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

ĐỒ ÁN CUỐI KÌ

MÔN PHÂN TÍCH XÁC SUẤT VÀ GIẢI THUẬT NGẪU NHIÊN

PREDICTING THE DISCRETION OF STOCK MARKET PRICES USING RANDOM FOREST

Người hướng dẫn: **TS. NGUYỄN CHÍ THIỆN**

Người thực hiện: **DUNG CẨM QUANG – 186005002**

TAYMANY BOUNNHALITH – 176005003

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2019

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

ĐỒ ÁN CUỐI KÌ

MÔN PHÂN TÍCH XÁC SUẤT VÀ GIẢI THUẬT NGẪU NHIÊN

PREDICTING THE DISCRETION OF STOCK MARKET PRICES USING RANDOM FOREST

Người hướng dẫn: **TS. NGUYỄN CHÍ THIỆN**

Người thực hiện: **DUNG CẨM QUANG – 186005002**

TAYMANY BOUNNHALITH – 176005003

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2019

LỜI CẢM ƠN

Nhóm xin được gửi lời cảm ơn chân thành đến Thầy Nguyễn Chí Thiện – giảng viên môn học. Thầy đã cung cấp những kiến thức về xác suất, giải thuật ngẫu nhiên để chúng tôi có thể hoàn thành được bài báo cáo này. Thầy đã tận tình giải đáp thắc mắc cũng như cho chúng tôi những lời nhận xét về bài làm để chúng tôi có thể hoàn thiện được bài báo cáo này.

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

Tôi xin cam đoan đây là sản phẩm 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)

Dung Cẩm Quang

Taymany Bounnhalith

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

Bài báo cáo sẽ tìm hiểu, giải thích về bài báo Predicting the direction of stock market prices using random forest của tác giả Luckyson Khaidem và cộng sự xuất bản năm 2016. Bài báo này nói về việc xây dựng hệ thống dự đoán xu hướng của chứng khoán tăng hay giảm dựa vào bộ dữ liệu chứng khoán với 06 đặc trưng: chỉ số RSI, chỉ số Stochastic Oscillator (%K), chỉ số Williams %R, đường Moving Average Convergence Divergence (MACD), chỉ số Price Rate of Change, chỉ số On Balance Volume. Các đặc trưng này sẽ được sử dụng vào mô hình dự đoán bằng Random Forest. Random Forest là giải thuật học máy dựa trên ý tưởng từ cây quyết định (Decision Tree). Thông qua giải thuật này tác giả của bài báo sẽ so sánh với các phương pháp trước đây của các tác giả khác trong cùng đề tài dự đoán chứng khoán. Bài báo cáo cũng sẽ phân tích demo mô phỏng giải thuật của tác giả.

MỤC LỤC

DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

CÁC CHỮ VIẾT TẮT

RF	Random Forest – Rừng ngẫu nhiên
OOB	Out of bag

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

DANH MỤC HÌNH

DANH MỤC BẢNG

CHƯƠNG 1 – TỔNG QUAN

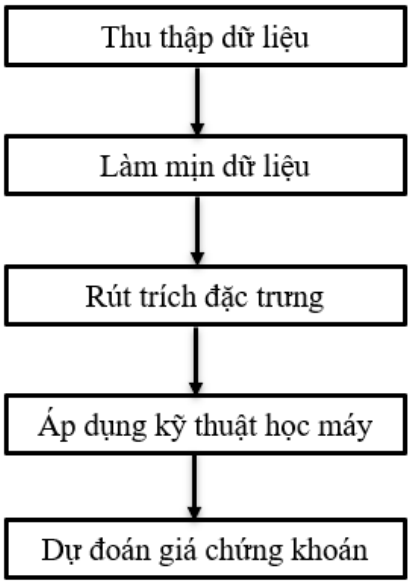
1.1 Giới thiệu bài toán

Dự đoán chứng khoán là bài toán xử lý dữ liệu trên chuỗi thời gian (time series). Giá của chứng khoán sẽ thay đổi theo thời gian phụ thuộc vào nhiều yếu tố khác nhau. Bài toán dự đoán chứng khoán bao gồm nhiều bài toán con như: dự đoán giá cổ phiếu trong một khoản thời gian xác định ở tương lại, dự đoán xu hướng tăng hay giảm của một loại chứng khoán trong thời gian tương lai, dự đoán các đường chỉ báo của chứng khoán, … Và nhiều phương pháp đã được áp dụng cho các bài toán này như: lý thuyết thống kê Efficient Market Hypothesis (Fama and Malkiel (1970)), các giải thuật học máy: Support Vector Machine (SVM), Linear Discriminant Analysis, Linear Regression, KNN and Naive Bayesian Classifier, Neural Network [1]. Trong bài báo chúng tôi tìm hiểu thì tác giả sử dụng giải thuật Random Forest là một trong các giải thuật học máy kết hợp (Ensemble learning algorithm).

Đóng góp chính của bài báo là việc áp dụng giải thuật Random Forest để dự đoán xu hướng của giá chứng khoán. Sau đó so sánh giải thuật này với các giải thuật học máy trước đây như: Logistic Regression, Gaussian Discriminant Analysis, Quadratic Discriminant Analysis, and SVM, trên cùng bộ dữ liệu của giải thuật đó. Kết quả khi so sánh thì giải thuật Random Forest của tác giả mang lại độ chính xác cao hơn các bài báo trước.

Với giải thuật Random Forest tác giả sử dụng bộ dữ liệu có 5 thuộc tính theo từng ngày của một loại chứng khoán: giá mở cửa, giá đóng cửa, giá thấp nhất, giá cao nhất, khối lượng giao dịch. Dựa vào 5 thuộc tính này tác giả đã tính toán 6 đặc trưng của chứng khoán để làm đầu vào cho mô hình Random Forest. Đầu ra của mô hình này là giá trị phân lớp 1 và -1. Với giá trị 1 nghĩa là chứng khoán có xu hướng tăng giá và ngược lại thì -1 nghĩa là chứng khoán có xu hướng giá sẽ giảm.

Mô hình dự đoán xu hướng giá chứng khoán được tác giả xây dựng qua các bước như sau:



Hình 1.1: Quá trình xây dựng mô hình dự đoán

Ở bước thu thập dữ liệu, tác giả sử dụng dữ liệu chứng khoán của Apple (AAPL) với cửa sổ thời gian là 30 ngày để huấn luyện và kiểm thử mô hình. Ở bước xử lý dữ liệu tác giả đã làm mịn dữ liệu và rút 6 đặc trưng từ 5 thuộc tính trong tập dữ liệu. Cụ thể của việc xử lý dữ liệu sẽ được trình bày trong phần tiếp đi.

1.2 Tiền xử lý dữ liệu và rút trích đặc trưng

1.2.1 Tiền xử lý dữ liệu

Trước khi tác giả rút trích đặc trưng từ các thuộc tính trong dữ liệu, tác giả sử dụng một hàm để làm mịn dữ liệu được gọi là Exponential smoothing. Hàm làm mịn dữ liệu sẽ loại bỏ bớt những dữ liệu ngẫu nhiên và nhiễu từ các giá trị trong quá. Việc làm này để tránh các nhiễu trong quá khứ sẽ ảnh hưởng nhiều đến việc dự đoán.

Ta đặt:

$$S_0 = Y_0$$

Hàm exponentially smoothed được tính như sau:

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

Với S_t là giá trị đã được làm mịn tại thời gian t , Y_t là giá trị thực tế tại thời gian t.

Và $0 < \alpha < 1$, khi $\alpha = 1$ thì giá trị của S_t bằng giá trị thực tế.

Mong muốn của bài toán là dự đoán xu hướng giá chứng khoán của ngày thứ i, xu hướng tăng hay giảm của chứng khoán biểu diễn qua giá trị $target$ được định nghĩa như sau:

$$target_i = Sign(close_{i+d} - close_i)$$

Với Sign là hàm lấy dấu + và –, $close_i$ là giá đóng cửa của ngày trong quá khứ và $close_{i+d}$ là giá đóng cửa của ngày số ngày d trong tương lai.

