

DỰ ĐOÁN GIÁ BITCOIN

Phạm Long Huân
Kỹ Thuật Máy Tính

Đại Học Công nghệ thông tin , Đại học quốc gia TP.HCM
Hồ Chí Minh, Việt Nam
18520788@gm.uit.edu.vn

Hoàng Minh Nghĩa
Kỹ Thuật Máy Tính

Đại Học Công nghệ thông tin, Đại học quốc gia TP.HCM
Hồ Chí Minh, Việt Nam
18521141@gm.uit.edu.vn

Tóm tắt nội dung — Bài toán dự đoán giá của Bitcoin có ý nghĩa quan trọng vì sự biến động về giá trị với biên độ vô cùng lớn của đồng tiền ảo này nên việc dự đoán được giá trị Bitcoin sẽ giúp mang lại lợi ích kinh tế vô cùng lớn. Trong đồ án này, chúng tôi xây dựng hệ thống phân tích các yếu tố ảnh hưởng đến đồng tiền ảo này.

Từ khóa: Bitcoin, Bitcoin price, Bitcoin price prediction, The Future

I. GIỚI THIỆU

Với những ưu điểm nổi bật hơn so với các đồng tiền thông thường, Bitcoin đã đem lại cho nền kinh tế, đời sống, xã hội của các nước. Bitcoin thường là một phương pháp thanh toán không thông qua trung gian, mà được chuyển trực tiếp từ tài khoản của người này sang tài khoản của người khác. Nhờ vậy, Bitcoin đã giảm thiểu được khoản phí giao dịch tương đối cao. Nguyên nhân sâu xa nhất của việc ưu tiên sử dụng Bitcoin để thanh toán xuất phát từ Mỹ, chiến dịch chiếm phố Wall, do người dân không muốn bị lệ thuộc vào giới tài phiệt tài chính, họ không muốn bị mất phí trung gian nữa, mà chỉ muốn trực tiếp chủ động trong việc mua bán và thanh toán. Ngoài ra thì đồng tiền bitcoin không phải sử dụng hóa chất in ấn hay khai thác để tạo ra nó nên rất an toàn với môi trường. Hệ thống máy tính xử lý giao dịch Bitcoin tiêu tốn ít điện năng hơn nhiều so với hệ thống tài chính hiện tại.

Bitcoin ngày càng trở nên phổ biến và nhận được sự quan tâm của rất nhiều người. Nếu xem Bitcoin là một loại hàng hóa, thì ta có thể xem đây là loại hàng hóa đặc biệt, bởi nó không tồn một loại chỉ phí nào để quảng cáo mà vẫn thu hút được nhiều sự chú ý và quan tâm toàn cầu. Ngoài ra, Bitcoin là minh chứng cụ thể cho sự phát triển của công nghệ Blockchain. Dù là đồng tiền kỹ thuật số nhưng Bitcoin hoàn toàn không bị tình trạng double-spending (sử dụng hai lần cho cùng một đồng tiền) do các giao dịch Bitcoin trên hệ thống Internet đều được ghi lại, và lưu trên một cuốn sổ cái. Không có bất kỳ hacker nào có thể tấn công và xóa các ghi nhận này. Nhờ đó mà Blockchain ngày càng được ứng dụng rộng rãi trong nhiều lĩnh vực như ngân hàng, y tế....

Vì sự phổ biến và tính ứng dụng cao của đồng tiền ảo này chúng tôi sẽ thực hiện thiết kế hệ thống dự đoán giá trị Bitcoin. Về phương pháp để dự đoán, chúng tôi xây dựng mô hình dựa trên bộ dữ liệu đã được làm sạch và kết luận xem mô hình nào sẽ tối ưu trong bài toán này.

