ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC & KĨ THUẬT MÁY TÍNH

LUẬN VĂN TỐT NGHIỆP

Xây Dựng Công Cụ Hỗ Trợ Dự Đoán Giá Trị BITCOIN Bằng Machine Learning

Hội đồng: Mạng và Hệ Thống Máy Tính

Người thực hiện:

Phan Sơn Tự 51204436

Giáo viên hướng dẫn:

TS. Nguyễn Đức Thái

TP. Hồ Chí Minh, ngày 19 tháng 12 năm 2016

Lời cam kết

Tôi tên Phan Sơn Tự - 51204436, hiện đang là sinh viên khoa Khoa Học và Kĩ Thuật Máy Tính, Đại học Bách Khoa TP.HCM. Tôi xin cam kết báo cáo luận văn tốt nghiệp với đề tài "Ứng Dụng Machine Learning Vào Bài Toán Dự Đoán Xu Hướng Giá Trị BITCOIN" là công trình nghiên cứu độc lập, tự tìm hiểu của bản thân, không sao chép bất kì công trình nghiên cứu nào.

Đề tài được thực hiện cho mục đích tìm hiểu và nghiên cứu ở bậc đại học.

Tất cả những tài liệu tham khảo được ghi trong báo cáo đều được trích dẫn rõ ràng từ các nguồn đáng tin cậy và từ một số bài báo khoa học.

Tất cả số liệu trong bài báo cáo đều được thực hiện một cách trung thực, không gian dối, không sao chép từ bất kì nguồn nào.

Các công cụ hỗ trợ cho việc thực hiện giải thuật, đo đạt số liệu đều là mã nguồn mở và tập dữ liệu được cung cấp hoàn toàn công khai của chủ nhân, tổ chức sở hữu.

Hình ảnh trong bài báo cáo đều được trích dẫn nguồn gốc rõ ràng.

Lời cảm ơn

Lời đầu tiên, xin cám ơn mẹ Hoa yêu quý của tôi, người đã chăm lo cuộc sống đầy đủ cho tôi trong suốt quãng đời sinh viên, giúp tôi an tâm trong học tập, nghiên cứu. Và thật lòng, tôi không thể có ngày hôm nay nếu bên cạnh tôi không phải là mẹ.

Sau đó, tôi xin được phép gửi lời cảm ơn sâu sắc đến TS. Nguyễn Đức Thái, người thầy đã giúp đỡ tôi trong thời gian thực hiện đề tài. Không chỉ đơn thuần là một giảng viên truyền thụ kiến thức, cũng không đơn thuần là một giáo viên hướng dẫn luận văn, thầy đối với tôi còn nhiều hơn thế. Thầy là một trong số ít những người đã truyền cảm hứng cho tôi trong lĩnh vực Machine Learning và đó là còn đường tôi đang chọn. Làm việc trong một khoảng thời gian với thầy, nhờ những sự chỉ bảo, định hướng tận tình của thầy cũng như không ngại ngần đưa ra những khuyết điểm đã giúp tôi hiểu rõ bản thân và ngày càng hoàn thiện mình hơn. Thầy hết lòng vì sinh viên đó là điều tôi thán phục ở thầy.

Cuối cùng, tôi xin gửi lời cảm ơn đến những người thầy, người cô đã và đang công tác tại mái trường Đại học Bách Khoa TP. Hồ Chí Minh và đặc biệt là khoa Khoa Học và Kĩ Thuật Máy Tính, chúc thầy cô sức khỏe dồi dào, tiếp tục công tác giảng dạy đào tạo để cho ra những thế hệ kỹ sư có chuyên môn giỏi, đạo đức tốt, góp phần xây dựng một xã hội vững mạnh.

Thành phố Hồ Chí Minh, ngày 19 tháng 12 năm 2016
Phan Sơn Tự

Lời giới thiệu

"Rủi ro càng cao, lợi ích càng nhiều" đó là một trong những câu nói thường được nghe trong môi trường kinh tế, điều này đa số đúng với các hành vi đầu tư kinh tế. Người đầu tư giỏi là người đầu tư có khả năng đoán biết rủi ro từ đó giảm thiểu rủi ro nhưng vẫn gia tăng lợi nhuận, để làm được điều này người đầu tư cần có kiến thức chuyên sâu về kinh tế và kinh nhgiệm, trong đó kinh nghiệm chiếm một vị trí rất quan trọng.

Về một lĩnh vực khác, ngành Công nghệ thông tin đang trở thành một ngành không thể thiếu đối mọi lĩnh vực, nó làm thay đổi phương thức lao động, tạo ra các giá trị hoàn toàn mới, thúc đẩy các lĩnh vực khác cực kỳ mạnh mẽ. Và Kinh tế cũng không nằm ngoài tác động đó. Cũng vì vậy mà lĩnh vực Business Intelligence được sinh ra, đây là một sản phẩm của quá trình sử dụng chất xám Công nghệ thông tin để giải quyết thông minh các vấn đề kinh tế.

Business Intelligence là một bức tranh rộng lớn, riêng trong phạm vi luận văn này, tác giả xin trình bày một đề tài cụ thể, đó là sử dụng Machine Learning để giải quyết bài toán giảm thiểu rủi ro trong đầu tư BITCOIN.

Mục lục

Danh sách hình vẽ

Danh sách bảng

Danh mục chữ viết tắt

MNN	Multilayer Neural Network
KNN	K-Nearest Neighbors
LR	Logistic Regression
SVM	Support Vector Machine
GDA	Gaussian Discriminant Analysis
QDA	Quadratic Discriminant Analysis
BTC	BITCOIN
ROC	Rate of Change
SO	Stochastic Oscillator
RDP	Relative Difference Percentage
UI	User Interface

Chương 1

Giới thiệu đề tài

1.1 Tính cấp thiết của đề tài

BITCOIN - một loại tiền mã hóa (hay tiền điện tử) được xuất hiện lần đầu tiên vào năm 2009 bởi Satoshi Nakamoto [1], với những đặc tính ưu việt hơn cả tiền tệ truyền thống hiện nay khiến cho sự tăng lên nhanh chóng về giá trị. Nhận thấy được sức mạnh của tiền mã hóa có thể sẽ là tương lai của kinh tế và chính trị nên việc hiểu rõ cũng như đầu tư vào BITCOIN là việc đáng để suy ngẫm.

Hiển nhiên, đối với nước ta BITCOIN là rất mới và việc đầu tư là hết sức rủi ro khi không có nền tảng kiến thức và kinh nghiệm đầu tư. Nhận thấy vấn đề này, bản thân đã đặt ra vấn đề"Tại sao không tạo ra một công cụ để cho nhà đầu tư có thể dựa vào như một yếu tố tham khảo tin cậy".

Đồng thời, trong lĩnh vực Công nghệ thông tin nói riêng, Machine Learning đang là nền tảng cho hàng loạt các sản phẩm công nghệ mang tính dự đoán thông minh, ngoài ra còn ứng dụng trong các lĩnh vực về trí thông minh nhân tạo, xử lý ngôn ngữ tự nhiên... và điều đó đang đi đúng với mục tiêu của vấn đề được đưa ra trong phạm vi luận văn này.

1.2 Đặc tả đề tài

Trên một sàn giao dịch tiền mã hóa điển hình, quá trình mua bán BITCOIN được chia ra thành các giai đoạn thời gian và được gọi là phiên giao dịch. Một phiên giao dịch được diễn tả bởi các giá trị điển hình như sau:

- Giá mở phiên: giá bán (mua) BITCOIN của (các) giao dịch ngay tại thời điểm mở phiên.
- Giá đóng phiên: giá bán (mua) BITCOIN của (các) giao dịch tại thời điểm kết thúc phiên.

- Giá cao nhất: giá bán (mua) BITCOIN cao nhất của giao dịch trong khoảng thời gian mở phiên đến kết thúc phiên.
- Giá thấp nhất: giá bán (mua) BITCOIN thấp nhất của giao dịch trong khoảng thời gian mở phiên đến kết thúc phiên.

Thời gian của một phiên giao dịch thường được chọn là 5 phút, 30 phút, 1 tiếng, 2 tiếng, 4 tiếng hoặc 1 ngày, ... Trong phạm vi luận văn chúng ta chọn thời gian một phiên giao dịch là 30 phút. Ở đây, bài toán là đi dự đoán giá trị BITCOIN trong phiên tiếp theo sẽ tăng hay giảm so với phiên hiện tại. Cụ thể, gọi n là phiên hiện tại và n(close) là giá đóng phiên hiện tại, n+1 là phiên tiếp theo và n+1(close) là giá đóng phiên tiếp theo. Nếu n+1(close) $>$ n(close) thì giá tăng (Up), ngược lại thì giá giảm (Down).

Sau khi cụ thể được yêu cầu bài toán, ta sẽ đi đặc tả hướng tiếp cận giải quyết vấn đề. Machine Learning là lựa chọn của luận văn này, cụ thể phương pháp giải quyết sẽ sử dụng giải thuật phân lớp để dự đoán nhãn của phiên giao dịch sẽ là Up hay Down.

1.3 Mục tiêu của đề tài

Vấn đề cơ bản của việc đầu tư là lợi nhuận, bám sát với mục tiêu này phương hướng đề ra sẽ đi giải quyết bài toán cụ thể như sau:

Sử dụng USD để mua/bán BITCOIN, với mỗi phiên giao dịch là 30 phút, chúng ta sẽ đi dự đoán giá trị BITCOIN trong phiên tiếp theo sẽ tăng hay giảm - bài toán phân lớp trong Machine Learning.

