.11: Loại bỏ data (NaN)	31
Hình 3.12: Phân chia dữ liệu test training	32
Hình 3.13: Câu lệnh training, predict.....	32
Hình 3.14: Đoạn code tính độ chính xác	33
Hình 3.15: Kết quả dự đoán với bộ data test	34
Hình 3.16: Kết quả xác suất khả năng dự đoán với bộ data test.....	34

CHƯƠNG 1 – CÁC KIẾN THỨC CƠ BẢN

1.1 RANDOM FOREST

1.1.1 Giới Thiệu về RANDOM FOREST:

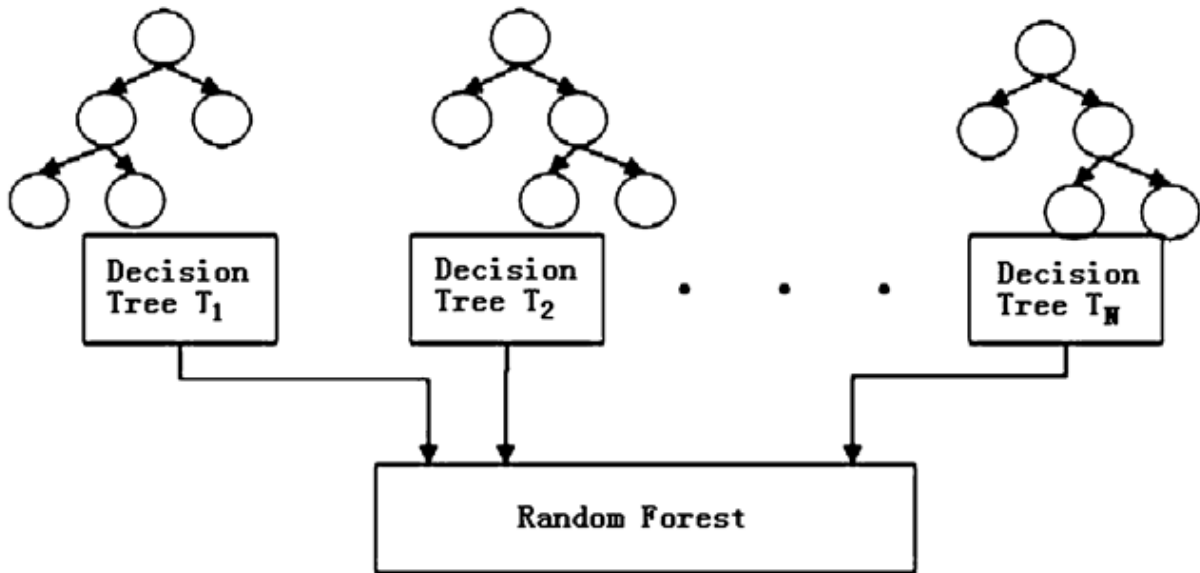
Phương phân lớp thuộc tính được phát triển bởi Leo Breiman tại đại học California, Berkeley là Random Forest (rừng ngẫu nhiên). Breiman cũng đồng thời là đồng tác giả của phương pháp CART (Classification and Regression Trees) được đánh giá là một trong 10 phương pháp khai phá dữ liệu kinh điển. Random Forest được xây dựng dựa trên 3 thành phần chính là: CART, học toàn bộ, hội đồng các chuyên gia, kết hợp các mô hình, và tổng hợp bootstrap (bagging).

Do phương pháp Statistical learning (tức Machine learning) ngày càng phổ biến trong nghiên cứu y học, nhu cầu diễn giải các mô hình Machine learning trở thành nhu cầu thiết yếu. Do đó, Nhi sẽ lần lượt chuyển đến các hướng dẫn sử dụng những packages mới này. Bài đầu tiên này sẽ là package “randomForestExplainer”, chuyên dụng cho mô hình Random Forest. Đây là một package vừa được công bố vào cuối tháng 7 năm 2017 bởi tác giả Aleksandra Paluszynska. Công dụng của package này cho phép khảo sát nội dung bên trong một mô hình Random Forest. Như chúng ta đã biết, Random Forest là một tập hợp mô hình (ensemble). Mô hình Random Forest rất hiệu quả cho các bài toán phân loại vì nó huy động cùng lúc hàng trăm mô hình nhỏ hơn bên trong với quy luật khác nhau để đưa ra quyết định cuối cùng. Mỗi mô hình con có thể mạnh yếu khác nhau, nhưng theo nguyên tắc « wisdom of the crowd », ta sẽ có cơ hội phân loại chính xác hơn so với khi sử dụng bất kỳ một mô hình đơn lẻ nào.

Như tên gọi của nó, Random Forest (RF) dựa trên cơ sở:

Random = Tính ngẫu nhiên

Forest = nhiều cây quyết định (decision tree).

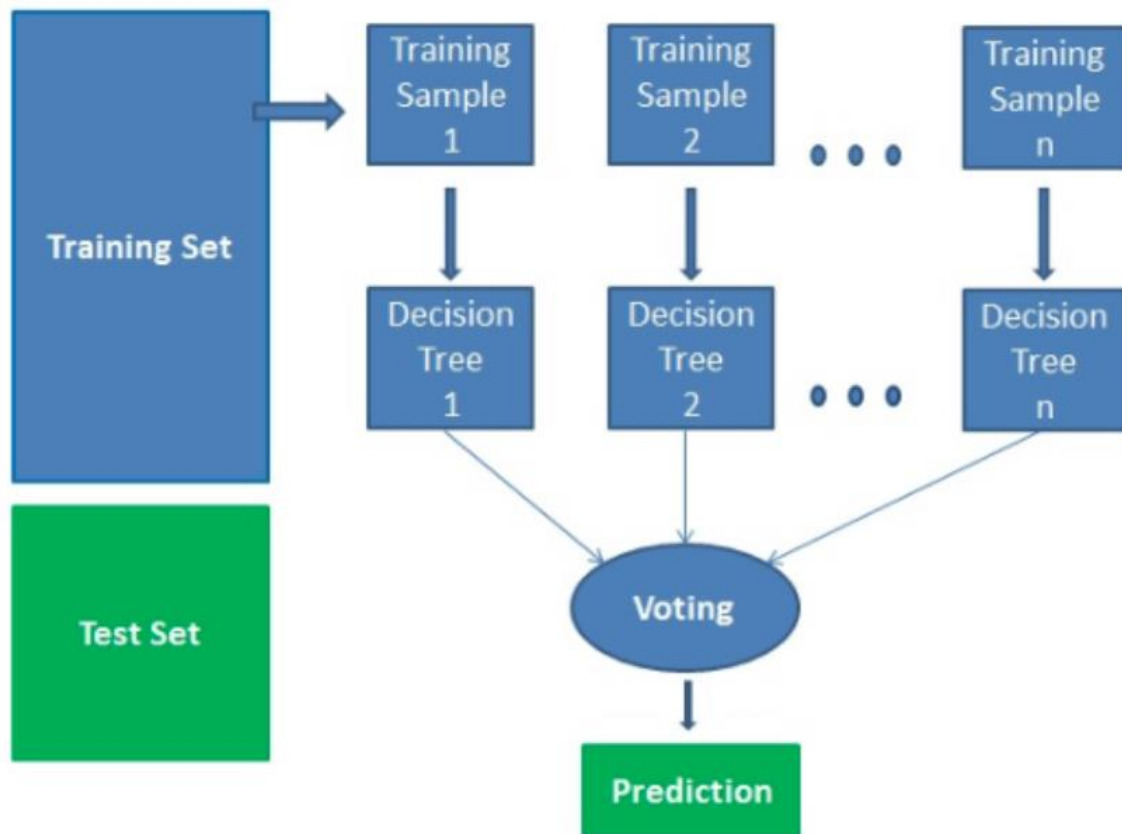


Hình 1.1: Random Forest

Đơn vị của RF là thuật toán cây quyết định, với số lượng hàng trăm. Mỗi cây quyết định được tạo ra một cách ngẫu nhiên từ việc: Tái chọn mẫu (bootstrap, random sampling) và chỉ dùng một phần nhỏ tập biến ngẫu nhiên (random features) từ toàn bộ các biến trong dữ liệu. Ở trạng thái sau cùng, mô hình RF thường hoạt động rất chính xác, nhưng đôi lại, ta không thể nào hiểu được cơ chế hoạt động bên trong mô hình vì cấu trúc quá phức tạp. RF do đó là một trong số những mô hình hộp đen (black box). Trong quá khứ, chúng ta thường chấp nhận đánh đổi tính tường minh để đạt được tính chính xác.

1.1.2 Hoạt Động Của Thuật Toán Random Forest

- Bước 1: Chọn các mẫu ngẫu nhiên từ tập dữ liệu đã cho.
- Bước 2: Thiết lập cây quyết định cho từng mẫu và nhận kết quả dự đoán từ mỗi quyết định cây.
- Bước 3: Hãy bỏ phiếu cho mỗi kết quả dự đoán.
- Bước 4: Chọn kết quả được dự đoán nhiều nhất là dự đoán cuối cùng.