Nếu $target_i$ ra +1 thì có nghĩa giá chứng khoán tăng sau d ngày và ngược lại là -1 thì giá chứng khoán là giảm.

1.2.2 Rút trích đặc trưng

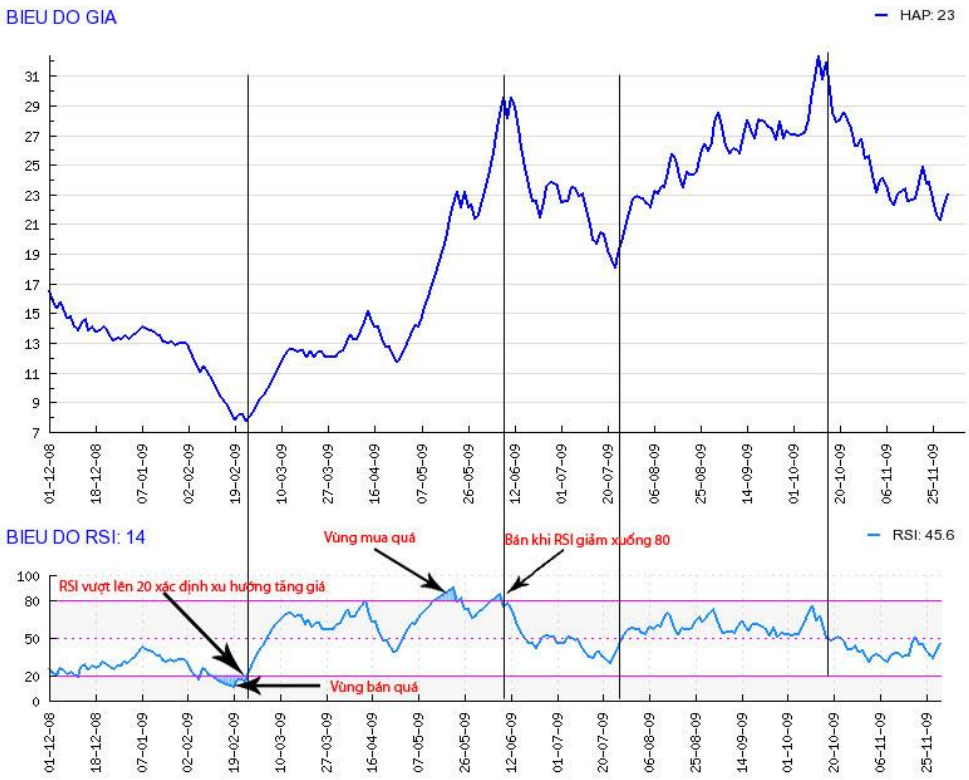
6 đặc trưng mà tác giả sử dụng bao gồm: chỉ số RSI, chỉ số Stochastic Oscillator (%K), chỉ số Williams %R, đường Moving Average Convergence Divergence (MACD), chỉ số Price Rate of Change, chỉ số On Balance Volume. Các đặc trưng này sẽ được định nghĩa và giải thích như sau:

- **Chỉ số Relative Strength Index (RSI)**

$$RSI = 100 - \frac{100}{1 + RS}$$

$$RS = \frac{Giá\ trị\ tăng\ trung\ bình\ trong\ 14\ ngày}{Giá\ trị\ giảm\ trung\ bình\ trong\ 14\ ngày}$$

Chỉ số này mang giá trị từ 0 – 100, khi RSI của chứng khoán > 70 là dấu hiệu giá chứng khoán sắp giảm và RSI của chứng khoáng < 30 là dấu hiệu giá chứng khoán sắp tăng.



Hình 1.2: Hình minh họa chỉ số RSI

(Nguồn: https://www.cophieu68.vn/document/rsi.php)

– **Chỉ số Stochastic Oscillator (%K)**

$$\%K = 100 * \frac{(C - L14)}{(H14 - L14)}$$

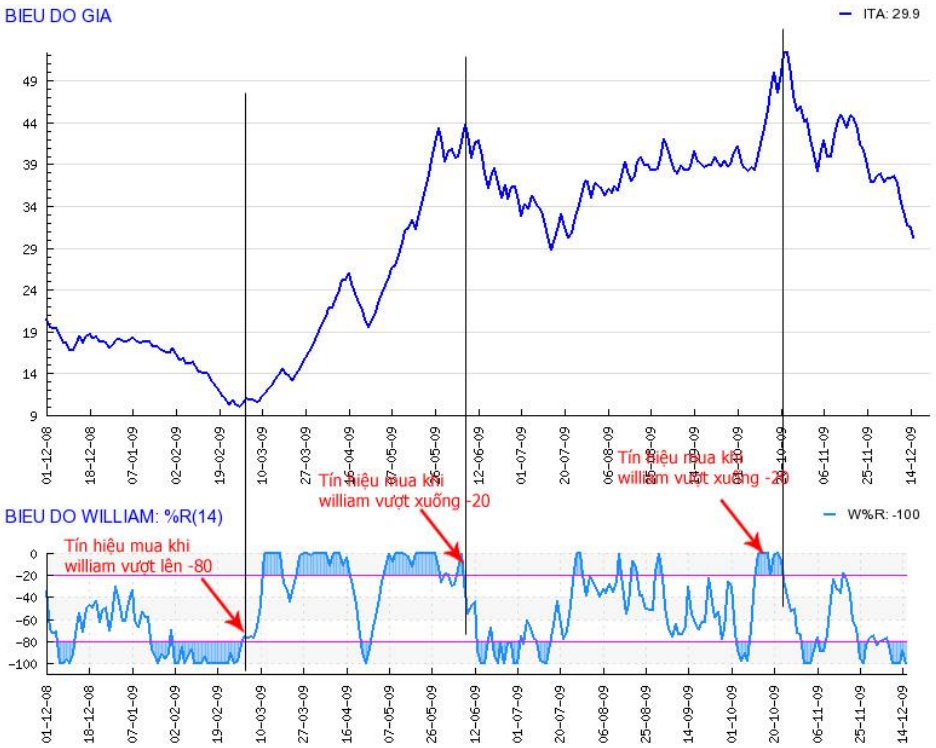
Với C là giá đóng cửa hiện tại, $L14$ là giá thấp nhất trong 14 ngày và $H14$ là giá cao nhất trong vòng 14 ngày. Nếu chỉ số này tăng trên 75 thì nên mua lại. Nếu chỉ số giảm dưới 25 thì nên bán ra.

– **Chỉ số Williams %R**

$$\%R = \frac{(H14 - C)}{(H14 - L14)} * -100$$

Với C là giá đóng cửa hiện tại, $L14$ là giá thấp nhất trong 14 ngày và $H14$ là giá cao nhất trong vòng 14 ngày. Chỉ số này giao động giữa -100 và 0. Giá trị này trên -20

là dấu hiệu chứng khoán được bán ra nhiều, ngược lại nếu dưới -80 thì chứng khoán này đang được mua lại nhiều.