Để thực hiện được điều đó chúng ta cần vạch ra những bước đi cụ thể để hiện thực mục tiêu:

- Thu thập, xử lí dữ liệu BITCOIN.
- Áp dụng các giải thuật phân lớp vào tập dữ liệu có được.
- Đánh giá trên lý thuyết hệ thống.
- Vận hành, khảo sát và đánh giá hệ thống trên thực tế.
- Xây dựng, hoàn thiện sản phẩm.

Sản phẩm hoàn thiện mà người dùng được sử dụng sẽ là một Ứng dụng nền Web cung cấp các thông tin, quan điểm để tham khảo cho việc đầu tư.

1.4 Phương pháp thực hiện đề tài

Vì bài toán dự đoán về xu hướng giá trị BITCOIN hầu như chưa có bất kì công trình hoặc bài báo nào được công bố công khai (theo tìm hiểu của cá nhân) nên việc phải tham khảo các hướng giải quyết đã từng có là bất khả thi. Thay vào đó chúng ta sẽ đi tham khảo các bài báo, công trình có mức độ liên quan khá cao như dự đoán xu hướng giá vàng và giá cổ phiếu - các tài liệu này được dẫn tại phần tài liệu tham khảo. Từ những kinh nghiệm của các bài báo, bản thân sẽ đúc kết một vài phương pháp tổng quát, từ đó áp dụng ngược trở lại cho vấn đề dự đoán xu hướng giá trị BITCOIN.

Đồng thời, ngoài việc tham khảo các công trình liên quan, bản thân còn phải sử dụng chính những kinh nghiệm về khai phá dữ liệu và kiến thức Machine Learning, để áp dụng vào nhằm đem lại kết quả tốt nhất. Việc tìm ra lời giải tốt nhất sẽ tiến hành theo phương pháp so sánh các giải thuật, chúng ta sẽ đi chạy các giải thuật phân lớp khác nhau từ đó đánh giá xem giải thuật nào là tốt hơn và từ đó sẽ tập trung tối ưu cho giải thuật đó.

Sản phẩm hoàn thiện là sản phẩm đã được chạy và khảo nghiệm trên thực tế, vì vậy sau khi xây dựng hoàn chỉnh, hệ thống sẽ được chạy thực tế và đánh giá kết quả trong một khoảng thời gian.

1.5 Bố cục luận văn

Để phục vụ tốt cho việc phát triển sau này, bố cục luận văn sẽ được trình bày theo hướng diễn dịch và được chia thành các phần nhỏ để người đọc có thể nắm bắt nội dung.

Trước hết, chúng ta sẽ đi tìm hiểu qua công trình liên quan nhằm hiểu được công việc chúng ta sẽ làm là gì? Và những hướng giải quyết tổng quát đã được sử dụng ra sao?

Sau đó, phần Nền tảng lý thuyết sẽ trang bị các kiến thức về Đại số, Giải tích và Kinh tế để phục vụ cho việc đọc hiểu nội dung các chương sau, đặc biệt là phục vụ cho quá trình phân tích giải thuật phân lớp trong Machine Learning - cụ thể là sử dụng Multilayer Neural Network để phân lớp.

Cuối cùng, thu thập dữ liệu và khai phá dữ liệu cho phù hợp với giải thuật, chạy giải thuật, đánh giá giải thuật và hiện thực sản phẩm.

Chương 2

Những công trình liên quan

Như đã nhắc tới trước đó, các công trình về dự đoán xu hướng giá trị BITCOIN hầu như chưa có hoặc chưa được công khai vì thế mà việc tiếp cận chính xác vấn đề là điều không thể. Thay vào đó chúng ta sẽ đi sử dụng các vấn đề liên quan khác như là dự đoán xu hướng giá trị vàng và dự đoán xu hướng giá trị cổ phiếu. Hai công trình cụ thể được tham khảo trong luận văn là:

1. Predicting Gold Prices - Megan Potoski [2]
2. Machine Learning in Stock Price Trend Forecasting - Yuqing Dai & Yuning Zhang [3]

Ở bài báo thứ nhất - Predicting Gold Prices - đã đề cập đến hai giải thuật phân lớp là SVM và Logistic Regression. Trong đó, vì va vấp với vấn đề mất cân đối trong tập dữ liệu (nhãn positive lớn hơn rất nhiều sao với nhãn negative) nên SVM chỉ được đề cập như một phép so sánh và không được sử dụng trong quá trình giải quyết vấn đề chính. Thay vào đó, Logistic Regression được sử dụng để giải quyết bài toán phân lớp với kết quả khá khả quan.

Logistic Regression (Optimal Feature Set):

Precision	69.90%
Recall	72.31%
Accuracy	69.30%

Bảng 2.1: Bảng đánh giá - Predicting Gold Prices

Ở đây, ta nhận thấy bài báo sử dụng ba tham số đánh giá, chưa vội quan tâm đến ý nghĩa từng tham số ta có thể hiểu rằng các tham số này càng cao thì tương đương với giải thuật càng được xem là tốt. Chi tiết ba tham số này sẽ được nhắc đến ở phần Nền tảng lý thuyết.

Bước qua bài báo thứ hai - Machine Learning in Stock Price Trend Forecasting - nhóm tác giả đã sử dụng bốn giải thuật đó là:

- GDA
- Logistic Regression
- SVM
- QDA

Kết quả đánh giá của 4 giải thuật được nhóm tác giả trình bày:

Model	Logistic Regression	GDA	QDA	SVM
Accuracy	44.5%	46.4%	58.2%	55.2%

Bảng 2.2: Bảng đánh giá - Machine Learning in Stock Price Trend Forecasting

Thật sự kết quả cho ra không tốt so với bài báo thứ nhất và tham số đánh giá chỉ sử dụng một tham số đó là Accuracy, chúng ta không thể dựa vào đó để đánh giá một cách toàn diện về độ hiệu quả của giải thuật. Nhưng riêng trong công trình này, nhóm tác giả có nêu ra Next-Day Model nhằm dự đoán xu hướng giá cổ phiếu trong ngày tiếp theo và có vẻ khá tương đồng với vấn đề đặt ra trong phạm vi luận văn này.

Tổng quan qua hai công trình và tham khảo một số công trình khác, nhận thấy đa số các hướng tiếp cận đều đi theo một phương pháp tổng quát chung, nó bao gồm các bước cơ bản như:

1. Xây dựng không gian vector thuộc tính phù hợp với tính chất bài toán
2. Sử dụng các giải thuật phân lớp điển hình trong Machine Learning như là SVM, Logistic Regression ...
3. Đánh giá giải thuật bằng các tham số Accuracy, Recall, Precision.

Từ những đục kết trên, bản thân nhận thấy các bước trên cũng chính là phương pháp nên dùng để tiếp cận đề tài. Ngoài ra, nhận thấy ở hai công trình trên chưa hề sử dụng một giải thuật rất được phổ biến hiện nay, nó nổi lên như một đại diện của Deep Learning đó là Multilayer Neural Network. Do đó mà luận văn này sẽ sử dụng Multilayer Neural Network như là một giải thuật chính trong quá trình so sánh và đánh giá so với các giải thuật phân lớp khác.

Chương 3

Nền tảng lý thuyết

3.1 Kiến thức nền tảng

Trong phần này chúng ta sẽ đi vào việc ôn tập, nhắc lại một số định nghĩa, kiến thức của các môn nền tảng về Xác suất thống kê, Giải tích và Đại số tuyến tính. Đó là các kiến thức tối quan trọng được sử dụng rất nhiều trong việc xây dựng cũng như tối ưu thuật toán trong Machine Learning [4, 5].

Sau đó, chúng ta sẽ đi đến trình bày các kiến thức vận dụng chính yếu về Machine Learning và Kinh tế, nó được sử dụng để xây dựng nên mô hình phương pháp giải quyết vấn đề phân lớp.

3.1.1 Đại số tuyến tính

- Ma trận A cấp $m \times n$ là một mảng hình chữ nhật gồm m hàng và n cột với các phần tử là số hoặc các đối tượng toán học được biểu diễn như sau:

$$\begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} = (a_{ij}),\ \forall\, i = \overline{1,m},\ j = \overline{1,n}$$

  Trong đó:

  $a_{ij} \in R$: là phần tử thuộc dòng i và cột j của ma trận A.
  m: số dòng của ma trận A.
  n: số cột của ma trận A.

  Ký hiệu $M_{m\times n}$ là tập hợp các ma trận $m \times n$.