Hình 1.2: Hoạt động của thuật toán Random Forest

1.1.3 Giải Thuật RF Cho Phân Lớp Được Diễn Giải Như Sau

- Lấy ra K mẫu bootstrap từ tập huấn luyện.
- Đối với mỗi mẫu bootstrap xây dựng một cây phân lớp không được tỉa (unpruned tree) theo hướng dẫn sau: Tại mỗi nút thay vì chọn một phân chia tốt nhất trong tất cả các biến dự đoán, ta chọn ngẫu nhiên một mẫu m của các biến dự đoán sau đó chọn một phân chia tốt nhất trong các biến này.
- Đưa ra các dự đoán bằng cách tổng hợp các dự đoán của K cây.

Quá trình học của Random Forest bao gồm việc sử dụng ngẫu nhiên giá trị đầu vào, hoặc kết hợp các giá trị đó tại mỗi node trong quá trình dựng từng cây quyết định. Kết quả của Random Forest, qua thực nghiệm cho thấy, là tốt hơn khi so sánh với thuật

toán Adaboost. Trong đó Random Forest có một số thuộc tính mạnh như:

- Độ chính xác của nó tương tự Adaboost, trong một số trường hợp còn tốt hơn.
- Thuật toán giải quyết tốt các bài toán có nhiều dữ liệu nhiễu.
- Thuật toán chạy nhanh hơn so với bagging hoặc boosting.
- Có những sự ước lượng nội tại như độ chính xác của mô hình phỏng đoán hoặc độ

mạnh và liên quan giữa các thuộc tính.

- Dễ dàng thực hiện song song.
- Tuy nhiên để đạt được các tính chất mạnh trên, thời gian thực thi của thuật toán khá lâu và phải sử dụng nhiều tài nguyên của hệ thống.

1.1.4 Đặc Điểm của Random Forest

OOB: Khi tập mẫu được rút ra từ một tập huấn luyện của một cây với sự thay thế (bagging), thì theo ước tính có khoảng $1/3$ các phần tử không có nằm trong mẫu này. Điều này có nghĩa là chỉ có khoảng $2/3$ các phần tử trong tập huấn luyện tham gia vào trong các tính toán của chúng ta, và $1/3$ các phần tử này được gọi là dữ liệu out-of-bag. Dữ liệu out-of-bag được sử dụng để ước lượng lỗi tạo ra từ việc kết hợp các kết quả từ các cây tổng hợp trong random forest cũng như dùng để ước tính độ quan trọng thuộc tính (variable important).

1.1.5 Ứng Dụng của Random Forest

Giả sử bạn muốn đi tham quan du lịch Anh và có sự cân nhắc cho việc tham quan thành phố nào như: Manchester, Liverpool hay Birmingham. Để trả lời câu hỏi này bạn sẽ cần tham khảo rất nhiều ý kiến từ bạn bè, blog du lịch, tour lữ hành ... Mỗi một ý kiến tương ứng với một Decision Tree trả lời các câu hỏi như: thành phố này đẹp không, có được tham quan các sân vận động khi đến thăm không, số tiền bỏ ra là bao nhiêu, thời gian để tham quan thành phố là bao lâu... Sau đó bạn sẽ có một rừng các câu trả lời để quyết định xem mình sẽ đi tham quan thành phố nào. Random Forest hoạt động bằng cách đánh giá các Decision Tree sử dụng cách thức voting để đưa ra

kết quả cuối cùng.

Ứng dụng trong dự đoán thiên nhiên, du lịch, thiên văn, y khoa, ngân hàng. Lấy ví dụ với ngành Ngân hàng, chúng ta có hai bài toán phổ biến cho Random Forest, là tìm kiếm khách hàng tiềm năng và khách hàng lừa đảo. Hoặc trong chính bài báo của chúng ta, đề tài mà chúng ta đang tìm hiểu “Predicting the direction of stock market prices using random forest” dự đoán hướng của giá thị trường chứng khoán sử dụng rừng ngẫu nhiên....



Hình 1.3: Các Ứng dụng của Random Forest

1.1.6 Ưu Điểm Và Nhược Điểm Của Random Forest

Ưu điểm:

Random forests được coi là một phương pháp chính xác và mạnh mẽ vì số cây quyết định tham gia vào quá trình này. Nó không bị vấn đề overfitting. Lý do chính là nó mất

trung bình của tất cả các dự đoán, trong đó hủy bỏ những thành kiến. Thuật toán có thể được sử dụng trong cả hai vấn đề phân loại và hồi quy. Random forests cũng có thể xử lý các giá trị còn thiếu. Có hai cách để xử lý các giá trị này: sử dụng các giá trị trung bình để thay thế các biến liên tục và tính toán mức trung bình gần kề của các giá trị bị thiếu. Có thể nhận được tầm quan trọng của tính năng tương đối, giúp chọn các tính năng đóng góp nhiều nhất cho trình phân loại.

Nhược Điểm:

Random forests chậm tạo dự đoán bởi vì nó có nhiều cây quyết định. Bất cứ khi nào nó đưa ra dự đoán, tất cả các cây trong rừng phải đưa ra dự đoán cho cùng một đầu vào cho trước và sau đó thực hiện bỏ phiếu trên đó. Toàn bộ quá trình này tốn thời gian. Mô hình khó hiểu hơn so với cây quyết định, nơi bạn có thể dễ dàng đưa ra quyết định bằng cách đi theo đường dẫn trong cây.

1.2 Tìm Hiểu Về Thị Trường Chứng Khoán:

Thị trường tài chính hay còn gọi là thị trường chứng khoán hoạt động theo những xu hướng thị trường nhất định và những xu hướng này được gọi là xu hướng thị trường chứng khoán. Những xu hướng này còn được hỗ trợ bởi phép phân tích kỹ thuật. Phép phân tích này được xem như là một ứng dụng khoa học chưa trưởng thành và bị phản đối rộng rãi bởi lý thuyết Efficient Market Hypothesis (tạm dịch là giả thuyết thị trường hiệu quả).

Phép phân tích kỹ thuật sẽ giúp tìm hiểu xem cổ phiếu trên thị trường chứng khoán đang nằm trong giai đoạn đầu cơ giá lên hay đầu cơ giá xuống và nó sẽ đề xuất kế hoạch giao dịch chiến lược để tận dụng lợi thế của từng giai đoạn. Phép phân tích cũng chỉ ra rằng thị trường luôn hoạt động theo chu kỳ lên xuống.

Lý thuyết Random Walk phản đối ý kiến cho rằng thị trường chứng khoán hoạt động theo xu hướng lên xuống nhất định. Lý thuyết này khẳng định rằng những thay đổi trong giá cổ phiếu có cùng một xu hướng giá trị và chúng không phụ thuộc vào nhau vì vậy không thể dựa vào những thay đổi trong quá khứ hay những xu hướng của thị trường

để dự đoán những thay đổi trong tương lai. Lý thuyết này cũng cho rằng các xu hướng của cổ phiếu là không thể đoán trước và nó biến động rất ngẫu nhiên.

Lý thuyết Efficient Market Hypothesis cho rằng không thể định hướng cho thị trường bởi vì sự hiệu quả của thị trường chứng khoán luôn hiển thị tất cả những thông tin thích hợp. Điều này có nghĩa là cổ phiếu luôn được giao dịch ở mức giá hợp lý trên sàn giao dịch và các nhà đầu tư không thể mua được cổ phiếu với giá dưới định mức hoặc bán cổ phiếu với giá vượt định mức.

Thị trường chứng khoán có thể có ba loại xu hướng thị trường:

- Xu hướng chính: Thị trường đầu cơ giá lên và thị trường đầu cơ giá xuống. Một thị trường đầu cơ giá lên cho thấy điều kiện nền kinh tế đang rất tốt, có rất nhiều nguồn việc làm, chỉ số GDP tăng và giá cổ phiếu đang tăng.

Thị trường đầu cơ giá lên sẽ làm gia tăng niềm tin nơi nhà đầu tư. Tuy nhiên thị trường đầu cơ giá lên không phải là một điều kiện lâu dài và thỉnh thoảng nó có thể dẫn đến những tình huống nguy hiểm nếu như những cổ phiếu được định giá quá cao.

Thị trường đầu cơ giá xuống cho thấy một nền kinh tế nghèo nàn, suy thoái có thể xuất hiện và giá cổ phiếu đang giảm từng ngày. Thị trường đầu cơ giá xuống luôn đi kèm với sự bi quan sâu rộng và các nhà đầu tư thì lo sợ việc thua lỗ.