Hình 1.3: Hình minh họa chỉ số William %R

(Nguồn: https://www.cophieu68.vn/document/william.php)

– **Đường Moving Average Convergence Divergence (MACD)**

$$MACD = EMA_{12}(C) - EMA_{26}(C)$$

$$SignalLine = EMA_9(MACD)$$

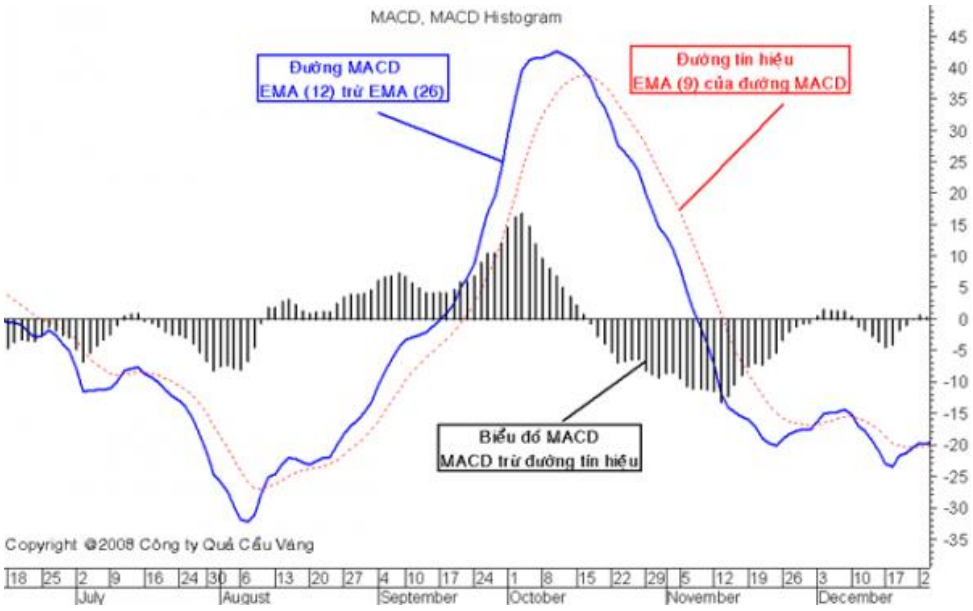
Với:

EMA là viết tắt của Exponential Moving Average.

EMA_{12} là đường trung bình giá của 12 phiên gần nhất và EMA_{26} là đường trung bình giá của 26 phiên gần nhất.

Đường tín hiệu $SignalLine$ là đường EMA_9 của đường $MACD$.

Nếu đường MACD dưới đường tín hiệu thì chứng khoán đang có dấu hiệu bán và ngược lại nằm trên đường tín hiệu thì chứng khoán đang có hiệu mua.



Hình 1.4: Hình minh họa đường MACD

(Nguồn: https://www.cophieu68.vn/document/macd.php)

– **Chỉ số Price Rate of Change**

$$PROC(t) = \frac{C(t) - C(t-n)}{C(t-n)}$$

Với t là thời điểm hiện tại và n là số ngày trước đó. $C(t)$ là giá đóng cửa của thời điểm t. Chỉ số này thể hiện sự liên quan giữa giá đóng cửa của thời điểm đang xét và giá đóng cửa của n ngày trước đó.

– **Chỉ số On Balance Volume**

$$OBC(t)\begin{cases} OBV(t-1) + Vol(t) & if\ C(t) > C(t-1) \\ OBV(t-1) - Vol(t) & if\ C(t) < C(t-1) \\ OBV(t-1) & if\ C(t) = C(t-1) \end{cases}$$

Với $Vol(t)$ là khối lượng giao dịch tại thời điểm t. Chỉ số này dùng để xác nhận xu hướng của chứng khoán và tìm các điểm khả năng đảo chiều giá.

CHƯƠNG 2 – MÔ HÌNH DỰ ĐOÁN GIÁ CHỨNG KHOÁN

2.1 Decision tree

Cây quyết định là mô hình học máy có giám sát (supervised learning) dùng cho việc dự báo một sự việc. Một cây quyết định là một đồ thị chia thành các nút mang các đặc trưng quyết định. Nút lá của cây quyết định có thể mang kết quả rời rạc hoặc liên tục nhưng đều đại diện cho kết quả phân loại hoặc lựa chọn.

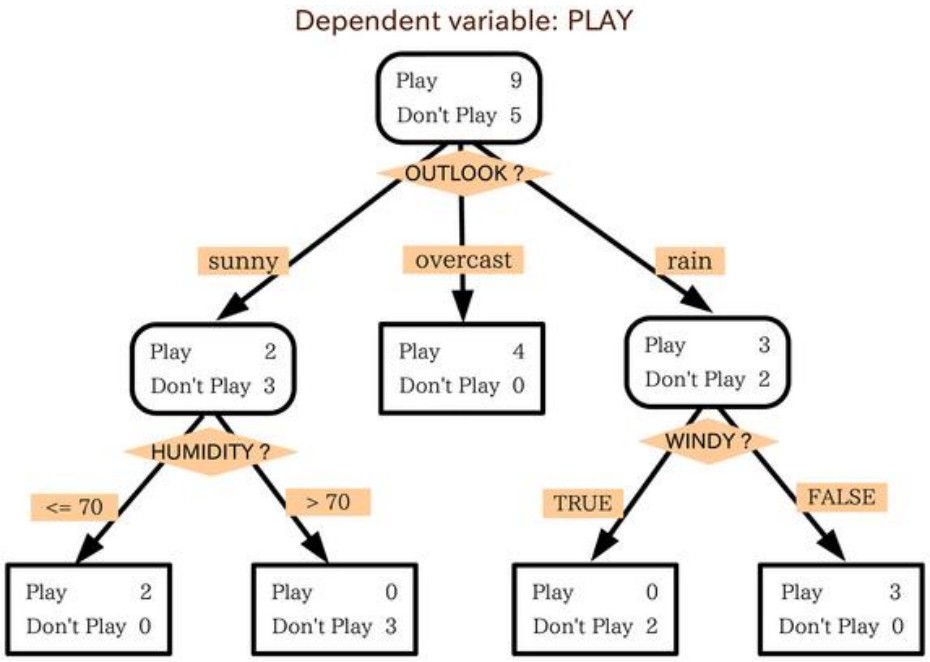
Một ví dụ của cây quyết định:

Play golf dataset

Independent variables				Dep. var
OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
sunny	85	85	FALSE	Don't Play
sunny	80	90	TRUE	Don't Play
overcast	83	78	FALSE	Play
rain	70	96	FALSE	Play
rain	68	80	FALSE	Play
rain	65	70	TRUE	Don't Play
overcast	64	65	TRUE	Play
sunny	72	95	FALSE	Don't Play
sunny	69	70	FALSE	Play
rain	75	80	FALSE	Play
sunny	75	70	TRUE	Play
overcast	72	90	TRUE	Play
overcast	81	75	FALSE	Play
rain	71	80	TRUE	Don't Play

Hình 2.1: Dữ liệu minh họa để xây dựng cây quyết định

(Nguồn: https://vi.wikipedia.org/wiki/Cây_quyết_định)



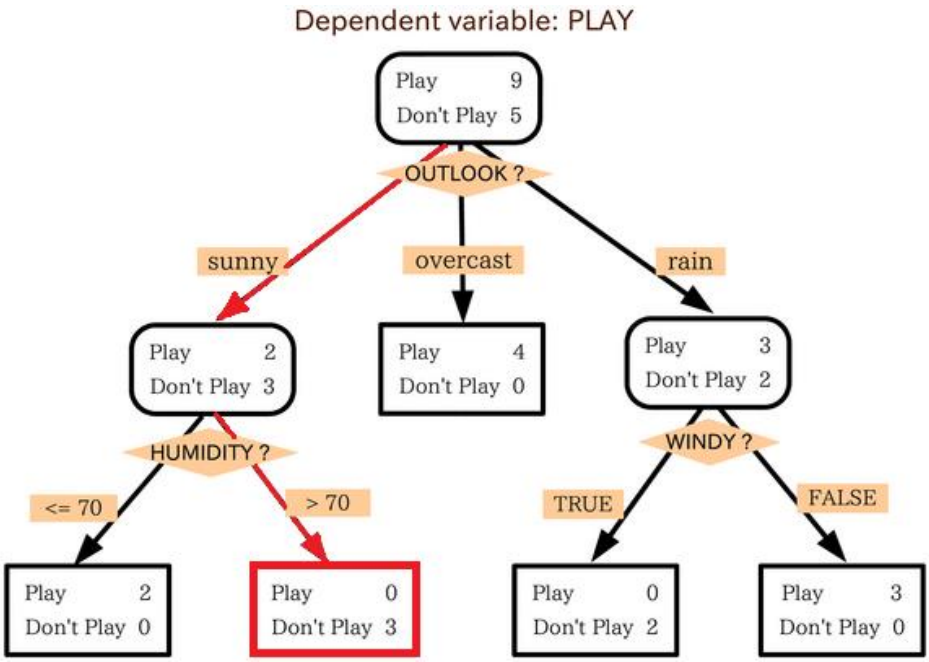
Hình 2.2: Hình minh họa cho cây quyết định

(Nguồn: https://vi.wikipedia.org/wiki/Cây_quyết_định)

Với cây quyết định minh họa trên, các thuộc tính trong bảng dữ liệu sẽ được thống kê và dùng hàm tính toán (Gini Impurity hoặc Information Gain) để lựa chọn ra thuộc tính mang nhiều thông tin làm nút gốc, các thuộc tính mang càng ít thông tin thì càng gần lá.