- Vector là một ma trận A_{m1} có một cột và m dòng.

$$A = \begin{bmatrix} a_{11} \\ \vdots \\ a_{m1} \end{bmatrix}$$

- Phép cộng và phép trừ hai ma trận. Để cộng và trừ hai ma trận ta thực hiện việc cộng và trừ lần lượt cho từng phần tử tương ứng nhau của hai ma trận. Lưu ý, hai ma trận cần có cùng chiều, nghĩa là số hàng và số cột của hai ma trận là như nhau.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} w & x \\ y & z \end{bmatrix} = \begin{bmatrix} a+w & b+x \\ c+y & d+z \end{bmatrix}$$

- Phép nhân vô hướng là phép nhân giữa một ma trận và một số, ta thực hiện phép nhân vô hướng bằng cách nhân số đó cho tất cả các phần tử trong ma trận.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \times x = \begin{bmatrix} a \times x & b \times x \\ c \times x & d \times x \end{bmatrix}$$

- Nhân hai ma trận: Cho $A \in M_{m \times k}$ và $B \in M_{k \times n}$. Gọi $A_1, A_2, \ldots, A_m$ là m dòng của A; $B^{(1)}, B^{(2)}, \ldots, B^{(n)}$ là n cột của B.
  Ta viết:

$$A = \begin{bmatrix} A_1 \\ \vdots \\ A_m \end{bmatrix}; \quad B = \begin{bmatrix} B^{(1)} & \ldots & B^{(n)} \end{bmatrix}$$

  Với

$$A_i = \begin{bmatrix} a_{i1} & a_{i2} & \ldots & a_{ik} \end{bmatrix}; \quad B^{(j)} = \begin{bmatrix} b_{1j} \\ b_{2j} \\ \vdots \\ b_{kj} \end{bmatrix}$$

  Khi đó $C = A \times B$ gọi là ma trận tích của A với B và phần tử của C_{ij} được xác định như sau: $c_{ij} = A_i \times B^{(j)} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{ik}b_{kj}$
  Một số tính chất của phép nhân hai ma trận:

  - Không có tính giao hoán: $A \times B \neq B \times A$
  - Tính kết hợp: $(A \times B) \times C = A \times (B \times C)$

- Ma trận đảo của A được ký hiệu là A^{-1} với tích của hai ma trận là một ma trận đơn vị.

- Ma trận chuyển vị của ma trận A là ma trận có được từ A bằng cách viết các hàng của ma trận A theo thứ tự thành cột, ký hiệu là A^T.

$$A = \begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix} \Rightarrow A^T = \begin{bmatrix} a & c & e \\ b & d & f \end{bmatrix}$$

- Ma trận đối xứng: Gọi a_{ij} là phần tử của ma trận đối xứng A, thì $\forall\, i, j :\ a_{ij} = a_{ji}$
  Tính chất liên quan: Gọi x là một vector n chiều $(n \times 1)$ và $A = x.x^T$ thì A là một ma trận đối xứng $(n \times n)$

- Ma trận bán định dương (positive semi-definite): Ma trận M $(n \times n)$ được định nghĩa là ma trận bán định dương khi vào chỉ khi với vetor V bất kỳ có n chiều ta luôn có: $V^T MV > 0$

3.1.2 Đại số tuyến tính mở rộng đối với ma trận

- Phép toán trace: Phép toán trace được ký hiệu là "tr" và nó thực hiện phép tính tổng đường chéo của ma trận vuông.

$$trA = \sum_{i=1}^{n} A_{ii}$$

  Các tính chất của phép toán trace với A, B, C là ma trận vuông và a là hằng số:

  1. $trABC = trCAB = trBCA$
  2. $trA = trA^T$
  3. $tr(A + B) = trA + trB$
  4. $tr(a.A) = a.trA$

- Đạo hàm của hằng số theo ma trận (scalar-by-matrix): Với ma trận $A(m \times n)$

$$A = \begin{bmatrix} \frac{\partial f}{\partial A_{11}} & \cdots & \frac{\partial f}{\partial A_{1n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial A_{m1}} & \cdots & \frac{\partial f}{\partial A_{mn}} \end{bmatrix}$$

  Thì

$$\nabla_A f(A) = \begin{bmatrix} \frac{\partial f}{\partial A_{11}} & \cdots & \frac{\partial f}{\partial A_{m1}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial A_{1n}} & \cdots & \frac{\partial f}{\partial A_{mn}} \end{bmatrix}$$

Ví dụ: Cho ma trận A = $\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ và $f(A) = A_{11} + 2A_{12}^2 + 3A_{21}A_{22}$ suy ra

$$\nabla_A f(A) = \begin{bmatrix} 1 & 3A_{22} \\ 4A_{12} & 3A_{21} \end{bmatrix}$$

Các tính chất của đạo hàm ma trận với A, B, C là ma trận vuông nxn và x là vector n chiều:

1. $\nabla_A trAB = B^T$
2. $\nabla_{A^T} f(A) = (\nabla_A f(A))^T$
3. $\nabla_A trABA^TC = CAB + C^TAB^T$
4. $\nabla_A |A| = |A|(A^{-1})^T$
5. $\nabla_x(x^TAx) = x^T(A+A^T)$. Nếu ma trận A đối xứng thì $\nabla_x(x^TAx) = 2x^TA$

- Eigenvalues và Eigenvectors:

$$Av = \lambda v$$

Trong đó
A là ma trận $(n \times n)$
v là vector n chiều $(n \times 1)$
λ là một eigenvalue của A
v là một eigenvector của A
Mệnh đề liên quan:

1. Một ma trận A có thể có nhiều eigenvalue và nhiều eigenvector
2. Cho các eigenvector của A là $v_1, v_2, \ldots, v_n$ không phụ thuộc tuyến tính vào nhau, và $\lambda_1, \lambda_2, \ldots, \lambda_n$ là các eigenvalue tương ứng. Khi đó ta có:

$$A = PDP^{-1}$$

Với:

$$P = \begin{bmatrix} v_1 & v_2 & \ldots & v_n \end{bmatrix}$$

$$D = \begin{bmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \lambda_n \end{bmatrix}$$

- Ma trận trực giao
Một ma trận A được gọi là trực giao nếu tích của ma trận A và chuyển vị của nó là một ma trận đơn vị. Ví dụ

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{3}{5} & -\frac{4}{5} \\ 0 & \frac{4}{5} & \frac{3}{5} \end{bmatrix}$$

$$AA^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{3}{5} & -\frac{4}{5} \\ 0 & \frac{4}{5} & \frac{3}{5} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{3}{5} & \frac{4}{5} \\ 0 & -\frac{4}{5} & \frac{3}{5} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Ma trận đường chéo
  Ma trận đường chéo là ma trận có các giá trị $a_{ij} = 0$ nếu $i \neq j$ và các giá trị còn lại của ma trận là khác 0

$$\begin{bmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{nn} \end{bmatrix}$$

3.1.3 Xác suất thống kê

- Xác suất có điều kiện: Cho hai biến cố A và B. Ta gọi xác suất của biến cố A khi biến cố B đã xảy ra là xác suất của A với điều kiện B, ký hiệu là P(A|B).

- Công thức xác suất đầy đủ: Cho $A_1, A_2, \dots, A_m$ là nhóm đầy đủ các biến cố, với mọi biến cố F ta có:

$$P(F) = P(A_1).P(F|A_1) + P(A_2).P(F|A_2) + \cdots + P(A_n).P(F|A_n)$$

- Công thức Bayes: Cho $A_1, A_2, \dots, A_m$ là nhóm đầy đủ các biến cố, với mỗi $k(k = \overline{1,n})$, ta có:

$$P(A_k|F) = \frac{P(A_k).P(F|A_k)}{P(F)} = \frac{P(A_k).P(F|A_k)}{\sum_{i=1}^{n} P(A_i).P(F|A_i)}$$

- Kỳ vọng: Cho X là đại lượng ngẫu nhiên rời rạc có các bảng phân phối xác suất.

X	X_1	X_2	...	X_n	...
P	P_1	P_2	...	P_n	...

Bảng 3.1: Bảng phân phối xác suất

Khi đó ta gọi kỳ vọng của X là số:

$$E(X) = x_1p_1 + x_2p_2 + \cdots + x_np_n + \cdots = \sum_{n=1}^{\infty} x_np_n$$

Nếu X là đại lượng ngẫu nhiên liên tục có hàm mật độ f(x) thì kỳ vọng của X là:

$$E(X) = \int_{-\infty}^{+\infty} xf(x)dx$$

- Phương sai: Cho X là một đại lượng ngẫu nhiên có kỳ vọng E(X). Khi đó ta ký hiệu phương sai của X là D(X):

$$D(X) = E[(X - E(X))^2]$$

- Phân phối nhị thức: Đại lượng ngẫu nhiên rời rạc $X = 0, 1, 2, \ldots, n$ gọi là có phân phối nhị thức nếu tồn tại số $p \in (0, 1)$ sao cho:

$$p_k = P(X = k) = C_n^k p^k q^{n-k},\ q = 1 - p,\ k = \overline{0, n}$$

  Ta ký hiệu: $X \sim B(n, p)$

- Phân phối chuẩn: Đại lượng ngẫu nhiên X gọi là có phân phối chuẩn nếu hàm mật độ của X có dạng:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-a)^2}{2\sigma^2}},\ \sigma > 0$$

  Ta ký hiệu: $X \sim \mathcal{N}(a, \sigma^2)$

- Phân phối chuẩn tắc: Đại lượng : $X \sim \mathcal{N}(0, 1)$ gọi là có phân phối chuẩn tắc. Nếu X có phân phối chuẩn tắc thì hàm mật độ của X là:

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

- Phân phối Gaussian: Phân phối chuẩn với n chiều còn được gọi với tên khác là phân phối Gaussian, nó được tổng quát hóa từ phân phối chuẩn của số thực lên cho vector nhiều chiều. Phân phối này được tham số hóa bằng hai đại lượng: một vector trung bình $\mu \in R^n$ và một ma trận tương quan (Covariance matrix) $\Sigma \in R^{n \times n}$ với $\Sigma \geq 0$ là một ma trận đối xứng (symmetric) và bán định dương (positive semi-definite). Ký hiệu: $\mathcal{N}(\mu, \Sigma)$.
  Ta định nghĩa, một vector x có quy luật phân phối chuẩn $\mathcal{N}(\mu, \Sigma)$ khi và chỉ khi hàm mật độ của X được biểu diễn dưới dạng:

$$p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)\right)$$

  Từ đó ta suy ra kỳ vọng của vector X chính là μ:

$$E[X] = \int_x xp(x; \mu, \Sigma)dx = \mu$$

  Tổng quát hóa Covariance của một biến giá trị thực ngẫu nhiên ta có Covariance của một biến giá trị vector ngẫu nhiên và được định nghĩa:

$$Cov(Z) = E[(Z - E[Z])(Z - E[Z]^T)] = E[Z.Z^T] - (E[Z])(E[Z])^T$$

  Mặc khác, nếu $X \sim \mathcal{N}(\mu, \Sigma)$ thì:

$$Cov(X) = \Sigma$$

- Relative Difference Percentage: Cho bộ số $\{x_1, x_2, ..., x_n\}$ khi đó

$$RDP_i(n) = \frac{x_n - x_{n-i}}{x_{n-i}}$$

3.2 Kinh tế

Vì luận văn này đang giải quyết một bài toán về kinh tế nên để hiểu được công việc, ta cần nắm được các ý niệm cơ bản về kinh tế.