- Xu hướng thứ cấp (ngắn hạn): Thị trường đầu cơ giá xuống và điều chỉnh. Thị trường thứ cấp là một sự thay đổi giá một cách vô thường và giai đoạn này kéo dài từ vài tuần đến vài tháng. Sự điều chỉnh thị trường bị ảnh hưởng bởi các yếu tố như thiên tai và bất ổn chính trị.
- Xu hướng trường kỳ (ngắn hạn): Thị trường đầu cơ giá lên lâu dài và thị trường đầu cơ giá xuống lâu dài. Xu hướng trường kỳ là một xu hướng có thể kéo dài từ 5 đến 20 năm. Trong xu hướng trường kỳ, thị trường đầu cơ giá lên nhỏ hơn so với thị trường đầu cơ giá xuống và không thể bù lỗ cho những khoản thua lỗ của thị trường đầu cơ giá xuống trước đó.

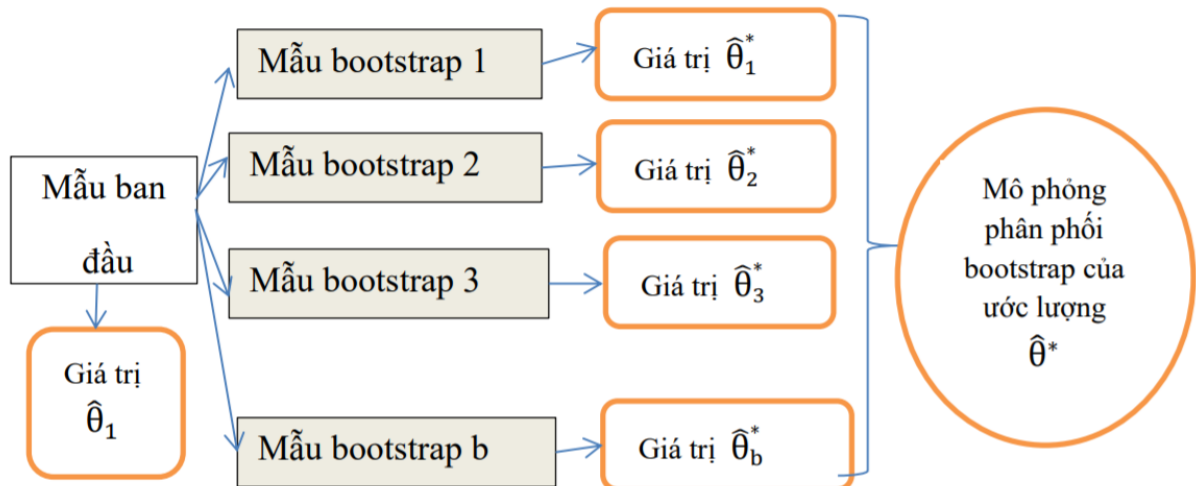
1.3 Bootstrap

1.3.1 Bootstrap là gì?

Bootstrap là một phương pháp rất nổi tiếng trong thống kê được giới thiệu bởi Bradley Efron vào năm 1979. Phương pháp này chủ yếu dùng để ước lượng lỗi chuẩn (standard errors), độ lệch (bias) và tính toán khoảng tin cậy (confidence interval) cho các tham số. Phương pháp này được thực hiện như sau: Từ một quần thể ban đầu lấy ra một mẫu $H = (x_1, x_2, \dots, x_n)$ gồm n thành phần, tính toán các tham số mong muốn. Trong các bước tiếp theo lặp lại b lần việc tạo ra mẫu H_b cũng gồm n phần tử từ H bằng cách lấy lại mẫu với sự thay thế các thành phần trong mẫu ban đầu sau đó tính toán các tham số mong muốn.

1.3.1 Nội Dung Phương Pháp Bootstrap

Phương pháp Bootstrap là phương pháp coi mẫu gốc ban đầu đóng vai trò tổng thể mà từ đó nó được rút ra. Từ mẫu ban đầu lấy lại các mẫu ngẫu nhiên cùng cỡ với mẫu gốc bằng phương pháp lấy mẫu có hoàn lại, gọi là mẫu bootstrap. Với mỗi mẫu lấy lại ta tính được giá trị tham số thống kê quan tâm gọi lại tham số bootstrap. Sự phân bố của các tham số thống kê mẫu bootstrap là phân phối bootstrap.



Hình 1.4: Sơ đồ mô phỏng phân phối bootstrap

1.4 Bagging

Phương pháp này được xem như là một phương pháp tổng hợp kết quả có được từ các bootstrap. Tư tưởng chính của phương pháp này như sau: Cho một tập huấn luyện $D = \{(x_i, y_i) : i=1, 2, \dots, n\}$ và giả sử chúng ta muốn có một dự đoán nào đó đối với biến x . Một mẫu gồm B tập dữ liệu, mỗi tập dữ liệu gồm n phần tử được chọn lựa ngẫu nhiên từ D với sự thay thế (giống như bootstrap). Do đó $B = (D_1, D_2, \dots, D_B)$ trông giống như là một tập các tập huấn luyện được nhân bản.

Tập huấn một máy hoặc một mô hình đối với mỗi tập D_b ($b=1, 2, \dots, B$) và lần lượt thu thập các kết quả dự báo có được trên mỗi tập D_b . Kết quả tổng hợp cuối cùng được tính toán bằng cách trung bình hóa (regression) hoặc thông qua số phiếu bầu nhiều nhất (classification).

1.5 Khai Phá Dữ Liệu

1.5.1 Khai Phá Dữ Liệu Là Gì ?

Khai phá dữ liệu là một tập hợp các kỹ thuật được sử dụng để tự động khai thác và tìm ra các mối quan hệ lẫn nhau của dữ liệu trong một tập hợp dữ liệu khổng lồ và phức tạp, đồng thời cũng tìm ra các mẫu tiềm ẩn trong tập dữ liệu đó.

Khai phá dữ liệu (datamining) được định nghĩa như là một quá trình chất lọc hay khai phá tri thức từ một lượng lớn dữ liệu. Một ví dụ hay được sử dụng là việc khai thác vàng từ đá và cát, Datamining được ví như công việc "Đãi cát tìm vàng" trong một tập hợp lớn các dữ liệu cho trước. Thuật ngữ Datamining ám chỉ việc tìm kiếm một tập hợp nhỏ có giá trị từ một số lượng lớn các dữ liệu thô. Có nhiều thuật ngữ hiện được dùng cũng có nghĩa tương tự với từ Datamining như Knowledge Mining (khai phá tri thức), knowledge extraction (chất lọc tri thức), data/pattern analysis (phân tích dữ liệu/mẫu), data archaeology (khảo cổ dữ liệu), data dredging (nạo vét dữ liệu),... Mục đích của bước này là nhận dạng các dòng của các hình ảnh bị nghiêng, giúp giảm sự mất thông tin khi nhận dạng ảnh nghiêng. Các bộ phận quan trọng của quá trình này là lọc dây màu (còn được gọi là blobs) và xây dựng dòng. Khai phá dữ liệu là một bước trong

bảy bước của quá trình KDD (Knowledge Discovery in Database) và KDD được xem như 7 quá trình khác nhau theo thứ tự sau:

1. Làm sạch dữ liệu (data cleaning & preprocessing): Loại bỏ nhiễu và các dữ liệu không cần thiết.

2. Tích hợp dữ liệu: (data integration): quá trình hợp nhất dữ liệu thành những kho dữ liệu (data warehouses & data marts) sau khi đã làm sạch và tiền xử lý (datacleaning & preprocessing).