Giả sử cho một dữ liệu mới, cách cây quyết định trên đưa ra lựa chọn với dữ liệu mới với đưa vào như sau:

Outlook = sunny, Temperature = 73, Humudity = 98, Windy = FALSE thì cây quyết định sẽ ra lựa chọn là Don’t Play.



Hình 2.3: Hình minh họa cho cây quyết định lựa chọn

Tuy nhiên cây quyết định dễ xảy ra tình trạng quá khớp dữ liệu huấn luyện do đó rất nhạy cảm với dữ liệu nhiễu. Trong tập dữ liệu huấn luyện chỉ có cần vài dữ liệu nhiễu sẽ làm cho cây bị thay đổi theo hướng dự đoán kết quả sai với các dữ liệu mới. Nguyên nhân xảy ra tình trạng này là do cây quyết định có độ lệch (bias) thấp và phương sai (variance) cao.

Bias và variance là đại lượng đặc trưng cho việc mô hình sử dụng dữ liệu huấn luyện trong quá trình huấn luyện. Bias cao nghĩa là mô hình không quan tâm tới dữ liệu huấn luyện, bias cao khi dữ liệu huấn luyện quá nhiều nhưng mô hình huấn luyện quá đơn giản. Variance cao nghĩa là mô hình quá phụ thuộc vào dữ liệu huấn luyện dẫn đến chỉ khớp với dữ liệu huấn luyện mà không khớp với dữ liệu mới để kiểm thử (Overfit), variance cao khi dữ liệu huấn luyện ít so với mô hình được sử dụng để huấn luyện quá phức tạp. Mô hình của cây quyết định là mô hình phức tạp do đó dễ bị tình trạng Overfit.

Random forest (rừng ngẫu nhiên) được phát triển để khắc phục những nhược điểm này.

2.2 Random forest

Random forest là thuật toán xây dựng nhiều cây quyết định và kết quả lựa chọn cuối cùng là kết quả được nhiều cây bỏ phiếu chọn nhất. Bằng cách lấy ngẫu nhiên D' dữ liệu là tập con từ *tập D* dữ liệu ban đầu để huấn luyện ra các cây quyết định khác nhau, điều này đồng nghĩa với việc không cây quyết định nào được xây dựng bằng toàn bộ dữ liệu huấn luyện, với cách làm này cây quyết định được dựng ra trong Random forest sẽ có không bị quá khớp (Overfit) với tập huấn luyện nhưng đổi lại bias của mô hình này sẽ tăng một ít so với cây quyết định. Dữ liệu sẽ được chia thành các tập nhỏ để xây dựng thành các cây và các thuộc tính được chọn làm nút trong cây thông qua hàm Gini Impurity hoặc Shannon Entropy. Đặc điểm của cả 2 hàm này là tìm ra thuộc tính có mang nhiều thông tin sẽ được để ở những nút gần gốc của cây. Khái niệm mang nhiều thông tin, ví dụ ở dữ liệu của cây quyết định được xây dựng tại hình minh họa phần 2.1

Play golf dataset

Independent variables				Dep. var
OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
sunny	85	85	FALSE	Don't Play
sunny	80	90	TRUE	Don't Play
overcast	83	78	FALSE	Play
rain	70	96	FALSE	Play
rain	68	80	FALSE	Play
rain	65	70	TRUE	Don't Play
overcast	64	65	TRUE	Play
sunny	72	95	FALSE	Don't Play
sunny	69	70	FALSE	Play
rain	75	80	FALSE	Play
sunny	75	70	TRUE	Play
overcast	72	90	TRUE	Play
overcast	81	75	FALSE	Play
rain	71	80	TRUE	Don't Play

Thuộc tính Outlook mang 3 giá trị sunny, overcast và rain.

sunny: 3 *Don't Play*, 2 *Play*

overcast: 0 *Don't Play*, 4 *Play*

rain: 2 *Don't Play*, 3 *Play*

Thuộc tính Windy mang giá trị TRUE và FALSE.

TRUE: 3 *Don't Play*, 3 *Play*

FALSE: 2 *Don't Play*, 6 *Play*

Khi nhìn vào số liệu trên ta có thể thấy **Outlook** mang giá trị ***overcast*** ra thẳng được quyết định là *Play*. **Windy** chỉ có giá trị ***FALSE*** là có kết quả *Don't Play* và *Play* chênh lệch lớn giữa nhưng vẫn chưa đủ để ra quyết định ngay. Càng nhiều giá trị có kết quả đầu ra bị nghiêng về một lớp chứng tỏ thuộc tính mang nhiều thông tin quyết định hơn (Information Gain lớn) để làm nút gốc.

Gini Impurity

Xét tại nút N

$$g(N) = \sum_{i \neq j} P(\omega_i)P(\omega_j)$$

Với $P(\omega_i)$ là tỉ số của từng giá trị trong thuộc tính cho ra kết quả lớp i và tổng số giá trị là kết quả của lớp i, ở ví dụ ở phía trên ta có 5 giá trị lớp *Don't Play* và 9 giá trị lớp *Play*, giả sử xét **Outlook** là nút đầu tiên thì Gini(Outlook) sẽ là:

sunny: $\frac{3}{5} * \frac{2}{9} = \frac{2}{15}$

overcast: $\frac{0}{5} * \frac{3}{9} = 0$

rain: $\frac{2}{5} * \frac{3}{9} = \frac{2}{15}$

$$Gini(Outlook) = \frac{2}{15} + 0 + \frac{2}{15} = \frac{4}{15} \approx 0.2667$$

Giá trị Gini của thuộc tính càng nhỏ thì thuộc tính càng mang nhiều giá trị lệch do đó các thuộc tính có Gini nhỏ được chọn làm gốc hoặc các nút gần gốc.

Shannon Entropy

Xét tại nút N

$$H(N) = -\sum_{i=1}^{i=d} P(\omega_i)\log_2(P(\omega_i))$$

Với d là số lớp và $P(\omega_i)$ là tỉ số của từng giá trị trong thuộc tính cho ra kết quả lớp i và tổng số giá trị là kết quả của lớp i do giá trị trong thuộc tính ảnh hưởng đến. Ví dụ của Shannon Entropy sẽ được kết hợp chung với hàm Information Gain bên dưới để làm ví dụ cụ thể.

Information Gain

$$\Delta I(N) = I(N) - P_L * I(N_L) - P_R * I(N_R)$$

Với $I(N)$ là hàm đo lường impurity – chính là hàm Gini Impurity hoặc Shannon Entropy đã được trình bày bên trên. P_L và P_R là tỉ số của kết quả phân lớp trên tổng số kết quả N theo từng giá trị của thuộc tính. Ví dụ dùng Shannon Entropy để tính Information Gain chọn nút gốc cho dữ liệu đã trình bày bên trên. Dữ liệu có 5 *Don't Play* và 9 *Play*, tổng là 14 kết quả.

$$H(tập\ dữ\ liệu) = -\frac{5}{14}\log_2\left(\frac{5}{14}\right) + \left(-\frac{9}{14}\log_2\left(\frac{9}{14}\right)\right) \approx 0.94$$

Với sunny có 3 *Don't Play*, 2 *Play*, tổng là 5 kết quả:

$$H(sunny) = -\frac{3}{5}\log_2\left(\frac{3}{5}\right) + \left(-\frac{2}{5}\log_2\left(\frac{2}{5}\right)\right) \approx 0.971$$

Với overcast có 0 *Don't Play*, 4 *Play*, tổng là 4 kết quả:

$$H(overcast) = -1\log_2(1) + (-0\log_2(0)) = 0$$

Với rain có 2 *Don't Play*, 3 *Play*, tổng là 5 kết quả:

$$H(rain) = -\frac{2}{5}\log_2\left(\frac{2}{5}\right) + \left(-\frac{3}{5}\log_2\left(\frac{3}{5}\right)\right) \approx 0.971$$

Tính Information Gain thuộc tính Outlook:

$$I(Outlook) = H(tập\ dữ\ liệu) - \frac{5}{14} * 0.971 - \frac{4}{14} * 0 - \frac{5}{14} * 0.971 = 0.247$$

Tương tự cách tính với 3 thuộc tính còn lại của tập dữ liệu thì Information Gain của Outlook là cao nhất do đó trong hình minh họa của phần 2.1 thì thuộc tính Outlook được chọn làm gốc của cây. Tương tự với các nút tiếp theo thì Information Gain được tính dựa vào thuộc tính với nút ngay bên trên thay vì tính với tập dữ liệu như nút gốc.