3.2.1 Phiên giao dịch và các giá trị cơ bản

Gọi T là một mốc thời gian bất kỳ, P là khoảng thời gian được chọn là một phiên giao dịch. Ta có thể nói một cách đơn giản là phiên giao dịch được mở tại thời điểm T và được kết thúc tại thời điểm T + P.

Cụ thể, giả sử chọn mốc mở phiên là 9:00 am và phiên giao dịch có thời hạn là 30 phút, điều đó có nghĩa là kết thúc phiên giao dịch sẽ là 9:30 am.

Ngoài ra:

- Giá mở phiên: là giá bán của một giao dịch gần nhất sau thời điểm T. Ví dụ tại thời điểm 9:01 am có một giao dịch bán 1 BTC là $779 và trong khoảng thời gian 9:00 am đến 9:01 am không hề có bất kỳ giao dịch nào khác ngoại trừ giao dịch này, thì ta có thể nói giá mở phiên sẽ là $779.

- Giá đóng phiên: là giá bán của một giao dịch gần nhất trước thời điểm T + P.

- Giá phiên cao nhất: là giá bán cao nhất của một giao dịch trong khoảng thời gian diễn ra phiên giao dịch, cụ thể là từ thời điểm T đến thời điểm T + P. Ví dụ, trong khoảng thời gian 9:00 am (thời điểm mở phiên) đến thời gian 9:30 am (thời điểm đóng phiên) có một giao dịch BTC với giá là $801 và là giao dịch có giá trị cao nhất. Vậy ta có thể nó giá phiên cao nhất là $801.

- Giá phiên thấp nhất: là giá bán thấp nhất của một giao dịch trong khoảng thời gian diễn ra phiên giao dịch, cụ thể là từ thời điểm T đến thời điểm T + P.

- Lượng giao dịch: số lượng BTC chênh lệch giữa giá phiên cao nhất và giá phiên thấp nhất của một phiên giao dịch.

- Trung bình giao dịch: giá trị USD trung bình của tất cả các giao dịch diễn ra trong khoảng thời gian một phiên giao dịch.

3.2.2 Rate of Change

Đại lượng đo sự khác nhau của giá tại phiên thứ x so với n phiên trước đó. Giá sử $P(x)$ là giá của phiên thứ x thì:

$$ROC_n(x) = \frac{P(x) - P(x-n)}{P(x-n)}$$

Nếu $ROC > 0$ thì giá thị trường đang có xu hướng đi lên (tăng giá). Ngược lại, với $ROC < 0$ thì giá thị trường đang có xu hướng giảm xuống.

3.2.3 Stochastic Oscillator

Đại lượng dùng để đo xu hướng mua/bán của thị trường tại thời điểm phiên x thông qua n phiên trước đó. Giả sử:

L_n = giá phiên thấp nhất trong n phiên
H_n = giá phiên cao nhất trong n phiên
$P(x)$ = giá của ngày x

$$\%K = \frac{P(x) - L_n}{H_n - L_n}$$

Nếu $\%K$ nhỏ hơn 20 thì thị trường đang có xu hướng mua vào và nếu lớn hơn 80 thì thị trường đang có xu hướng bán ra.

3.3 Machine Learning

3.3.1 Khái niệm cơ bản

3.3.1.1 Machine Learning

Machine Learning có hai cách định nghĩa chính và đang được chấp nhận phổ biến:

- Theo Arthur Samuel: *"Là một lĩnh vực nghiên cứu mà nó cung cấp cho máy tính khả năng học hỏi mà không cần lập trình một cách tường minh."*
- Theo Tom Mitchell: *"Một chương trình máy tính được chấp nhận là học hỏi được kinh nghiệm E bằng cách thực hiện một vài tác vụ T theo phép đo hiệu năng P, nếu và chỉ nếu việc thực thi các tác vụ trong T được đo bởi phép đo P đem lại kết quả là kinh nghiệm E được cải thiện."*

3.3.1.2 Supervised Learning - Học có giám sát

Chúng ta được cho một tập dữ liệu đã biết với các input và output tương ứng nhau. Ý tưởng là chúng ta sẽ đi tìm mối quan hệ giữa input và output đó chính là Supervised Learning.

Vấn đề của Supervised Learning được phân loại thành hai vấn đề chính là "Regression" và "Classification". Trong vấn đề "Regression", chúng ta sẽ cố gắng dự đoán kết quả output tiếp theo một cách liên tục, nghĩa là chúng ta đi tìm ra một hàm đầu ra liên tục tổng quát với biến là các thuộc tính đầu vào. Còn với vấn đề "Classification", chúng ta thay vì cố gắng dự đoán kết quả liên tục thì ta sẽ đi dự đoán chúng theo hướng rời rạc, hiểu theo một cách khác là chúng ta đi tìm một phép phân loại rời rạc cho output với các biến input.

3.3.1.3 Unsupervised Learning - Học không giám sát

Học không giám sát cho phép chúng ta tiếp cận các vấn đề mà ta chưa hề hoặc biết rất ít kết quả của chúng ta sẽ trông như thế nào. Chúng ta có thể xây dựng cấu trúc của dữ liệu mà không cần thiết phải biết ảnh hưởng của các biến đó.

Chúng ta thực hiện việc này dựa trên ý tưởng gom cụm dữ liệu bằng cách xem xét mối quan hệ giữa các thuộc tính của dữ liệu. Các hướng tiếp cận dựa trên nhưng phương pháp như vậy thường được gọi là "Clustering".

3.3.2 Thông số đánh giá

Có ba tham số cơ bản dùng để xem xét và đánh giá giải thuật trong Machine Learning. Gọi:

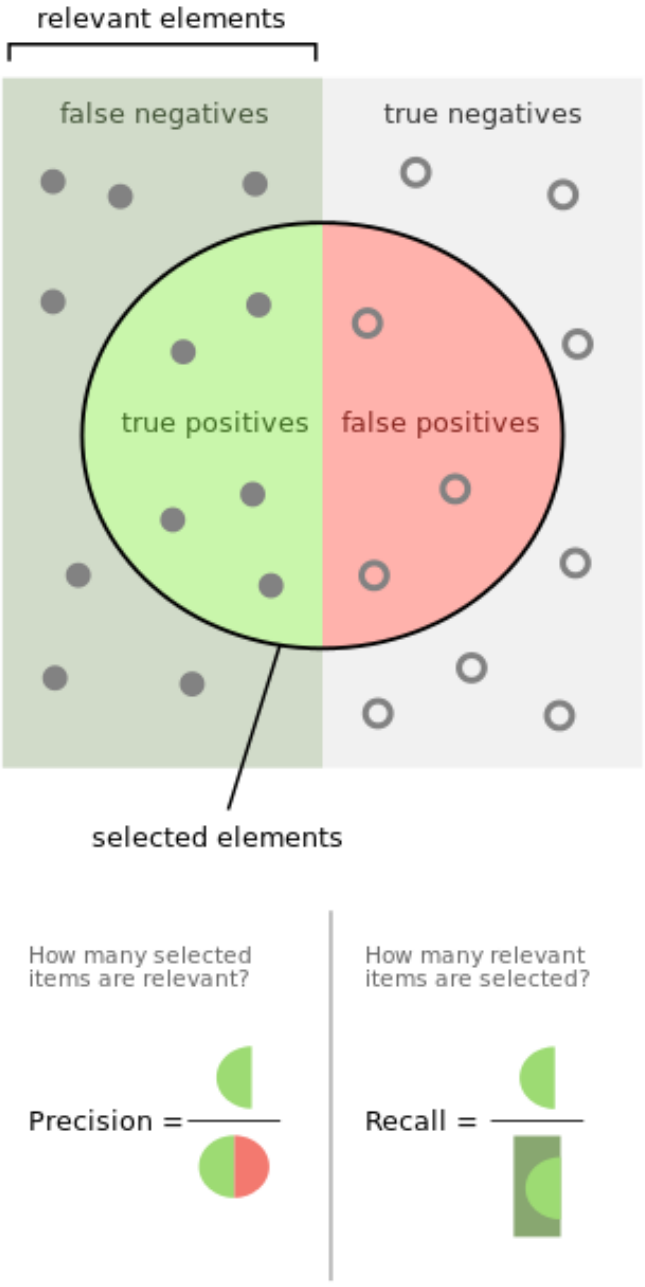
- True positive là TP
- False positive là FP
- True negative là TN
- False negative là FN