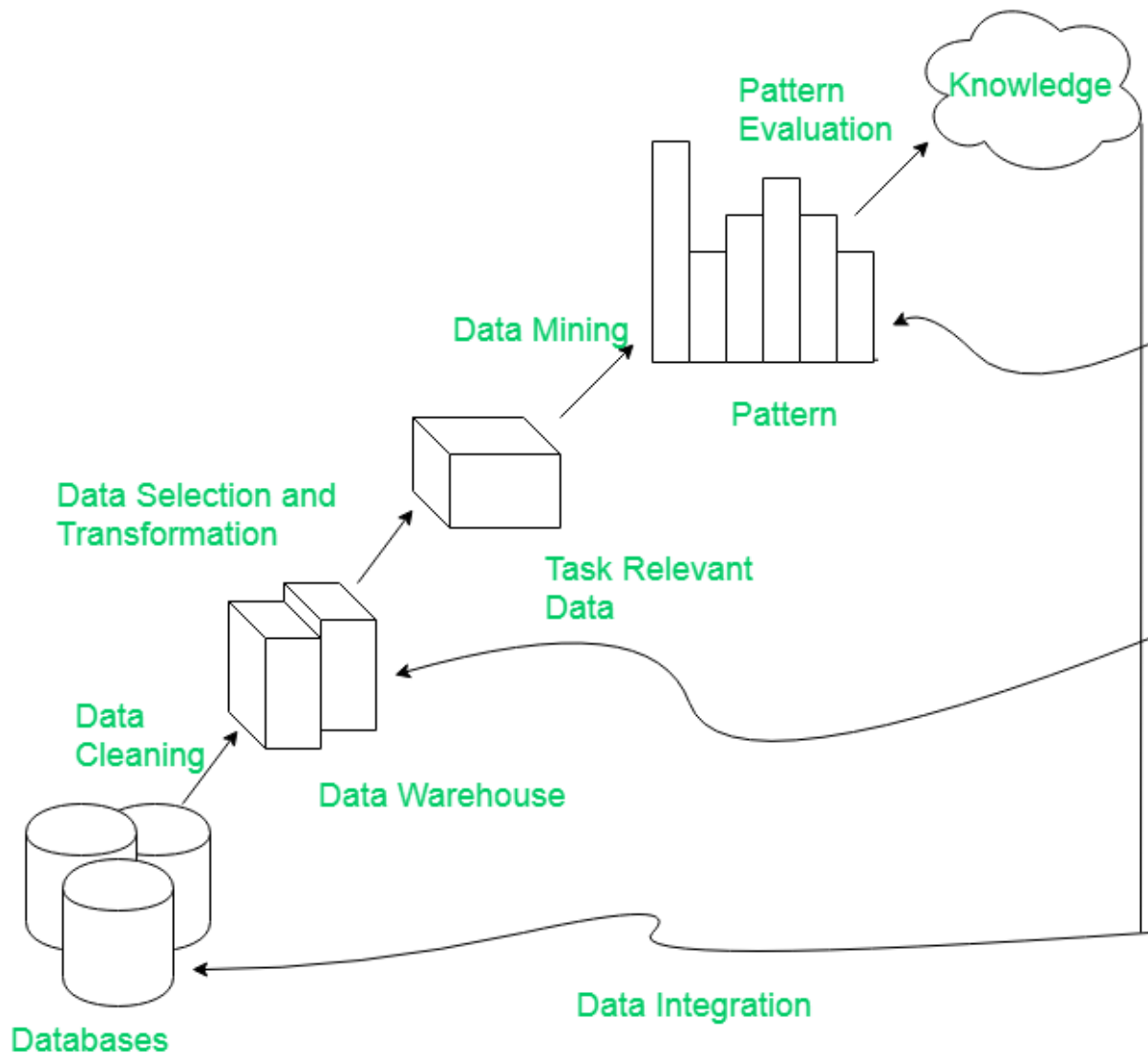
3. Trích chọn dữ liệu (data selection): trích chọn dữ liệu từ những kho dữ liệu và sau đó chuyển đổi về dạng thích hợp cho quá trình khai thác tri thức. Quá trình này bao gồm cả việc xử lý với dữ liệu nhiễu (noisy data), dữ liệu không đầy đủ (incomplete data), ...

4. Chuyển đổi dữ liệu: Các dữ liệu được chuyển đổi sang các dạng phù hợp cho quá trình xử lý

5. Khai phá dữ liệu(data mining): Là một trong các bước quan trọng nhất, trong đó sử dụng những phương pháp thông minh để chắt lọc ra những mẫu dữ liệu.

6. Ước lượng mẫu (knowledge evaluation): Quá trình đánh giá các kết quả tìm được thông qua các độ đo nào đó.

7. Biểu diễn tri thức (knowledge presentation): Quá trình này sử dụng các kỹ thuật để biểu diễn và thể hiện trực quan cho người dùng.



Hình 1.5: Các bước trong Data Mining & KDD

1.5.2 Chức Năng

Data Mining được chia nhỏ thành một số hướng chính như sau:

- Mô tả khái niệm (concept description): thiên về mô tả, tổng hợp và tóm tắt khái niệm. Ví dụ: tóm tắt văn bản.
- Luật kết hợp (association rules): là dạng luật biểu diễn tri thức ở dạng khá đơn giản. Luật kết hợp được ứng dụng nhiều trong lĩnh vực kinh doanh, y học, tin-sinh, tài chính & thị trường chứng khoán, ...

- Phân lớp và dự đoán (classification & prediction): xếp một đối tượng vào một trong những lớp đã biết trước. Ví dụ: phân lớp vùng địa lý theo dữ liệu thời tiết. Hướng tiếp cận này thường sử dụng một số kỹ thuật của machine learning như cây quyết định (decision tree), mạng nơ ron nhân tạo (neural network), Người ta còn gọi phân lớp là học có giám sát (học có thầy).

- Phân cụm (clustering): xếp các đối tượng theo từng cụm (số lượng cũng như tên của cụm chưa được biết trước. Người ta còn gọi phân cụm là học không giám sát (học không thầy).

- Khai phá chuỗi (sequential/temporal patterns): tương tự như khai phá luật kết hợp nhưng có thêm tính thứ tự và tính thời gian. Hướng tiếp cận này được ứng dụng nhiều trong lĩnh vực tài chính và thị trường chứng khoán vì nó có tính dự báo cao.

CHƯƠNG 2: XÂY DỰNG PHẦN MỀM

1.1 Chuẩn Bị:

Hệ Điều Hành: Window

Python: version 3.5.1

Package required:

1. Pandas : sử dụng Pandas làm nền tảng Data Frame trong việc xử lý
2. Numpy : tạo ra các giá trị ngẫu nhiên
3. Sklearn: sử dụng cho việc áp dụng Random Forests
4. Matplotlib: sử dụng cho việc plot các hình ảnh kết quả data

Data: Sử dụng nguồn dữ liệu từ các trang web uy tín như NYSE hay NASDAQ trong phần demo chúng tôi sử dụng mẫu data của NASDAQ

Results for: 1 Month, From 07-MAY-2019 TO 07-JUN-2019					
Date	Open	High	Low	Close / Last	Volume
06/07/2019	186.51	191.92	185.77	190.15	30,684,390
06/06/2019	183.08	185.47	182.1489	185.22	22,526,310
06/05/2019	184.28	184.99	181.14	182.54	29,773,430
06/04/2019	175.44	179.83	174.52	179.64	30,967,960
06/03/2019	175.6	177.92	170.27	173.3	40,396,070
05/31/2019	176.23	177.99	174.99	175.07	27,043,580
05/30/2019	177.95	179.23	176.67	178.3	21,218,410
05/29/2019	176.42	179.35	176	177.38	28,481,170
05/28/2019	178.92	180.59	177.91	178.23	27,948,160
05/24/2019	180.2	182.14	178.62	178.97	23,714,690
05/23/2019	179.8	180.54	177.81	179.66	36,529,740
05/22/2019	184.66	185.71	182.55	182.78	29,748,560
05/21/2019	185.22	188	184.7	186.6	28,364,850
05/20/2019	183.52	184.349	180.2839	183.09	38,612,290

Hình 2.1: Mẫu data từ trang web NASDAQ

1.2 Cài Đặt:

1.2.1 Cài Đặt Python:

Download Python, bạn truy cập địa chỉ: <https://www.python.org/downloads/> chọn phiên bản bạn cần, trong bài viết này chúng tôi chọn Python 3.5.1. Nhấp đúp vào file vừa tải về để cài đặt. Tại đây có 2 tùy chọn, bạn chọn một cái để cài.

Install Now: Mặc định cài Python vào ổ C, cài sẵn IDLE (cung cấp giao diện đồ họa để làm việc với Python), pip và tài liệu, tạo shortcut,... khi quá trình cài đặt hoàn tất ta có thể chạy Python.

1.2.2 Cài Đặt Package Pandas:

Thực hiện cài đặt Anaconda.

Bật cửa sổ terminal và chạy các lệnh sau:

<pre>conda create -n name_of_my_env python source activate name_of_my_env activate name_of_my_env</pre>
<pre>conda install pandas conda install pandas=0.20.3 conda install ipython conda install anaconda conda install pip pip install django</pre>

Bảng 2.1 chạy Pandas và các Package hỗ trợ

1.2.3 Cài Đặt Numpy

Bật cửa sổ terminal và chạy các lệnh sau:

<pre>conda install numpy</pre>

hoặc “conda install -c anaconda numpy”

Bảng 2.2 chạy Package Numpy

1.2.4 Cài Đặt Random Forest

Thực hiện chạy “pip install ta” trên cửa sổ terminal.