Ý tưởng của Random Forest là chia dữ liệu ra để xây dựng nhiều cây khác nhau, việc chia dữ liệu này được gọi là phương pháp Bootstrap aggregating hay bagging. Phương pháp này làm tăng độ chính xác khi huấn luyện và giảm Variance để tránh Overfit. Nói đơn giản hơn, thay vì chỉ có một cây duy nhất cho toàn bộ dữ liệu dễ gây Overfit thì ta chia nhỏ tập dữ liệu và xây dựng thành nhiều cây con. Cho một tập dữ liệu *D* kích thước *n*, phương pháp bagging sẽ tạo ra *B* tập dữ liệu mới với kích thước *n'* và dữ liệu có thể lấy trùng lắp với nhau – nghĩa là một dòng dữ liệu trong tập huấn luyện *D* có thể ngẫu nhiên được lấy nhiều lần và xuất hiện nhiều lần tập huấn luyện *D'* cho một cây con.

Mã giả của Random Forest được viết như sau:

Algorithm 1 Random Forest Classifier

```
1:  procedure RandomForestClassifier(D)   ▷ D is the labeled training data
2:      forest = new Array()
3:      for  do i = 0 to B
4:          D_i = Bagging(D)                          ▷ Bootstrap Aggregation
5:          T_i = new DecisionTree()
6:          features_i = RandomFeatureSelection(D_i)
7:          T_i.train(D_i, features_i)
8:          forest.add(T_i)
9:      end for
10:     return forest
11: end procedure
```

Diễn giải đoạn mã giả trên:

1. Tạo thủ tục RandomForestClassifier với tham số *D* là tập dữ liệu huấn luyện lớn
2. Tạo một mảng chứa cây gọi là *forest*

3. Chạy vòng lặp với số lần lặp là số lượng cây *B* mong muốn tạo ra cho mô hình
4. Rút ngẫu nhiên dữ liệu tạo thành tập *D'* từ tập *D* (Bootstrap aggregating)
5. Khởi tạo cây *T* mới
6. Lựa chọn ngẫu nhiên một số thuộc tính *features* trong tập *D'* để tham gia huấn luyện
7. Huấn luyện cây *T* với dữ liệu *D'* và các thuộc tính *features* đã được chọn
8. Tham cây *T* vào mảng *forest*

Tác giả bài báo đã thử nghiệm giải thuật Random Forest vào tập dữ liệu của Apple với cửa sổ thời gian là 30 ngày. Số lượng cây tạo ra là 30 cây.

Với dòng dữ liệu sau:

- RSI: 91.638968318801957
- Stochastic Oscillator: 88.201032228068314
- Williams: -11.798967771931691
- Moving Average Convergence Divergence: 5.9734426013145026
- Price Rate of Change: 0.11162583681857041
- On Balance Volume: 6697423901.3580704

Một cây được sinh ra cho kết quả có dạng như sau:

Tại nút 0:(MACD=5.97344260131) <= -8.6232?

Sai: Đến nút 10

Tại nút 10:(Stochastic Oscillator=88.2010322281) <= 80.9531?

Sai: Đến nút 134

Tại nút 134:(MACD=5.97344260131) <= 10.2566?

Đúng: Đến nút 135

Tại nút 135:(RSI=91.6389683188) <= 97.9258?

Đúng: Đến nút 136

Tại nút 136:(MACD=5.97344260131) <= -1.7542?

Sai: Đến nút 140

Tại nút 140:(Price Rate Of Change=0.0412189145804) <= 0.0708?

Đúng: Đến nút 141

Tại nút 141:(MACD=5.97344260131) <= 9.2993?

Đúng: Đến nút 142

Tại nút 142:(MACD=5.97344260131) <= 7.7531?

Đúng: Đến nút 143

Tại nút 143:(On Balance Volume=23722858211.2) <= 24938491904.0?

Đúng: Đến nút 144

Tại nút 144:(Williams=-11.7989677719) <= -15.7228?

Sai: Đến nút 154

Nút lá 154 có kết quả là Tăng

Trong thử nghiệm của tác giả thì 30 cây được tạo ra có 29 cây cho kết quả là Tăng (+1) và 1 cây cho kết quả Giảm (-1). Kết quả Tăng hay Giảm là kết quả tăng hay giảm giá đóng cửa với loại chứng khoán đang xét sau 30 ngày. Kết quả khớp với kết quả của tập thử nghiệm.

Việc lựa chọn ngẫu nhiên của Random Forest sẽ sinh ra vấn đề là liệu khi lựa chọn như vậy thì mô hình này có cho kết quả hội tụ hay không ? Để sự hội tụ của mô hình Random Forest, khái niệm độ lỗi Out of bag (OOB) được sử dụng. Giả sử trong cây T_1 được tạo ra từ bag D_1 với kích thước n' thì sẽ có một lượng dữ liệu m không nằm trong tập D_1. Sau khi cây T_1 được huấn luyện xong dữ liệu m sẽ được đưa vào cây T_1 để dự đoán. Lỗi OOB của cây T_1 là tỉ lệ dữ liệu trong m mà T_1 dự đoán sai. Lỗi OOB của một mô hình Random Forest là trung bình OOB của tất cả các cây.

Trading Period (Days)	No. of Trees	Sample Size	OOB error
30	5	6590	0.241729893778
30	25	6590	0.149165402124
30	45	6590	0.127617602428
30	65	6590	0.123672230653
60	5	6545	0.198472116119
60	25	6545	0.0890756302521
60	45	6545	0.0786860198625
60	65	6545	0.0707410236822
90	5	6500	0.191384615385
90	25	6500	0.0741538461538
90	45	6500	0.0647692307692
90	65	6500	0.0555384615385

Bảng 2.1: Độ lỗi OOB trên tập dữ liệu AAPL

(Nguồn: Khaidem et. al., Predicting the direction of stock market prices using random forest, (2016))

Với bảng trên cột đầu tiên là khoảng thời gian dự đoán giá, cột thứ 2 là số lượng cây sinh ra, cột thứ 3 là số lượng dữ liệu tập huấn luyện sử dụng và cột thứ 4 là độ lỗi OOB. Tác giả đã thử nghiệm với số lượng cây càng tăng thì độ lỗi OOB sẽ càng giảm.