Thì:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$



Hình 3.1: Validation Parameters

3.3.3 Neural Network - Deep Learning

Deep Learning nói chung và Neural Network nói riêng là hai phạm trù xuất hiện không lâu đối với Machine Learning, đại diện cho hướng tiếp cận gần với cái nhìn thực tế, học nhiều cấp và học từ bản chất dữ liệu. Deep Learning thường giải quyết rất tốt với các loại dữ liệu mang tính "con người" như hình ảnh, âm thanh ... [6]

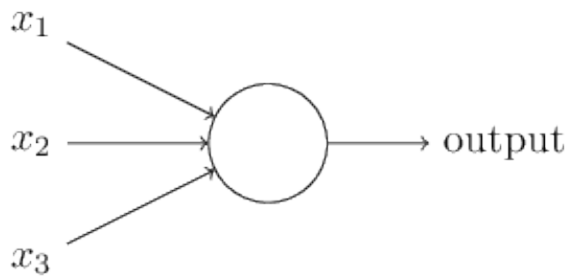
3.3.3.1 Ý tưởng giải thuật

Bộ não con người là một trong những phát minh vĩ đại nhất của tự nhiên, nó có thể giải quyết các bài toán mà đối với máy tính là cực kỳ phức tạp chỉ trong vài giây hoặc kể cả là vài phần giây, các khả năng phán đoán, học hỏi, tích lũy kinh nghiệm đó chính là điều tuyệt diệu của bộ não con người. Và các nhà khoa học khao khát một thứ gì đó trong giới máy tính có khả năng như vậy.

Dựa trên ý tưởng kết cấu của bộ não gồm hàng tỷ neural liên kết lại với nhau, mỗi neural chỉ đưa ra một tín hiệu hết sức đơn giản, nhưng khi hàng tỷ neural liên kết hình thành nên một hệ thống phức tạp thì từ đó có khả năng giải quyết các vấn đề phức tạp. Các nhà khoa học máy tính, đã cố gắng định nghĩa một neural đơn lẻ trong phạm trù máy tính và được gọi là perceptron, từ đó kết nối lại với nhau để tạo nên một hệ thống hữu hạn các perceptron có khả năng giả lập một bộ não người - mạng neural nhân tạo.

3.3.3.2 Cấu trúc một Perceptron

Một perceptron sẽ có các input $x_1, x_2, ...$ và output sẽ là một giá trị nhị phân.



Hình 3.2: Multilayer Neural Network

Một ví dụ đơn giản dựa vào hình trên, ta thấy perceptron này có 3 input là x_1, x_2, x_3 , giả sử đi kèm với mỗi input sẽ có một giá trị trọng số w_1, w_2, w_3. Output được định nghĩa là 0 và 1, nhận giá trị 0 khi $\sum_j w_j x_j$ nhỏ hơn giá trị ngưỡng và 1 khi lớn gơn giá trị ngưỡng.

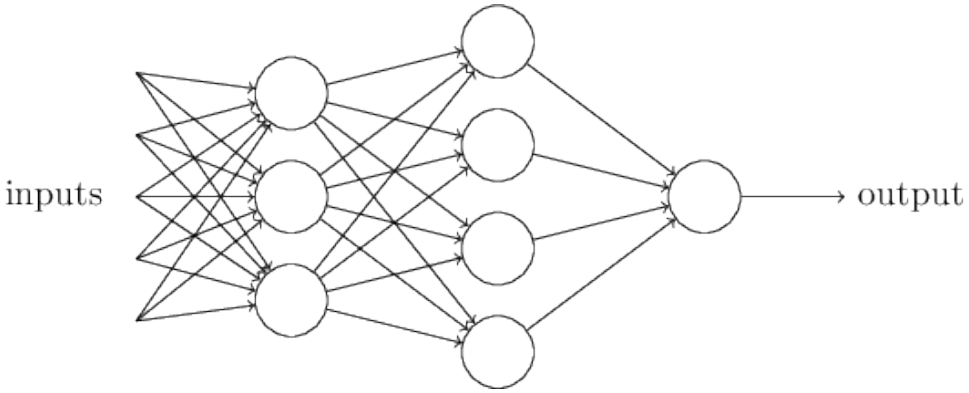
Biểu diễn đại số:

$$output = \begin{cases} 1 & if \sum_j w_j x_j > threshold \\ \\ 0 & if \sum_j w_j x_j \leq threshold \end{cases}$$

Các hàm số như trên được gọi là activation function, có nhiều loại activation function khác nhau như: $sigmoid, tang...$

3.3.3.3 Multilayer Neural Network

Hiển nhiên, một perceptron không thể mô phỏng nên được một bộ não người, để có thể đưa ra một quyết định tương tự như bộ não người các perceptron này cần được kết nối với nhau thành một mạng lưới - Multilayer Neural Network.



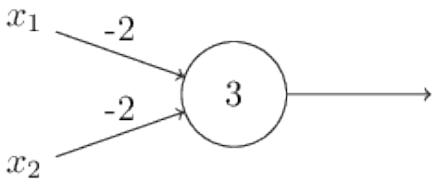
Hình 3.3: Perceptron

Multilayer Neural Network được cấu thành bằng cách sắp xếp các perceptron thành từng lớp. Các perceptron ở mỗi lớp sẽ kết nối với tất cả các perceptron ở các lớp liền kề, cột những perceptron đầu tiên được gọi là input layer, chúng có chức năng tiếp nhận các input để cho ra các output. Các output ở lớp trước sẽ chính là input cho các perceptron ở lớp tiếp theo. Các perceptron ở lớp cuối cùng được gọi là output layer, trong trường hợp này đặc biệt chỉ có duy nhất một perceptron. Còn lại các lớp perceptron khác được gọi là hidden layer.

Giả sử input của perceptron là $x_1, x_2, ...$ tương ứng là đó là các trọng số $w_1, w_2, ...$ Thêm vào đó định nghĩa về bias, ở đây bias là một giá trị đại diện độ lệch của từng perceptron và được ký hiệu $b_1, b_2, ...$ Ta có biểu diễn của activation function:

$$output = \begin{cases} 1 & if \sum_j w_j x_j + b_i > 1 \\ 0 & if \sum_j w_j x_j + b_i \leq 0 \end{cases}$$

Ví dụ:



Hình 3.4: Perceptron with Bias example

Ta có $w_1 = w_2 = -2$ và $b = 3$, khi đó nếu input $x_1 = 1, x_2 = 0$ suy ra $w_1 * x_1 + w_2 * x_2 + b = (-2) * 1 + (-2) * 0 + 3 = 1$, ta có thể chọn $threshold = 0$ vì $1 > 0$ nên $output = 1$.

3.3.3.4 Sigmoid Function - Hàm Sigmoid

Với dạng activation function được định nghĩa ở trên, giá trị của activation function gần như không có giới hạn. Vậy tại sao việc không có giới hạn lại cần được quan tâm. Trong một trường hợp cụ thể, với việc sử dụng activation function như trên có thể dẫn đến trường hợp đầu ra của một perceptron sẽ nhận giá trị rất lớn - giả sử là 1000, những một perceptron khác sẽ nhận giá trị rất bé - giả sử 0.001. Vì thế khi đến lớp tiếp theo thì gần như perceptron cho kết quả đầu ra là giá trị bé sẽ mất đi độ ảnh hưởng và làm mất cân đối cho toàn mạng.

Do đó để giới hạn giá trị của activation function chúng ta sẽ sử dụng hàm sigmoid. Sigmoid function được định nghĩa như sau:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

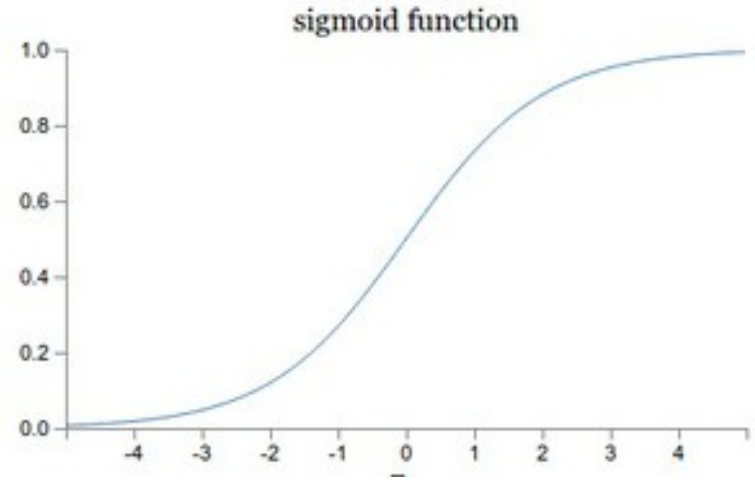
Áp dụng Sigmoid function vào activation function ta có activation function dạng sigmoid và khi đó activation function của chúng ta sẽ có dạng:

$$\frac{1}{1 + exp(-\sum_j w_j x_j - b)}$$

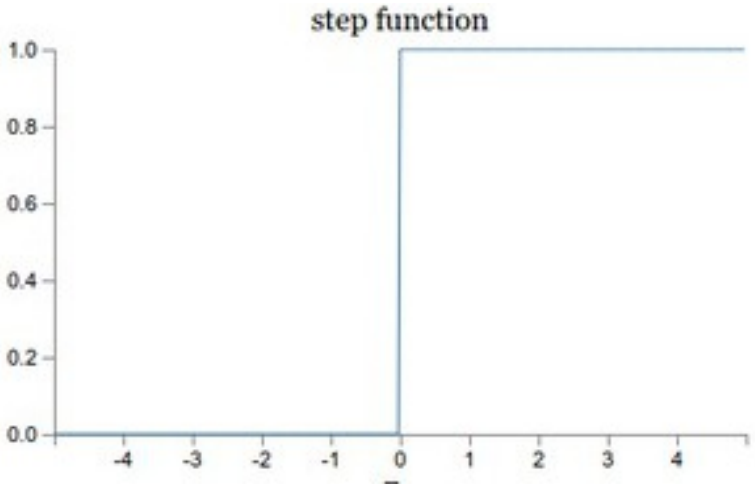
Lúc này ta có một activation function có giá trị được giới hạn trong khoảng từ 0 đến 1. Nhưng chú ý, giá trị của activation function là liên tục, để rời rạc hóa giá trị của activation function ta có thể sử dụng một phương pháp quen thuộc - sử dụng threshold. Điển hình ta chọn threshold = 0.5, nếu lớn hơn

thì activation function sẽ nhận 1 và ngược lại sẽ nhận 0.