Thêm tất cả các tính năng

```
import pandas as pd
from ta import *

# Load datas
df = pd.read_csv('your-file.csv', sep=',')

# Clean NaN values
df = utils.dropna(df)

# Add ta features filling NaN values
df = add_all_ta_features(df, "Open", "High", "Low", "Close", "Volume_BTC",
fillna=True)
```

Bảng 2.3: Thêm tất cả các tính năng

Ví dụ thêm các tính năng riêng lẻ:

```
import pandas as pd
from ta import *

# Load datas
df = pd.read_csv('your-file.csv', sep=',')
```

```
# Clean NaN values
df = utils.dropna(df)

# Add bollinger band high indicator filling NaN values
df['bb_high_indicator'] = bollinger_hband_indicator(df["Close"], n=20,
ndev=2, fillna=True)

# Add bollinger band low indicator filling NaN values
df['bb_low_indicator'] = bollinger_lband_indicator(df["Close"], n=20, ndev=2,
fillna=True)
```

Bảng 2.4: Thêm các tính năng riêng lẻ

Thực hiện chạy các lệnh:

```
$ git clone https://github.com/bukosabino/ta.git
$ cd ta
$ pip install -r dev-terms.txt
$ cd dev
$ python bollinger_band_features_example.py
```

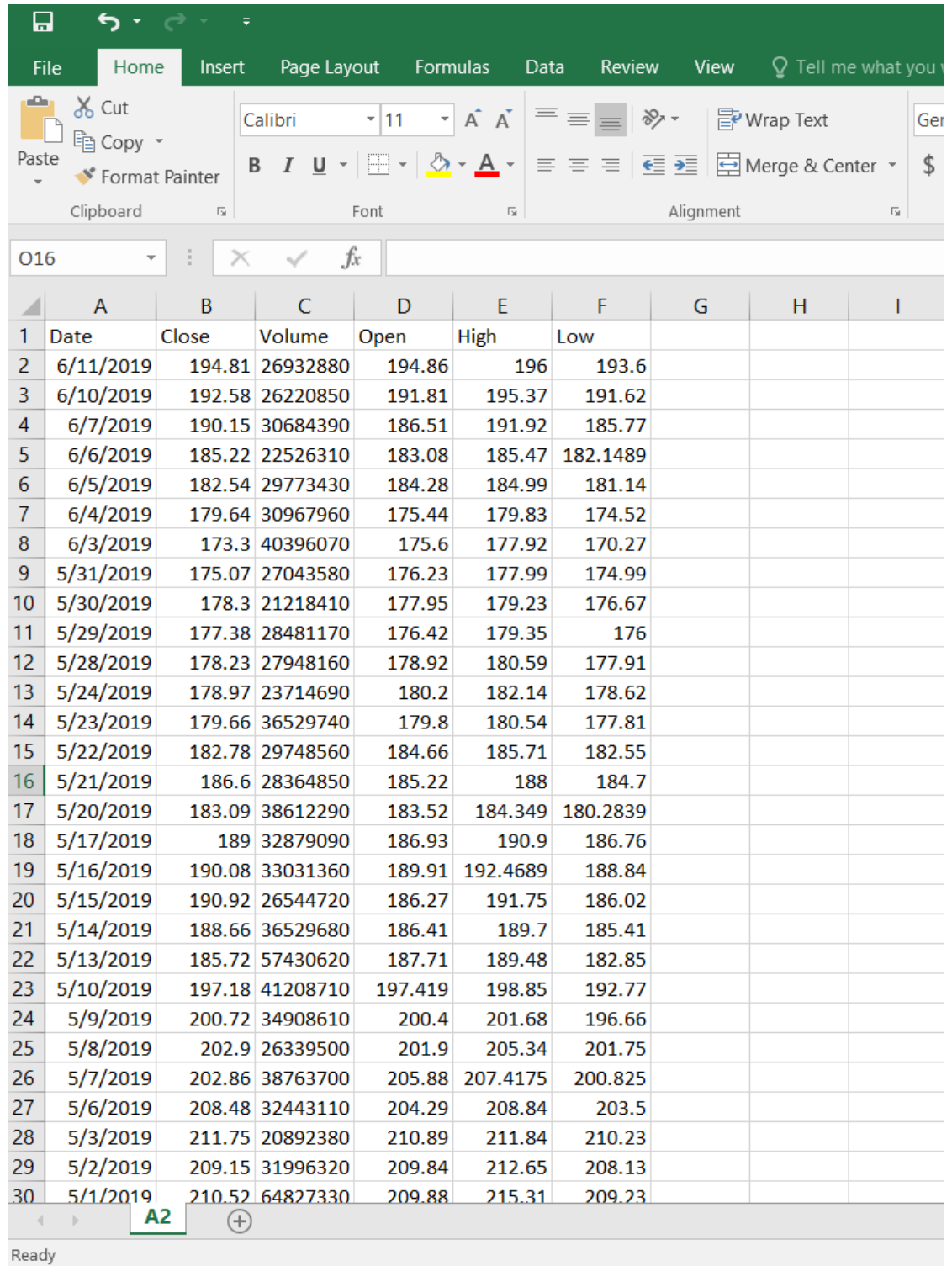
Bảng 2.5: Lệnh chạy

CHƯƠNG 3 – DEMO VÀ ĐÁNH GIÁ, NHẬN XÉT PHẦN MỀM ỨNG DỤNG

1.1 Chạy Demo

- Điều kiện thử nghiệm bị giới hạn và nguồn data không nhiều
- Do thời gian làm đồ án có hạn và quy mô hệ thống khá lớn nên em chỉ thực hiện thử nghiệm với một số giới hạn sau:
- Chương trình là một phần mềm ứng dụng dùng để dự đoán hướng của giá thị trường chứng khoán sử dụng rừng ngẫu nhiên.

Input data đầu vào: là Data Apple (APPL) từ 11/6/2018 đến 11/6/2019 được download tại trang NASDAQ lưu dưới định dạng file (.csv). Gồm các column cần thiết phục vụ cho việc rút trích các đặc trưng là Close, Open, High, Volume



	A	B	C	D	E	F	G	H	I
1	Date	Close	Volume	Open	High	Low			
2	6/11/2019	194.81	26932880	194.86	196	193.6			
3	6/10/2019	192.58	26220850	191.81	195.37	191.62			
4	6/7/2019	190.15	30684390	186.51	191.92	185.77			
5	6/6/2019	185.22	22526310	183.08	185.47	182.1489			
6	6/5/2019	182.54	29773430	184.28	184.99	181.14			
7	6/4/2019	179.64	30967960	175.44	179.83	174.52			
8	6/3/2019	173.3	40396070	175.6	177.92	170.27			
9	5/31/2019	175.07	27043580	176.23	177.99	174.99			
10	5/30/2019	178.3	21218410	177.95	179.23	176.67			
11	5/29/2019	177.38	28481170	176.42	179.35	176			
12	5/28/2019	178.23	27948160	178.92	180.59	177.91			
13	5/24/2019	178.97	23714690	180.2	182.14	178.62			
14	5/23/2019	179.66	36529740	179.8	180.54	177.81			
15	5/22/2019	182.78	29748560	184.66	185.71	182.55			
16	5/21/2019	186.6	28364850	185.22	188	184.7			
17	5/20/2019	183.09	38612290	183.52	184.349	180.2839			
18	5/17/2019	189	32879090	186.93	190.9	186.76			
19	5/16/2019	190.08	33031360	189.91	192.4689	188.84			
20	5/15/2019	190.92	26544720	186.27	191.75	186.02			
21	5/14/2019	188.66	36529680	186.41	189.7	185.41			
22	5/13/2019	185.72	57430620	187.71	189.48	182.85			
23	5/10/2019	197.18	41208710	197.419	198.85	192.77			
24	5/9/2019	200.72	34908610	200.4	201.68	196.66			
25	5/8/2019	202.9	26339500	201.9	205.34	201.75			
26	5/7/2019	202.86	38763700	205.88	207.4175	200.825			
27	5/6/2019	208.48	32443110	204.29	208.84	203.5			
28	5/3/2019	211.75	20892380	210.89	211.84	210.23			
29	5/2/2019	209.15	31996320	209.84	212.65	208.13			
30	5/1/2019	210.52	64827330	209.88	215.31	209.23			

Hình 3.1: Input data đầu vào

Input đầu vào chúng ta có column Date không cần sử dụng trong dự đoán có thể xóa đi qua câu lệnh "Del". Chúng ta cần exponential smoothing (làm mịn theo cấp số nhân) qua công thức dưới đây:

$$S_0 = Y_0$$

$$\text{For } t > 0, S_t = \alpha * Y_t + (1 - \alpha) * S_{t-1}$$

Hình 3.2: Công thức Exponential Smoothing

Giá trị α nằm trong khoảng $0 < \alpha < 1$. Ở đây tôi chọn $\alpha = 0.9$ và thực hiện câu lệnh code dưới đây

```
edata = df.ewm(alpha=0.9).mean()
```

Trong đó
df là Data Frame input đầu vào dữ liệu ban đầu

Hình 3.3: Câu lệnh Exponential Smoothing

Kết quả xuất ra sau khi Smoothing.