2.3 Độ lỗi OOB và sự hội tụ của Random Forest

Gọi θ là mô hình cây quyết định trong Random Forest, mỗi cây trong Random Forest được biểu diễn bởi $h(X|\theta)$ với X là biến ngẫu nhiên đại diện cho tập dữ liệu được bagging chọn ra. Hàm giá trị biên của cây quyết định:

$$margin_{RF}(x,y) = P_\theta(h(x|\theta) = y) - max_{j\neq y}P_\theta(h(x|\theta) = j)$$

Gọi s là độ mạnh của Random Forest, s được định nghĩa là giá trị kỳ vọng của xác suất biên:

$$s = E_{x,y}(margin_{RF}(x,y))$$

Dựa vào bất đẳng thức Chebyshev có:

$$Error = P_{x,y}(margin_{RF}(x,y) < 0) \leq P_{x,y}(|margin_{RF}(x,y) - s| \geq s) \leq \frac{var(margin_{RF}(x,y))}{s^2}$$

Với $var(margin_{RF}(x,y))$ là phương sai của giá trị biên. Chứng minh bất đẳng thức về độ lỗi của Random Forest:

$$Error \leq \frac{var(margin_{RF}(x,y))}{s^2}$$

Bất đẳng thức Chebyshev định nghĩa:

Với X là biến ngẫu nhiên và một biến $C > 0$ ta sẽ có:

$$P(|X - E(X)| \geq c) \leq \frac{var(x)}{c^2}$$

Ta có 2 tiên đề:

A) Biến ngẫu nhiên chỉ báo:

$$I(X \geq c) = \begin{cases} 1 & if\ X \geq c \\ 0 & Otherwise \end{cases}$$

Hàm I có giá trị là 1 khi $X \geq c$ xảy ra và ngược lại thì bằng 0.

B) Không gian đo lường

$$A = \{x \in \Omega | X(x) \geq c\}$$

$$E(X) = \sum_{x \in A} P(x)X(x) = \mu$$

Chứng minh lại bất đẳng thức Chebyshev:

Ta có, $A = \{x \in \Omega | X(x) - E(x) \geq c\}$

$$var(X) = \sum_{x \in \Omega} P(X = x)\big(X(x) - E(x)\big)^2$$

$$= \sum_{x \in A} P(X = x)\big(X(x) - E(x)\big)^2 + \sum_{x \notin A} P(X = x)\big(X(x) - E(x)\big)^2 \geq 0$$

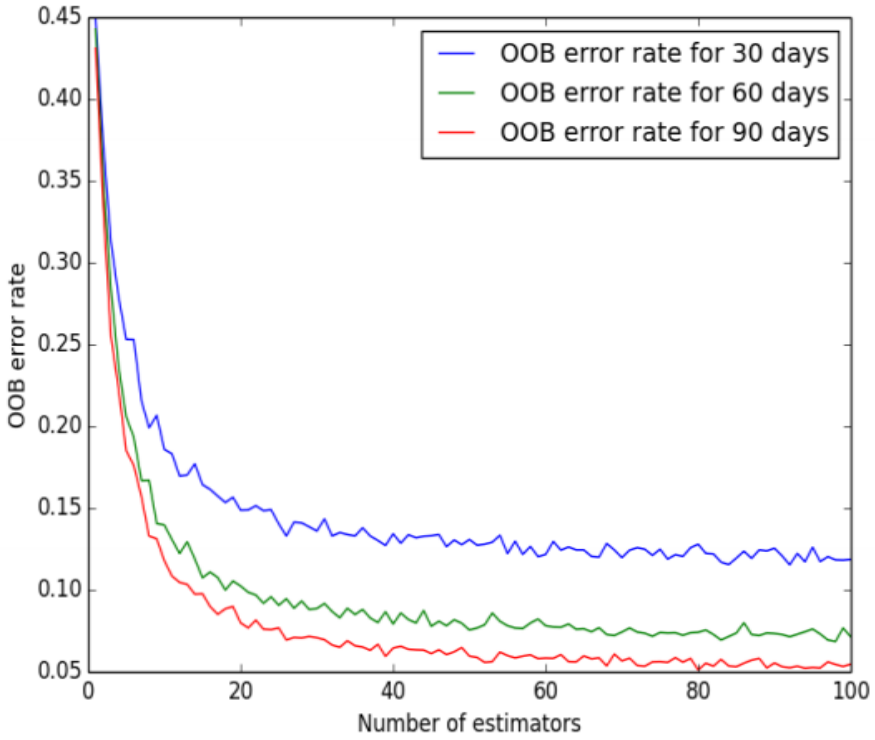
$$\geq \sum_{x \in A} P(X = x)\big(X(x) - E(x)\big)^2$$

$$\geq P(X = x)c^2 \text{ bởi vì } X(x) - E(x) \geq C; \forall x \in A$$

$$= c^2 P(A) = c^2 P(|X - E(X)| \geq c)$$

$$\Rightarrow \frac{var(X)}{c^2} \geq P(|X - E(X)| \geq c)$$

Tới đây đã chứng minh được độ lỗi của Random Forest luôn nhỏ hơn giá trị phương sai chia kỳ vọng bình phương. Do đó độ lỗi này không thể bằng 0 nên mô hình Random Forest sẽ không bị tình trạng Overfit.



Hình 2.4: Biểu đồ độ lỗi OOB với tập dữ liệu AAPL

Qua biểu đồ này cũng thấy được mô hình Random Forest càng nhiều cây thì độ phủ càng cao, lỗi OOB càng giảm nhưng không chạm 0, nghĩa là mô hình này đang không bị Overfit.

CHƯƠNG 3 – KẾT QUẢ VÀ DEMO

3.1 Kết quả thực nghiệm

Với các bài toán phân lớp nhị phân, một lớp sẽ được gọi là lớp positive và lớp còn lại gọi là lớp negative.

$\boldsymbol{tp}$ = số lượng giá trị positive được phân lớp đúng

$\boldsymbol{tn}$ = số lượng giá trị negative được phân lớp đúng

$\boldsymbol{fp}$ = số lượng giá trị negative bị phân lớp sai thành positive

$\boldsymbol{fn}$ = số lượng giá trị positive bị phân lớp sai thành negative

Các độ đo thường được sử dụng để đánh giá mô hình phân lớp nhị phân là:

Accuracy: đo độ chính xác của mô hình mà không quan tâm đến độ chính xác của từng lớp.

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{tp} + \boldsymbol{tn}}{\boldsymbol{tp} + \boldsymbol{tn} + \boldsymbol{fp} + \boldsymbol{fn}}$$

Precision: đo độ chính xác khi phân lớp positive

$$\boldsymbol{Precision} = \frac{\boldsymbol{tp}}{\boldsymbol{tp} + \boldsymbol{fp}}$$

Recall: đo độ phủ của lớp positive, độ đo này quan tâm đến kết quả phân lớp cho lớp positive.

$$\boldsymbol{Recall} = \frac{\boldsymbol{tp}}{\boldsymbol{tp} + \boldsymbol{fn}}$$

Specificity: đo độ phủ của lớp negative

$$\boldsymbol{Specificity} = \frac{\boldsymbol{tn}}{\boldsymbol{tn} + \boldsymbol{fp}}$$

Trading Period	Accuracy%	Precision	Recall	Specificity
1 month	86.8396	0.881818	0.870736	0.865702
2 months	90.6433	0.910321	0.92599	0.880899
3 months	93.9664	0.942004	0.950355	0.926174

Bảng 3.1: Kết quả với tập dữ liệu của chứng khoán Samsung

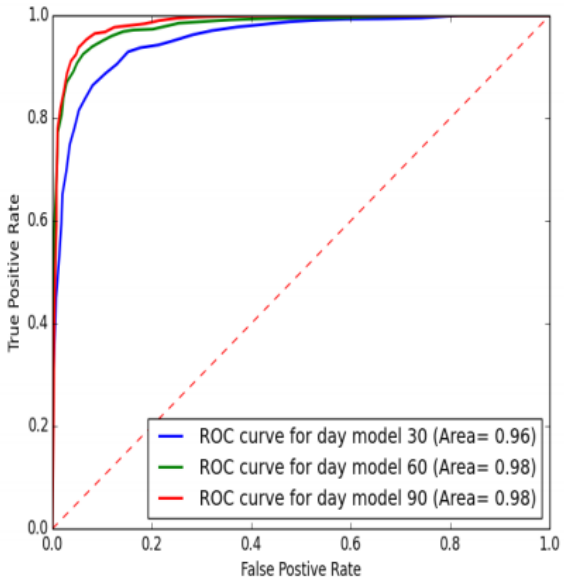
Trading Period	Accuracy%	Precision	Recall	Specificity
1 month	88.264	0.89263	0.90724	0.84848
2 months	93.065	0.94154	0.93858	0.91973
3 months	94.533	0.94548	0.96120	0.92341

Bảng 3.2: Kết quả với tập dữ liệu của chứng khoán Apple Inc.