Để hiểu rõ hơn tại sao chúng ta quan sát hai đồ thị của sigmoid function và activation function có dạng sigmoid và đã được rời rạc hóa giá trị:



Hình 3.5: Sigmoid Function



Hình 3.6: Step Function

3.3.3.5 Giải thuật Backpropagation

Sau khi đã có xây dựng thành công một mô hình Multilayer Neural Network, công việc cuối cùng là cung cấp khả năng tự học hỏi từ đó để bản thân mạng

có thể tự xây dựng mô hình và đưa ra các quyết định cụ thể.

Cụ thể, khi nhìn lại một Multilayer Neural Network với activation function là sigmoid function thì các tham sô w, b là chưa biết và việc cung cấp khả năng tự học hỏi chính là cung cấp một giải thuật giúp mạng tìm được các tham số w, b với một tập kinh nghiệm - hay tập huấn luyện - x, y cụ thể, trong đó x là input và y là output tương ứng với từng bộ x. Giải thuật backpropagation là một trong nhưng giải thuật chúng ta cần tìm.

Trước tiên chúng ta cần đi qua một số ký hiệu:

- w là vector của các giá trị trọng số

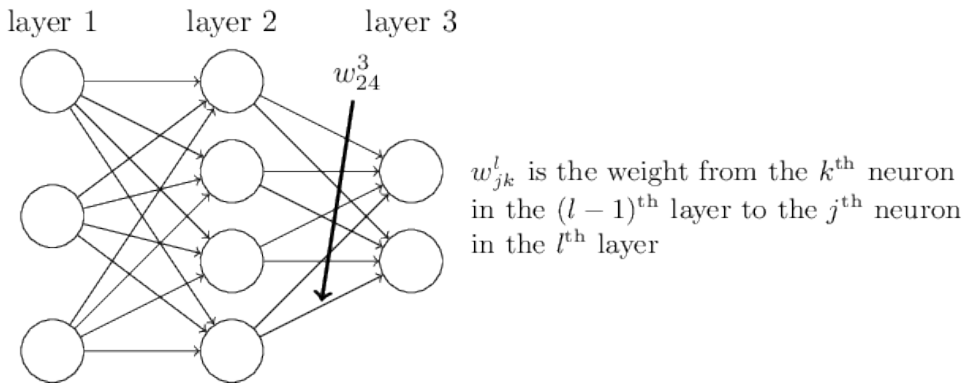
$$w = \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix}$$

- b là vector của các giá trị bias

$$b = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$

- σ là sigmoid function
- $a(\sigma)$ là activation function có dạng sigmoid

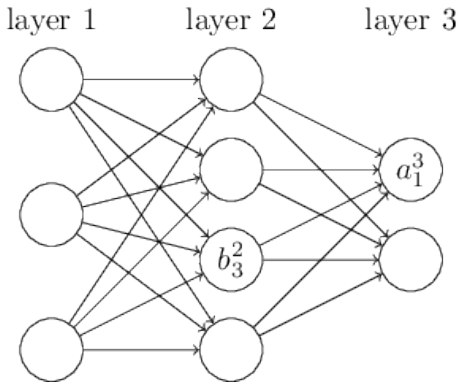
Biểu diễn activation function:



Hình 3.7: Weight Notation example

Ví dụ như hình trên, trọng số xuất phát từ perceptron thứ 4 thuộc layer thứ 2 và kết thúc tại perceptron thứ 2 thuộc layer thứ 3 được ký hiệu là w^3_{24}.

Tương tự như vậy với bias và activation function của perceptron thứ j thuộc layer thứ l của mạng sẽ được kí hiệu thứ tự là b^l_j, a^l_j. Ví dụ, bias của perceptron thứ 3 thuộc layer thứ 2 sẽ là b^2_3 và activation function của perceptron thứ 1 thuộc layer thứ 3 sẽ là a^3_1.



Hình 3.8: Bias Notation example

Lúc này ta có biểu diễn toán học đầy đủ của activation function:

$$a^l_j = \sigma(\sum_k w^l_{jk} a^l_k - 1 + b^l_j)$$

Với ký hiệu vector ta có thể tổng quát phát biểu với dạng:

$$a^l = \sigma(w^l a^{l-1} + b^l)$$

Cost function: Trước khi đi vào hiểu được backpropagation có thể làm gì, chúng ta cần phải biết một định nghĩa cost function. Vậy cost function là gì? Đúng theo tên của hàm, nó dùng để đo lường chi phí của thuật toán. Chi tiết:

$$C = \frac{1}{2n} \sum_x \|y(x) - a^L(x)\|^2$$

Ta có thể thấy, dạng hàm số trên hết sức quen thuộc với định nghĩa độ lệch chuẩn trong xác suất thống kê nhưng đã được biến đổi một chút. Thay vì giá trị kỳ vọng và các điểm xác suất, cost function sử dụng giá trị thực tế y của tập dữ liệu và giá trị $y = a$ là giá trị y tính toán được từ x với w và b. Vậy ta có thể hiểu được, cost function tính toán độ sai lệch của giá trị a so với y kỳ vọng thực tế. Do đó, cost function càng nhỏ thì biểu diễn giá trị của Multilayer Neural Network sẽ càng gần với thực tế.

Để tìm được giá trị cực tiểu cho cost function ta sẽ thực hiện vòng lặp:

$$w_i j^{(l)} := w_i j^{(l)} - \eta \frac{\sigma}{\sigma w_i j^{(l)}} C(w, b)$$

$$b_i^{(l)} := b_i^{(l)} - \eta \frac{\sigma}{\sigma b_i^{(l)}} C(w, b)$$

Trong đó η là learning rate - tỉ lệ học, việc hội tụ về giá trị cực tiểu với tốc độ và độ chính xác phụ thuộc vào tỉ lệ này.

Vậy, đi qua một quá trình tìm hiểu về Multilayer Neural Network, ta có thể hiểu được việc học hỏi kinh nghiệm của mạng cốt lõi vẫn là việc tìm ra bộ w và b tương ứng với x, y của bộ dữ liệu luyện tập, và để tìm ra được w và b ta có thể sử dụng giải thuật backpropagation.

Chương 4

Phân tích và thiết kế hệ thống

4.1 Xây dựng Multilayer Neural Network

4.1.1 Feature Selection - Dữ liệu luyện tập

Một trong những yếu tố hết sức quan trọng trong Machine Learning đó chính Feature. Feature chính là các giá trị thuộc tính đại diện cho tập dữ liệu luyện tập, ví dụ chúng ta có tập dữ liệu về loài chim thì có thể feature chính là các thông số về độ dài sải cánh, màu lông, vùng sinh sống... Một giải thuật có thể học được "kinh nghiệm" nhanh hay chậm, chính xác hay sai lệch phụ thuộc rất nhiều vào yếu tố feature. Vì vậy quá trình khai phá dữ liệu là hết sức cần chú ý.

Tập dữ liệu về các phiên giao dịch BITCOIN được thu thập từ ngày 20/2/2015 đến ngày 29/10/2016 và có tổng cộng 29634 phiên giao dịch.

Gọi S là đại diện cho một phiên giao dịch, các feature được xây dựng như sau:

- 10 feature RDP: $\{\ loop\{RDP_1(S_{i+j})\}_i\ \}_j$ Với $i \in [0:9]$, $j \in [0:29634]$
- 1 feature SO. Với $j \in [0:29625]$:

$$\{\%K_j = \frac{P(j+9) - L_{10}}{H_{10} - L_{10}}\}_j$$

- 1 feature ROC. Với $j \in [9:29634]$:

$$\{ROC_{10}(j) = \frac{P(j) - P(j-9)}{P(j-9)}\}_j$$

Ở đây, ta đã chủ ý chọn mỗi feature vector được hình thành bởi 10 phiên giao dịch. Các giá trị SO và ROC đều được tính trong thời gian là 10 phiên

giao dịch. Sau khi đã có feature, ta cần label để phân lớp tập luện tập. Ở đây đơn giản, nếu giá BITCOIN ở phiên thứ 11 lớn hơn phiên thứ 10 thì label sẽ là 1, ngược lại sẽ là 0. (Phiên 11 chính là phiên thứ 1 của nhóm 10 phiên liền sau nhóm 10 phiên hiện đang xét).

$$label_i = \begin{cases} 1 & if\ P_i(10) > P_{i+1}(1) \\ \\ 0 & if\ P_i(10) \leq P_{i+1}(1) \end{cases}$$

4.1.2 Training - Học giải thuật

Bên cạnh chạy giải thuật Multilayer Neural Network, chúng ta sẽ chạy các giải thuật khác nhằm so sánh và đánh giá giải thuật chính.