	A	B	C	D	E	F	G
1	Close	Volume	Open	High	Low		
2	194.81	26932880	194.86	196	193.6		
3	192.7827273	26285580	192.0873	195.4273	191.8		
4	190.4109009	30248472	187.0627	192.2676	186.3676		
5	185.7386229	23297831	183.4779	186.1491	182.5704		
6	182.8598335	29125928	184.1998	185.1059	181.283		
7	179.9619805	30783758	176.316	180.3576	175.1963		
8	173.9661974	39434840	175.6716	178.1638	170.7626		
9	174.9596198	28282706	176.1742	178.0074	174.5673		
10	177.965962	21924840	177.7724	179.1077	176.4597		
11	177.4385962	27825537	176.5552	179.3258	176.046		
12	178.1508596	27935898	178.6835	180.4636	177.7236		
13	178.888086	24136811	180.0484	181.9724	178.5304		
14	179.5828086	35290447	179.8248	180.6832	177.882		
15	182.4602809	30302749	184.1765	185.2073	182.0832		
16	186.1860281	28558640	185.1156	187.7207	184.4383		
17	183.3996028	37606925	183.6796	184.6862	180.6993		
18	188.4399603	33351873	186.605	190.2786	186.1539		
19	189.915996	33063411	189.5795	192.2499	188.5714		
20	190.8195996	27196589	186.6009	191.8	186.2751		
21	188.87596	35596371	186.4291	189.91	185.4965		
22	186.035596	55247195	187.5819	189.523	183.1147		
23	196.0655596	42612559	196.4353	197.9173	191.8045		

Hình 3.4: Kết quả data sau khi Exponential Smoothing

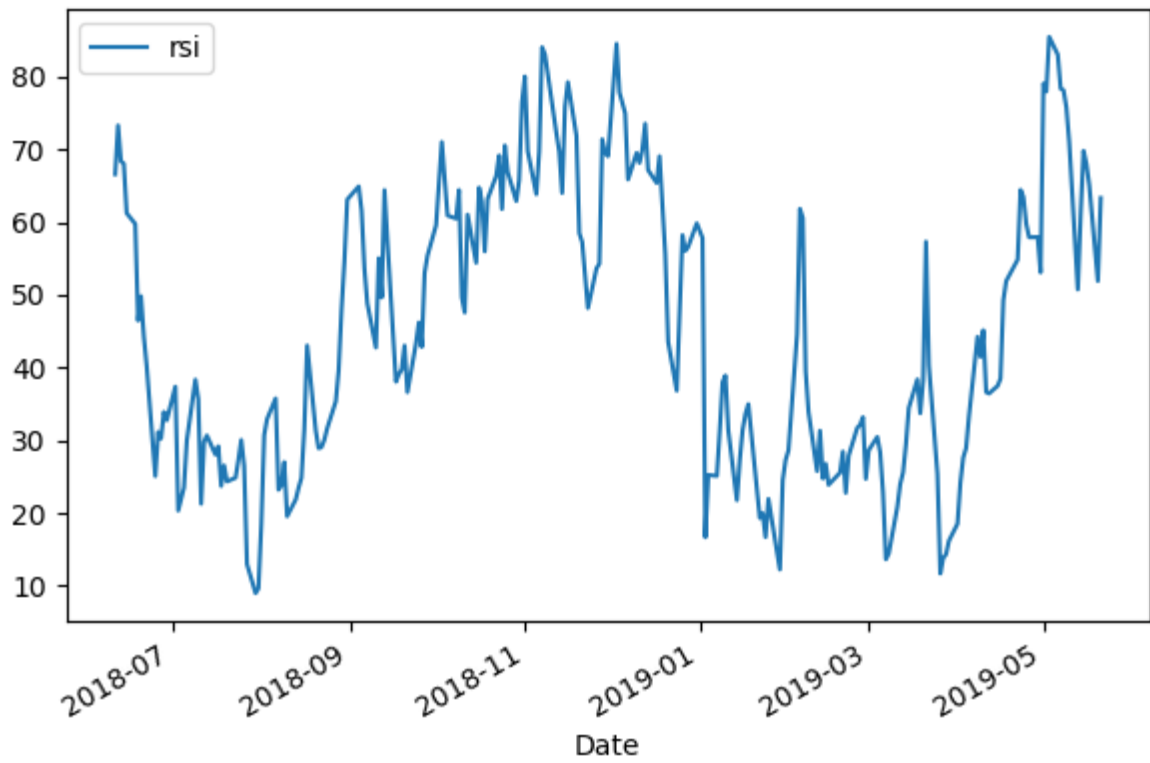
Thực hiện rút trích đặc trưng từ data input sau khi smoothing. Chúng ta cần có các đặc trưng Relative Strength Index (RSI), Stochastic Oscillator %K, Williams %R, Moving Average Convergence Divergence (MACD Signal), Price Rate of Change với tham số đầu vào “n” đây là [5, 14, 26, 44, 66] , On Balance Volume . Trong phần demo Chúng tôi sử dụng folder “ta” chứa các file python có các function sử dụng trong việc rút trích đặc trưng.

```
RSI14 = momentum.rsi(saapl['Close'])
STOCH = momentum.stoch(saapl['High'],saapl['Low'],saapl['Close'])
WR = momentum.wr(saapl['High'],saapl['Low'],saapl['Close'])
MACD = trend.macd_signal(saapl['Close'])
OBV = volume.on_balance_volume(saapl['Close'],saapl['Volume'])
for x in [5, 14, 26, 44, 66]:
    saapl = rate_of_change(saapl, n=x)
```

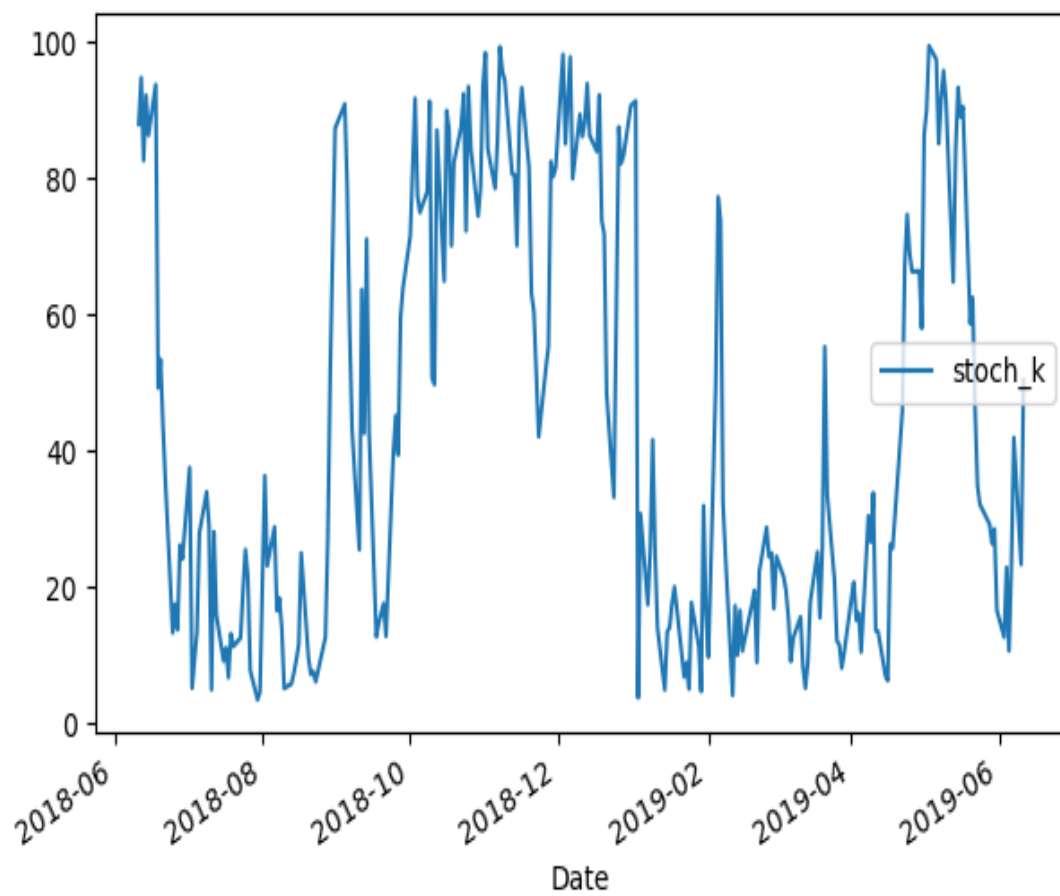
Trong đó
saapl là Data Frame input đầu vào dữ liệu đã Exponential Smoothing