Trading Period	Accuracy%	Precision	Recall	Specificity
1 month	84.717	0.85531	0.87637	0.80968
2 months	90.831	0.91338	0.93099	0.87659
3 months	92.543	0.93128	0.94557	0.89516

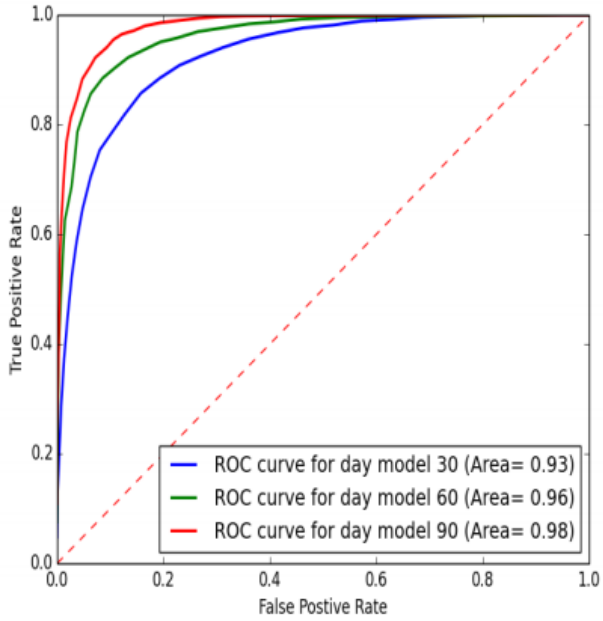
Bảng 3.3: Kết quả với tập dữ liệu của chứng khoán GE

Với 3 tập dữ liệu trên có thể thấy tất cả các độ đo đều đạt trên 0.8, đây là một kết quả cao đối với các bài toán về dự đoán phân lớp. Tác giả cũng đã vẽ ra các biểu đồ Receiver Operating Characteristic (ROC) để cho thấy độ chính xác của mô hình với tập kiểm thử, khi đường cong càng gần về góc trái biểu đồ nghĩa là độ chính xác khi kiểm thử cao.

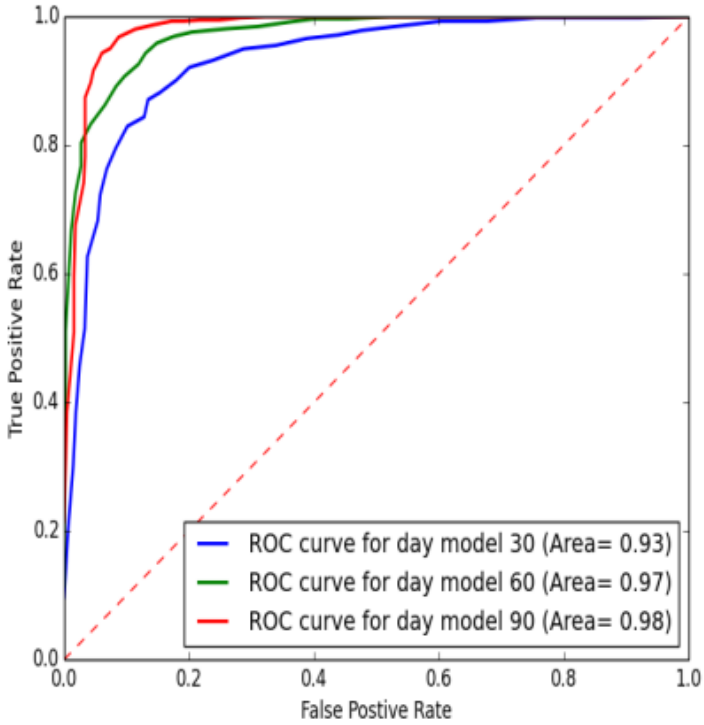


Hình 3.1: Biểu đồ với tập dữ liệu AAPL

Trên cả 3 tập dữ liệu thì mô hình 90 ngày luôn cho độ chính xác cao nhất.



Hình 3.2: Biểu đồ với tập dữ liệu GE

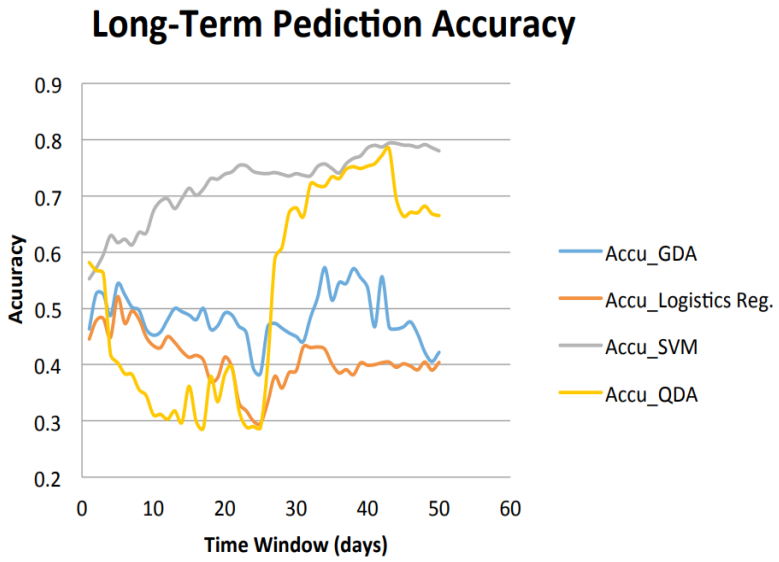


Hình 3.3: Biểu đồ với tập dữ liệu Samsung

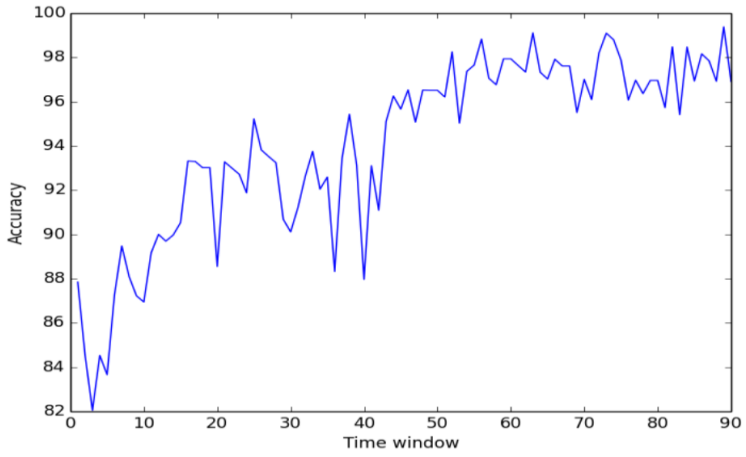
3.2 So sánh với các kết quả trước đây

Tác giả đã sử dụng mô hình trên vào dữ liệu của 3 bài báo dùng thuật toán khác và so sánh kết quả.

Với kết quả của tác giả Dai và Zhang (2013) [2] đạt độ chính xác cao nhất là thuật toán SVM sử dụng dữ liệu của 44 ngày đạt 79,3% và khi tác giả bài báo này áp dụng vào dữ liệu trên thì độ chính xác cao nhất là 96,92% khi sử dụng dữ liệu của 88 ngày.