Các giải thuật được chọn chạy:

- Multilayer Neural Network - MNN
- Support Vector Machine - SVM
- K-Nearest Neighbors - KNN
- Logistic Regression - LR

Sau khi chạy xong ta ghi nhận kết quả của các giải thuật được đề cập dưới dạng các tham số đánh giá sau:

- Accuracy
- Recall
- Precision

4.1.3 Validation - Đánh giá giải thuật

Để có được kết quả đánh giá, chúng ta sẽ chia tập dữ liệu ra hai phần:

- Training data: chiếm 7/10 tổng số dữ liệu, dùng để chạy trong quá trình học của giải thuật.
- Validation data: chiếm 3/10 tổng số dữ liệu, dùng để chạy trong quá trình đánh giá giải thuật.

	KNN	LR	SVM	MNN
Accuracy	62.93%	66.24%	66.40%	69.86%
Precision	44.69%	18.18%	0%	60.50%
Recall	43.62%	0.15%	0%	29.55%

Bảng 4.1: Bảng đánh giá

Kết quả chạy giải thuật:

Trước tiên theo bảng đánh giá, ta có các giải thuật LR và SVM cho kết quả Accuracy là gần khoảng 66%, nhưng khi nhìn và chi tiết các giá trị Precision và Recall ta nhận thấy các giải thuật này hầu như chỉ dự đoán kết quả là Down cho tất cả trường hợp. Điều này hoàn toàn không có ý nghĩa để dự đoán đầu tư.

Xét đến KNN và MNN, đối với KNN ta có thể thấy giải thuật có xu hướng cân bằng các giá trị Accuracy, Precision và Recall. Nhưng đối với MNN, giải thuật có xu hướng tối ưu hóa Accuracy và Precision. Vậy câu hỏi đặt ra ở đây là kết quả nào có giá trị đầu tư hơn?

Chú ý đến Recall, dựa theo định nghĩa thì Recall có thể hiểu nếu trong thực tế có 10 phiên là Up thì KNN sẽ dự đoán đúng khoảng 4 lần và MNN sẽ dự đoán đúng khoảng 3 lần. Nhìn thoáng qua có vẻ như Recall cao thì sẽ có ý nghĩa trong việc đầu tư hơn, nhưng hay khoan kết luận.

Xét đến Precision, ta có thể hiểu Precision là, với 10 lần dự đoán sẽ có phiên Up thì KNN sẽ đúng khoảng 4 lần và MNN sẽ dự đoán đúng 6 lần. Giả sử, mức độ tin tưởng của chúng ta vào hệ thống là 100%, cứ mỗi lần hệ thống dự đoán có phiên Up thì ta sẽ bỏ tiền đầu tư. Điều đó đồng nghĩa, nếu theo KNN sẽ có 6 lần ta chịu lỗ vì hệ thống dự đoán sai và với MNN thì ta sẽ có 4 lần ta chịu lỗ.

Quay lại với Recall, giá trị này không đo đạt được việc chúng ta sẽ lợi nhuận hoặc thua lỗ ra sao mà thực ra là giá trị đo đạt khả năng tận dụng cơ hội của hệ thống.

Tới lúc này, ta có thể kết luận, bộ giá trị chiếm ưu tiên cao hơn sẽ là Accuracy và Precision. Điều đó cũng có nghĩa là giải thuật Multilayer Neural Network nên là lựa chọn để tiếp cận giải quyết vấn đề này.

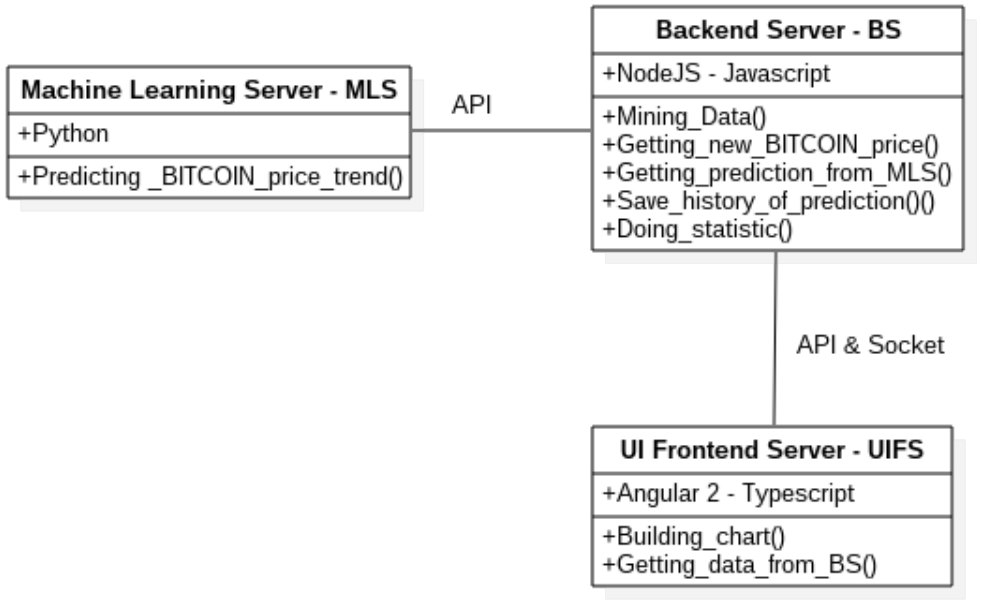
4.2 Xây dựng hệ thống - Web Application

4.2.1 Tổng quan hệ thống

Hệ thống được xem xét và được thiết kế với 3 khối server chức năng:

- Hệ thống Machine Learning server
- Hệ thống Backend server
- Hệ thống UI Frontend server

Các khối hệ thống giao tiếp với nhau bằng API và Socket - đối với các chức năng realtime.



Hình 4.1: System Structure

4.2.2 Hệ thống Machine Learning Server

Đây là hệ thống cốt lõi của của sản phẩm, nó đảm nhiệm khối chức năng dựa vào các tham số được truyền vào để đưa ra giá trị nhãn tương ứng cho bộ tham số đó.

Cụ thể, để đưa ra một dự đoán, hệ thống yêu cầu các tham số đầu vào

phải được xây dựng theo mô tả của Feature Selection - 12 features.

Trong hệ thống Machine Learning Server, được chia nhỏ thành hai phần:

1. Prediction: bao gồm các chức năng đọc mô hình Multilayer Neural Network đã xây dựng, chạy mô hình với tham số truyền vào và lấy các kết quả đầu ra. Kết quả đầu ra có giá trị nhãn Up-Down và xác suất dự đoán.
2. Django: bao gồm các chức năng để hình thành một API server ví dụ như tiếp nhận các yêu cầu thông qua API, phản hồi các yêu cầu...

Vì tính chất hỗ trợ tốt cho Machine Learning nên Python được lựa chọn là ngôn ngữ để phát triển hệ thống này.

4.2.3 Hệ thống Backend Server

Vì bản thân hệ thống Machine Learning Server không có các khối chức năng liên quan đến việc lấy dữ liệu giá BITCOIN cũng như khai phá dữ liệu nên hệ thống Backend Server được xây dựng để thực hiện các chức năng này. Đồng thời, Backend Server còn là cầu nối giữa trải nghiệm người dùng (hệ thống UI Frontend Server) và hệ thống Machine Learning Server.

Để thực hiện được công việc trên, hệ thống bao gồm được xây dựng các chức năng:

1. Cập nhật giá BITCOIN: thông qua các public API được các sàn giao dịch BITCOIN cung cấp, các hàm lấy giá được chạy liên tục để cập nhật giá BITCOIN mới nhất nhằm phục vụ cho quá trình dự đoán.
2. Khai phá dữ liệu: dữ liệu được các hàm cập nhật giá BITCOIN lấy được vẫn còn ở dạng thô, chưa qua xử lý. Khai phá dữ liệu là biến đổi các dữ liệu này về các bộ tham số có ý nghĩa với Machine Learning, các giá trị này mới đích thực dùng để làm đầu vào dự đoán xu hướng giá trị BITCOIN.
3. Giao tiếp với hệ thống Machine Learning Server: truyền tham số đi và nhận kết quả trả về từ hệ thống Machine Learning Server thông qua API.
4. Lưu trữ và thống kê dữ liệu: thực hiện việc lưu trữ dữ liệu, từ đó tạo nên một hệ thống các dữ liệu phục vụ cho việc phân tích, thống kê để

cung cấp cho người dùng đầu cuối. Đó là các thông tin hết sức quý giá phục vụ cho các nhà đầu tư.