Hình 3.5: Câu lệnh thực hiện các tính toán các đặc trưng

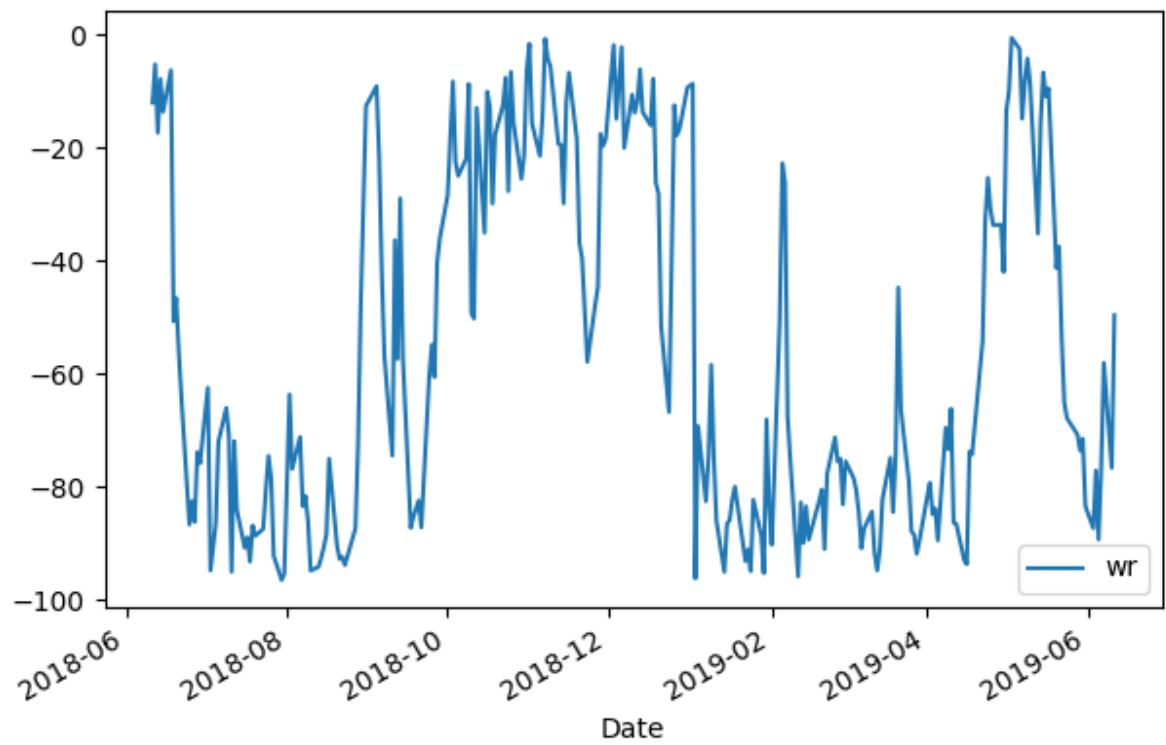
Dưới đây là hình ảnh kết quả tính toán của các đặc trưng



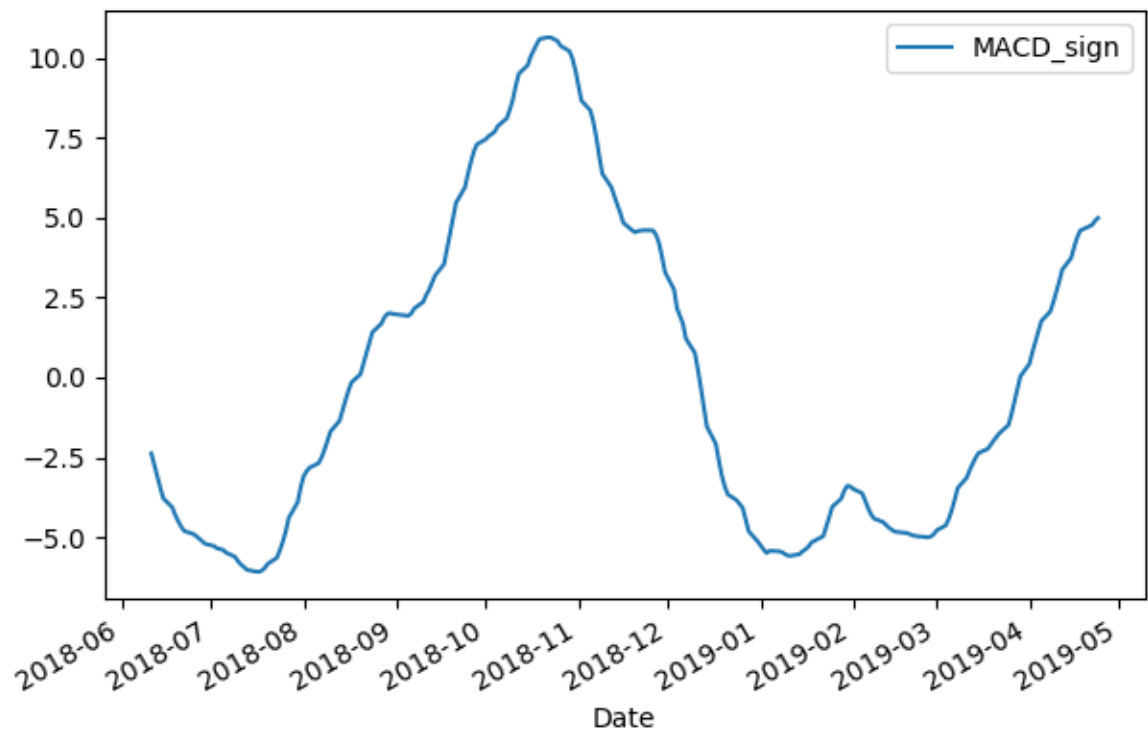
Hình 3.6: Kết quả tính RSI



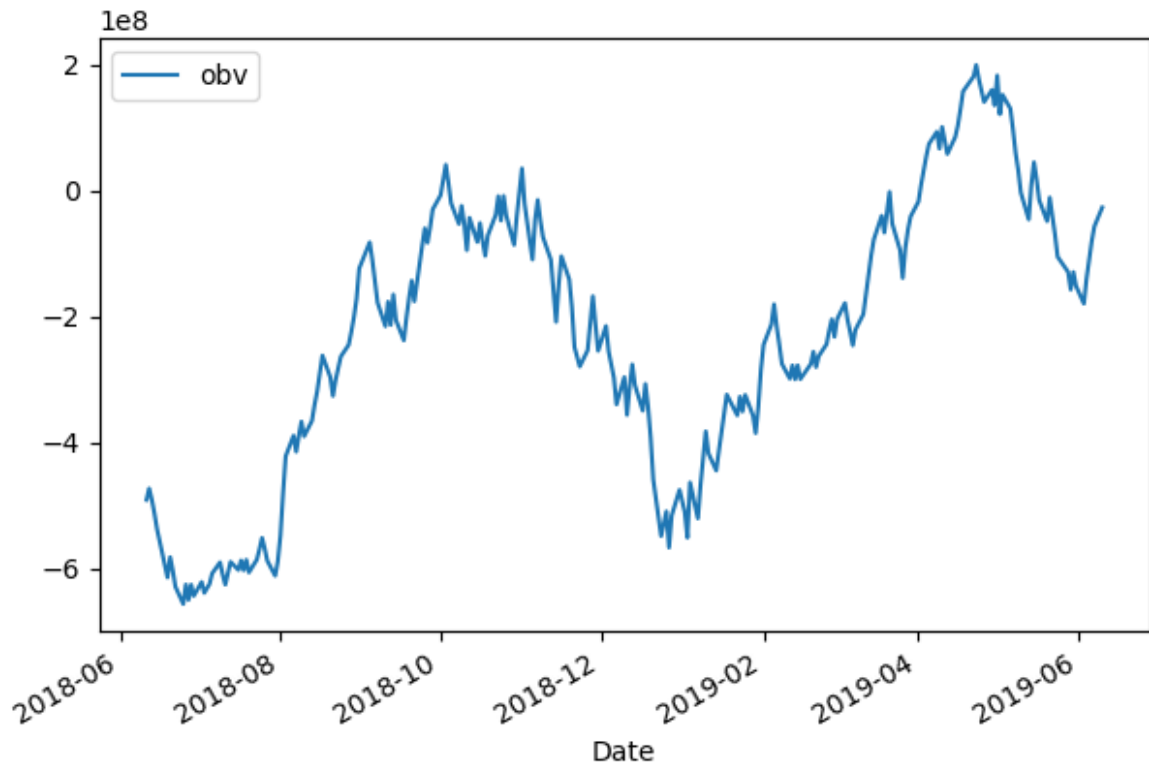
Hình 3.7: Kết quả tính Stochastic Oscillator %K



Hình 3.8: Kết quả tính Williams %R



Hình 3.9: Kết quả tính MACD_Signal



Hình 3.10: Kết quả tính OBV

Sau khi chúng ta trích đặc trưng xong sẽ có các dữ liệu trống (NaN) không sử dụng được trong việc training data cho Random Forest, loại bỏ dữ liệu trống bằng cách

```
saapl = saapl.dropna().
```

Trong đó

saapl là Data Frame input đầu vào dữ liệu đã Exponential Smoothing

Hình 3.11: Loại bỏ data (NaN)

Ở đây chúng tôi phân chia dữ liệu thành ba phần: hai phần đầu sử dụng cho việc training phần còn lại sử dụng cho data test thông qua câu lệnh `train_test_split` từ thư viện `sklearn.model_selection`.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size = 2*len(saapl) // 3)
```

Trong đó

`saapl` là Data Frame input đầu vào dữ liệu đã Exponential Smoothing

`X_train, y_train` là data bộ dữ liệu cho việc train Random Forest.

`X_test, y_test` là data bộ dữ liệu cho việc test phỏng đoán.

`X` là input đặc trưng đầu vào input.

`Y` là output kết quả thực tế, `Y` sẽ so sánh output của `X` sinh ra.

Hình 3.12: Phân chia dữ liệu test training

Bây giờ chúng ta sử dụng giải thuật Random Forest từ thư viện `sklearn.ensemble` chọn các tham số đầu vào `number of estimators` là 65 và số lần `random_state` là 42 (Hai thông số này có ảnh hưởng tới kết quả training) .

Sau đó sử dụng câu lệnh "fit" tiến hành training và câu lệnh "predict" tiến hành dự đoán trên tập data set cho test