Hình 3.4: Biểu đồ kết quả của tác giả Dai và Zhang (2013)



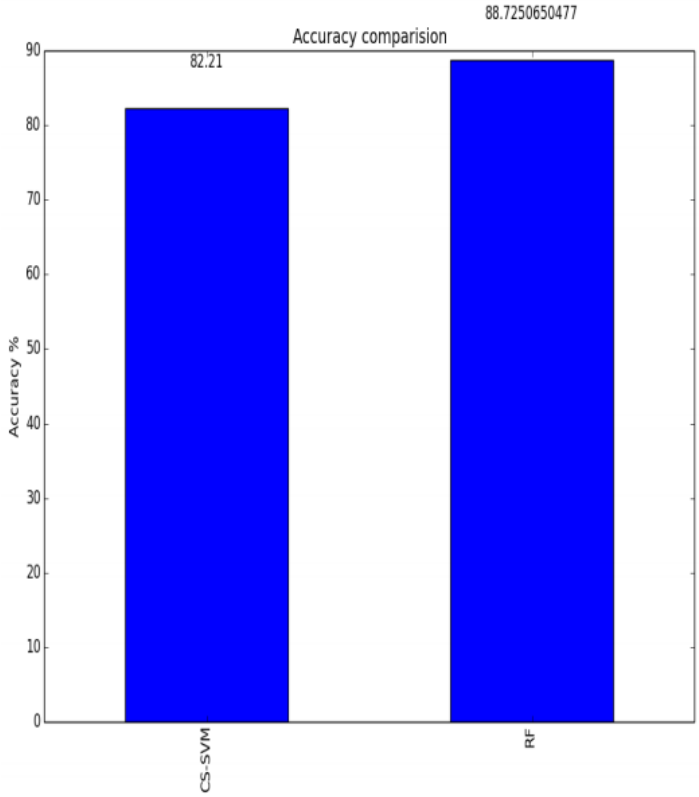
Hình 3.5: Biểu đồ kết quả của tác giả Khaidem (2016)

Company/Accuracy	Next 3-day	Next 5-day	Next 7-day	Next 10-day
Apple	73.4%	71.41%	70.25%	71.13%
Amazon	63%	65%	61.5%	71.25%
Microsoft	64.5%	73%	77.125%	77.25%

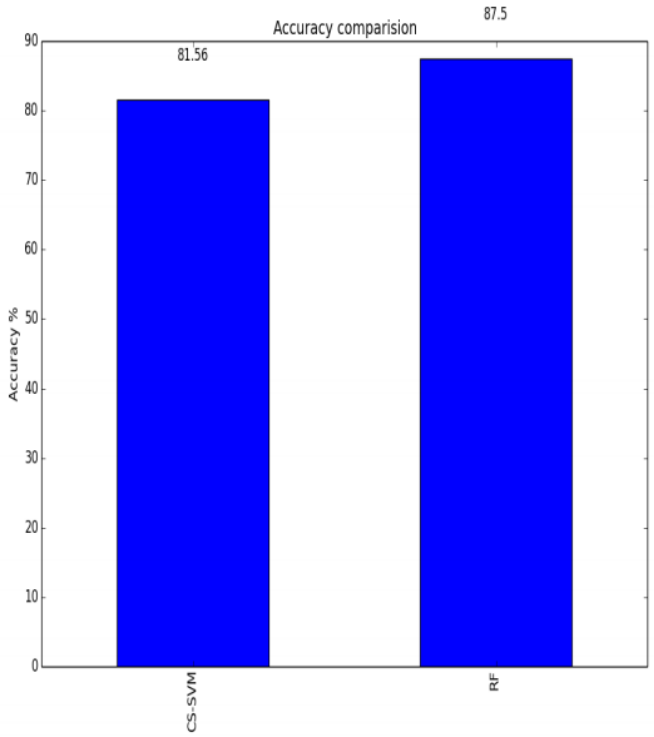
Hình 3.6: Kết quả của tác giả Xinjie (2014) [3]

Company/Accuracy	Next 3-day	Next 5-day	Next 7-day	Next 10-day
Apple	85.197%	83.88%	88.11%	92.08%
Amazon	86.51%	88.49%	85.14%	87.46%
Microsoft	84.59%	83.88%	89.47%	86.46%

Hình 3.7: Kết quả của tác giả Khaidem (2016)



Hình 3.8: Kết quả so sánh với tác giả Devi (2015) [4] trên bộ dữ liệu BSE-SENSEX



Hình 3.9: Kết quả so sánh với tác giả Devi (2015) [4] trên bộ dữ liệu CNX- NIFTY

3.3 Demo

Demo của bài toán chạy trên tập dữ liệu chứng khoán Apple từ ngày 31/03/2010 đến ngày 09/12/2014. Dữ liệu này có 1183 dòng dữ liệu và 6 cột gồm ngày tháng, giá mở cửa, giá đóng cửa, giá cao nhất, giá thấp nhất, khối lượng giao dịch.

Đầu tiên tập dữ liệu được loại bỏ cột ngày tháng, sau đó được làm mịn dữ liệu bằng Exponential smoothing với $\alpha = 0.9$

	Open	High	Low	Close	Volume
0	33.641430	33.801430	33.494286	33.571430	107664900
1	33.915714	34.104286	33.250000	33.709999	150786300
2	33.568573	34.072857	33.538570	34.070000	171126900
3	34.028572	34.320000	33.857143	34.220001	111754300
4	34.221428	34.560001	34.094284	34.371429	157125500

Hình 3.10: Bảng dữ liệu trước khi Smoothing

	Open	High	Low	Close	Volume
0	33.641430	33.801430	33.494286	33.571430	1.076649e+08
1	33.890779	34.076754	33.272208	33.697402	1.468662e+08
2	33.600503	34.073243	33.512174	34.033076	1.687227e+08
3	33.985804	34.295347	33.822677	34.201325	1.174460e+08
4	34.197868	34.533538	34.067126	34.354420	1.531579e+08

Hình 3.11: Bảng dữ liệu sau khi Smoothing

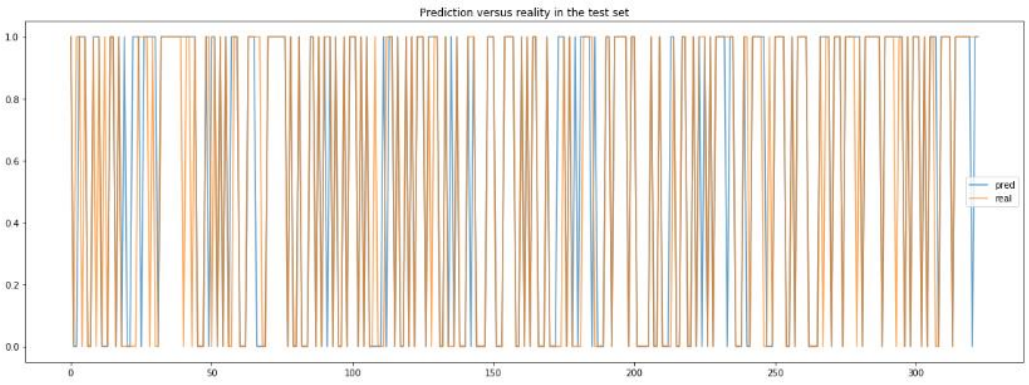
Sau bước Smoothing các thuộc tính RSI, Stochastic Oscillator, … được tính toán.

```
def feature_extraction(data):
    for x in [5, 14, 26, 44, 66]:
        data = ta.relative_strength_index(data, n=x)
        data = ta.rate_of_change(data, n=x)
        data = ta.on_balance_volume(data, n=x)

    #%K
    data['SO%k'] = (data['Close'] - data['Low']) / (data['High'] - data['Low'])

    #Williams %R
    data['ema14'] = data['Close'] / data['Close'].ewm(14).mean()


    data = ta.macd(data, n_fast=12, n_slow=26)
```

Chia tập dữ liệu để train và test.

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

Sau đó train mô hình Random Forest bằng thư viện có sẵn của sklearn và thể hiện kết quả dự đoán với tập testing trên biểu đồ.



Hình 3.12: Biểu đồ kết quả dự đoán và thực tế

TÀI LIỆU THAM KHẢO

[1] Khaidem, L., Saha, S., and Dey, S. R. (2016). Predicting the direction of stock market prices using random forest.

[2] Yuqing Dai, Yuning Zhang (2013). Machine Learning in Stock Price Trend Forecasting. Stanford University.

[3] Xinjie (2014). Stock Trend Prediction With Technical Indicators using SVM. Stanford University.

[4] Ms. K. Nirmala Devi, Dr.V.Murali Bhaskaran, G. Prem Kumar (2015). Cuckoo Optimized SVM for Stock Market Prediction.IEEE Sponsored 2nd International Conference on Innovations in Information, Embedded and Communication systems (ICJJECS)2015