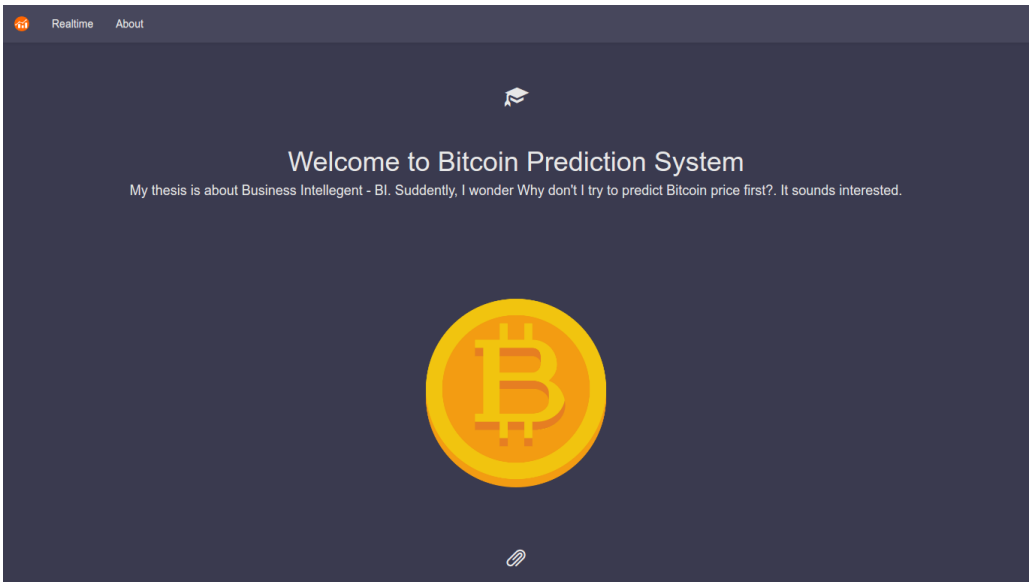
5. Giao tiếp với hệ thống UI Frontend Server: đưa ra những API chức năng nhằm phục vụ cho UI Frontend Server. Ví dụ như: yêu cầu dữ liệu dự đoán, yêu cầu thống kê đúng/sai, yêu cầu dữ liệu giá cho biểu đồ...

Với khả năng xử lý nhanh, được hỗ trợ tốt nên NodeJS được dùng để phát triển hệ thống. Đồng thời, cơ sở dữ liệu của hệ thống là MongoDB vì tính linh hoạt trong cấu trúc dữ liệu và khả năng mở rộng cao.

4.2.4 Hệ thống UI Frontend Server

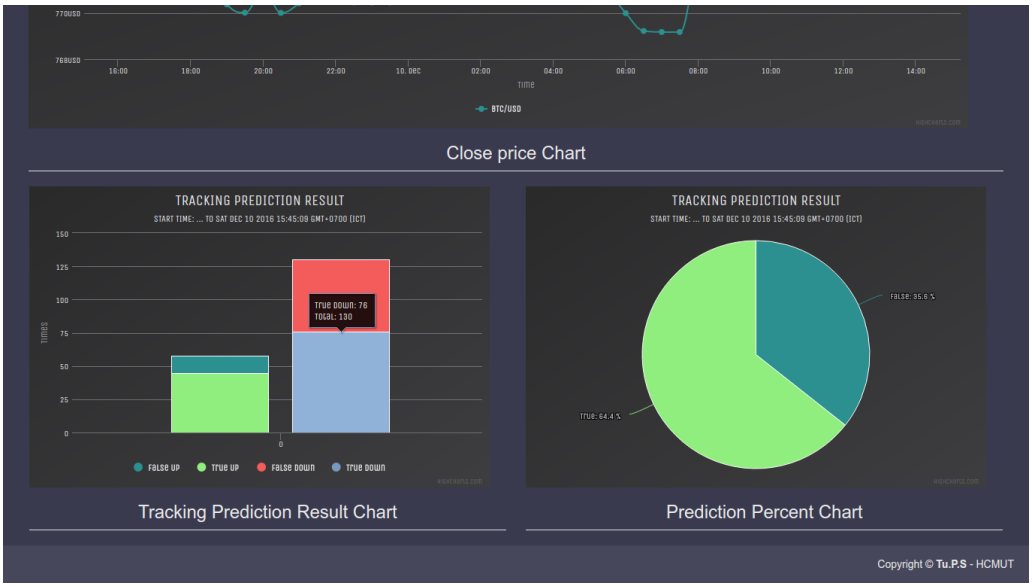
Hệ thống UI Frontend Server là một giao diện người dùng, nó cho phép người dùng có thể tiếp cận với các chức năng của toàn bộ hệ thống một cách dễ dàng. Hệ thống bao gồm nhiều biểu đồ, cũng như tham số cung cấp các thông tin có ý nghĩa đầu tư - dự đoán xu hướng giá trị BITCOIN - đồng thời với đó, là các thông tin về độ tin cậy của hệ thống, các thống kê về lịch sử dự đoán...

Một số hình ảnh về hệ thống thực tế.



Hình 4.2: UI Frontend Server 1

Hình 4.3: UI Frontend Server 2



Hình 4.4: UI Frontend Server 3

Hệ thống UI Frontend Server được xây dựng theo xu hướng one-page, cũng chính vì vậy mà Angular 2 là lựa chọn phù hợp, với khả năng phát triển nhanh, hỗ trợ tốt từ các bên thứ 3.

Chương 5

Kết luận và hướng phát triển

5.1 Kết luận

Kết thúc đề tài, sản phẩm cuối cùng được hoàn thiện là một công cụ nền Web hỗ trợ, cung cấp các thông tin có giá trị tham khảo để đầu tư BITCOIN. Dựa trên các con số lý thuyết, khả năng dự đoán chính xác là rất khả quan và đặc biệt, giải thuật được tối ưu cho phù hợp với góc nhìn của một người đầu tư.

Với không nhiều sai lệch khi so sánh bên cạnh các con số lý thuyết, khi hệ thống được cho chạy thực tế trong vòng 4 ngày liên tiếp (Cụ thể từ 22:30:00 13/11/2016 đến 20:30:00 17/11/2016) đã cho ra kết quả:

Accuracy	Precision	Recall
64.4%	77.6%	45.5%

Bảng 5.1: Bảng đánh giá hệ thống thực tế

Các tham số đánh giá chạy thực tế như vậy, có thể thấy với một lần đầu tư ta có tới hơn 70% là có lợi nhuận. Tuy vậy, bất kỳ một hệ thống cũng vẫn sẽ có những điểm thiếu sót.

Vì giới hạn của thời gian thực hiện đề tài, phạm vi của đề tài cũng được thu hẹp để phù hợp nên vì thế đã bỏ qua một số yếu tố thị trường ảnh hưởng khá lớn đối với hướng giải quyết. Trong lúc này, bản thân có thể nhận ra hai vấn đề:

- Phí giao dịch: ở tất cả các sàn giao dịch, đều có một khoảng phí trung gian từ 0.1% đến 0.3% và phí này được trừ trực tiếp vào các giao dịch. Hướng tiếp cận của đề tài bỏ qua hoàn toàn yếu tố này và có thể hiểu là phí bằng 0%

- Biên độ lợi nhuận và thua lỗ: chúng ta cũng đã bỏ qua yếu tố này, mặc dù dựa theo đánh giá thì số lần đầu tư lợi nhuận sẽ nhiều hơn thua lỗ. Nhưng, chúng ta không thể kết luận việc đầu tư sẽ chắc chắn đem về lợi nhuận. Hãy nói đến một trường hợp xấu, biên độ lợi nhuận chỉ có \$1 cho mỗi lần nhưng biên độ thua lô lại là \$100, tại đây chúng ta có thể thấy là việc đầu tư không hề có lợi.

Việc nhìn nhận được các vấn đề trên không hẳn là điều tồi tệ, mà ngược lại giúp chúng ta có thể hiểu rõ bài toán và đưa ra những hướng phát triển tiếp theo.

5.2 Hướng phát triển

Với các vấn đề còn tồn tại được nêu ra bên trên (Mục 5.1), giai đoạn tiếp theo của đề tài là đi giải quyết vẫn đề tài như hiện giờ nhưng thêm vào đó là yếu tố phí giao dịch. Tuy là một yếu tố nhỏ nhưng nó dẫn đến việc thay đổi hoàn toàn bộ dữ liệu ban đầu, điều này đồng nghĩa toàn bộ hệ thống hiện giờ sẽ không tương thích. Vì thế, cần thực hiện lại quá trình xây dựng giải thuật từ đầu.

Mặc khác, việc chỉ học duy nhất từ tập dữ liệu về giá BITCOIN là không đủ để đưa ra một dự đoán chính xác cao. Ngày nay, mạng xã hội đang phát triển như vũ bão, đây là một kênh thông tin cực kỳ quý giá, chính vì vậy mà hệ thống ở giai đoạn phát triển tiếp theo sự tận dụng tài nguyên này.

Phát triển hệ thống xử lý ngôn ngữ tự nhiên, xây dựng hệ thống lắng nghe các thông tin tài chính, chính trị có ảnh hưởng tới giá trị BITCOIN, phân tích, đánh giá và cho cân bằng với hệ thống học từ dữ liệu giá BITCOIN để cho ra một dự đoán tổng quát và chính xác hơn.

Đồng thời, hệ thống có thể mở rộng ra cho nhiều cryptocurrency khác như Ethereum, Zcash, Monero...

Tài liệu tham khảo

[1] https://vi.wikipedia.org/wiki/Bitcoin. *Wikipedia - BITCOIN*. Trích dẫn vào tháng 11/2016

[2] Megan Potoski. *Predicting Gold Prices*. CS229, Autumn 2013

[3] Yuqing Dai & Yuning Zhang. *Machine Learning in Stock Price Trend Forecasting*. 2013

[4] Andrew Ng. *Lecture Notes – CS229 Machine Learning*. 2012

[5] Andrew Ng. *Section Notes – CS229 Machine Learning*. 2012

[6] Michael Nielsen. *Neural Networks and Deep Learning - Free Online Book*. Jan 2016