```
rf = RandomForestClassifier(n_jobs=-1, n_estimators=65, random_state=42)
```

```
rf.fit(X_train, y_train.values.ravel());
```

```
pred = rf.predict(X_test)
```

Hình 3.13: Câu lệnh training, predict

Sau khi hoàn thành việc dự đoán, ta mô phỏng hình ảnh của việc dự đoán qua đoạn code dưới đây:

```
precision = precision_score(y_pred=pred, y_true=y_test)
recall = recall_score(y_pred=pred, y_true=y_test)
f1 = f1_score(y_pred=pred, y_true=y_test)
accuracy = accuracy_score(y_pred=pred, y_true=y_test)
```

Hình 3.14: Đoạn code tính độ chính xác

Kết quả tính thu được:

⇒ Precision: 0.94

⇒ Recall: 1

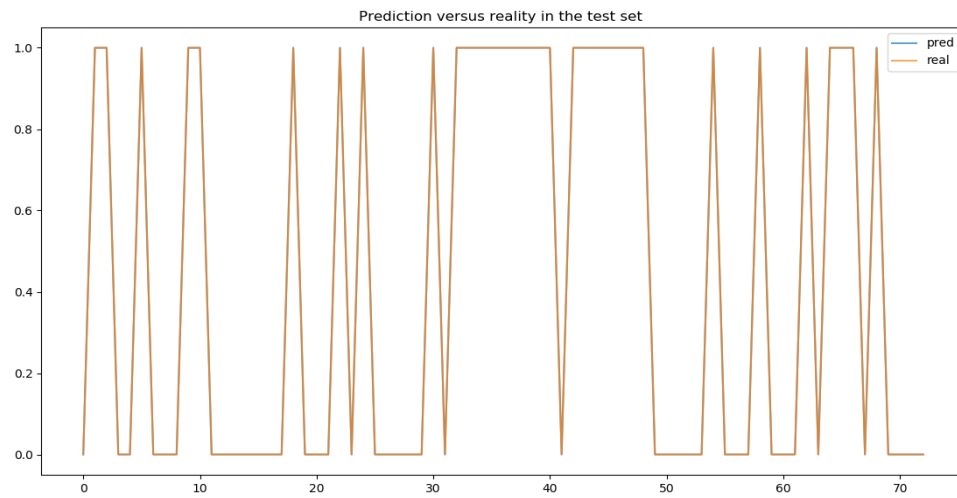
⇒ F1: 0.97

⇒ Accuracy: 0.97

(Kết quả này chưa xác thực vì số lượng đặc trưng sử dụng cho data còn ít, có khả năng cao bị Overfitting – Kết quả dự đoán luôn trùng khớp với dữ liệu cũ nhưng dữ liệu mới thì lại không phù hợp.)

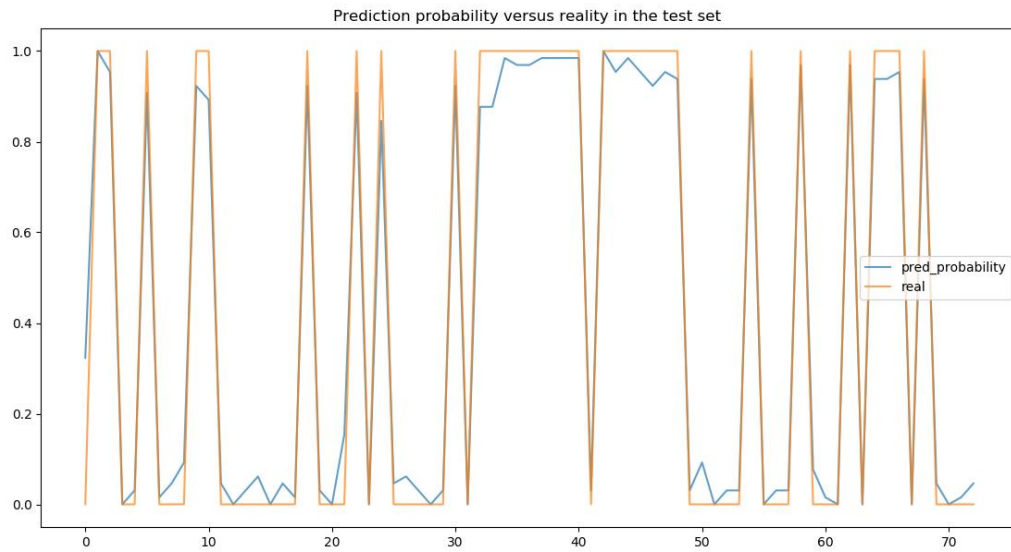
Dưới đây là một số hình ảnh đồ thị mô phỏng lại dự đoán sau khi training Random Forest với dữ liệu thực tế:

Figure 1



Hình 3.15: Kết quả dự đoán với bộ data test

Figure 3



Hình 3.16: Kết quả xác suất khả năng dự đoán với bộ data test

1.2 Đánh Giá và Nhận Xét:

Phần mềm hoạt động khá tốt tuy nhiên phần mềm ứng dụng dùng để dự đoán hướng của giá thị trường chứng khoán sử dụng rừng ngẫu nhiên này vẫn còn trong giai đoạn khởi đầu, tính năng và chất lượng dự đoán chỉ ở mức trung bình. Đối với một số nội dung dự báo lớn khác đòi hỏi phải có một sự chuẩn bị chu đáo logic thật tốt hơn nữa về dữ liệu, trang thiết bị, cũng như về mọi mặt phải thật chuẩn sát.

Vì thời gian triển khai nghiên cứu có hạn và việc tìm hiểu công nghệ mới còn gặp nhiều khó khăn do không có nhiều tài liệu nên không tránh được những sai sót. Em rất mong nhận được sự đóng góp ý kiến và hướng dẫn của thầy cô để đề tài của em thêm hoàn thiện.

Với kết quả thực nghiệm trên, hướng nghiên cứu phát triển tiếp theo của nhóm em sẽ là: Nâng cao hiệu quả hơn nữa về phần mềm ứng dụng dùng để dự đoán hướng của giá thị trường chứng khoán sử dụng rừng ngẫu nhiên và lần sang nhiều lĩnh vực khác về y khoa, thiên văn, khí tượng thủy văn, ngân hàng và còn nhiều hơn nữa trong tương lai.

TÀI LIỆU THAM KHẢO

❖ Tài liệu:

https://en.wikipedia.org/wiki/Main_Page

https://vi.wikipedia.org/wiki/Khai_ph%C3%A1_d%E1%BB%AF_li%E1%BB%87u

<https://www.geeksforgeeks.org/data-mining-kdd-process/>

❖ Hướng Dẫn Cài Đặt Và Cấu Hình:

<https://technical-analysis-library-in-python.readthedocs.io/en/latest/>

<https://www.quantopian.com/posts/technical-analysis-indicators-without-talib-code?fbclid=IwAR17dLC1DRFPFQNndMFQIDXO-ONXtpD7yJKyi-FRhVzComw-ROqs7kWEc6s>

[ROqs7kWEc6s](https://www.quantopian.com/posts/technical-analysis-indicators-without-talib-code?fbclid=IwAR17dLC1DRFPFQNndMFQIDXO-ONXtpD7yJKyi-FRhVzComw-ROqs7kWEc6s)

<https://github.com/jmartinezheras/reproduce-stock-market-direction-random-forests?fbclid=IwAR1o37ck8L1eLcqnPmJVJU3Tg-Me5S94TqgGgfefgMBPP2iB-XN-VrgENAE>

[forests?fbclid=IwAR1o37ck8L1eLcqnPmJVJU3Tg-Me5S94TqgGgfefgMBPP2iB-XN-VrgENAE](https://github.com/jmartinezheras/reproduce-stock-market-direction-random-forests?fbclid=IwAR1o37ck8L1eLcqnPmJVJU3Tg-Me5S94TqgGgfefgMBPP2iB-XN-VrgENAE)

[XN-VrgENAE](https://github.com/jmartinezheras/reproduce-stock-market-direction-random-forests?fbclid=IwAR1o37ck8L1eLcqnPmJVJU3Tg-Me5S94TqgGgfefgMBPP2iB-XN-VrgENAE)

[https://github.com/bukosabino/ta?fbclid=IwAR0fNa-](https://github.com/bukosabino/ta?fbclid=IwAR0fNa-plaLcmaTQ8oCgKJkK3rJLpjej3xZkHzy-katKtu_-IlrHi8nrJx4pIaLcmaTQ8oCgKJkK3rJLpjej3xZkHzy-katKtu_-IlrHi8nrJx4)

[plaLcmaTQ8oCgKJkK3rJLpjej3xZkHzy-katKtu_-IlrHi8nrJx4](https://github.com/bukosabino/ta?fbclid=IwAR0fNa-plaLcmaTQ8oCgKJkK3rJLpjej3xZkHzy-katKtu_-IlrHi8nrJx4pIaLcmaTQ8oCgKJkK3rJLpjej3xZkHzy-katKtu_-IlrHi8nrJx4)