II. DỮ LIỆU:

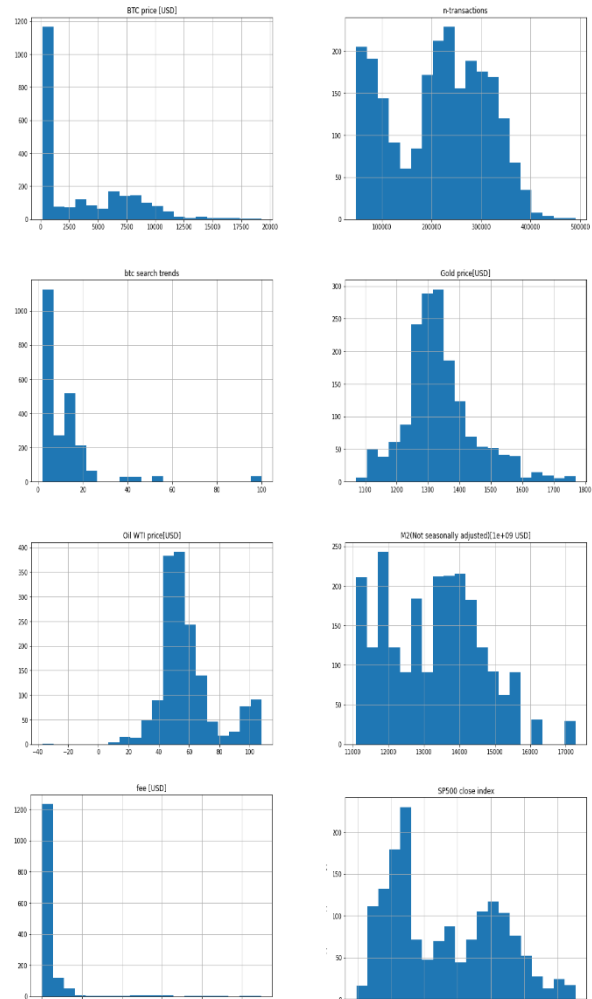
A. Giới thiệu bộ dữ liệu

Bộ dữ liệu chúng tôi sử dụng được tải từ kaggle, bộ dữ liệu được tác giả tổng hợp từ trang web blockchain.com/api với số lượng dữ liệu đủ lớn và phong phú. Vì vậy chúng tôi đã chọn bộ dữ liệu này cho quá trình nghiên cứu.

B. Đặc điểm dữ liệu:

Bộ dữ liệu gồm 8 thuộc tính ảnh hưởng đến giá trị của bitcoin gồm : N-transactions, Fee[USD], Btc searchs, Gold price [USD], SP500 close index, Oil WTI price [USD], M2

Bộ dữ liệu gồm 2313 dòng và 9 cột, trong đó thuộc tính Gold price, SP500 index và oil WTI chứa nhiều NaN vì thị trường không hoạt động vào cuối tuần. Phân bố giá trị của thuộc tính trong bộ dữ liệu được biểu diễn ở hình 1.



Hình 1. Biểu đồ Histogram của các thuộc tính

III. PHƯƠNG PHÁP MÁY HỌC :

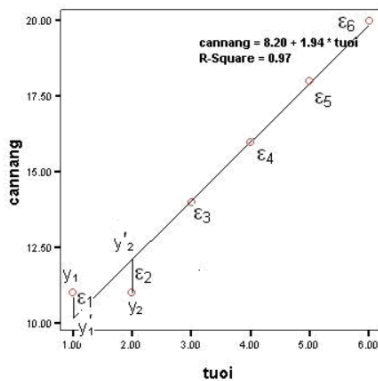
A. Linear Regression :

Trong thống kê, hồi quy tuyến tính (Linear Regression) là một phương pháp phân tích quan hệ giữa biến phụ thuộc Y với một hay nhiều biến độc lập X. mô hình hóa sử dụng hàm tuyến tính (bậc 1). Các tham số của mô hình (hay hàm số) được ước lượng từ dữ liệu.

Ví dụ: Ta có mẫu gồm 6 trẻ em khác nhau có cân nặng theo bảng sau:

Tuổi	Cân nặng (kg)
1	11
2	11
3	14
4	16
5	18
6	20

Hình 2.



Hình 3

Khi vẽ đường thẳng hồi quy, ta thấy các trị số quan sát y_3, y_4, y_5, y_6 nằm trên đường thẳng, còn y_1 và y_2 không nằm trên đường thẳng này và sự liên hệ giữa x_i và y_i không còn là 100% mà chỉ còn 97% vì có sự sai lệch tại y_1, y_2 . Sự sai lệch này trong thống kê gọi là phần dư (residual) hoặc errors.

Gọi $y_1, y_2, y_3, y_4, y_5, y_6$ là trị số quan sát và $y'_1, y'_2, y'_3, y'_4, y'_5, y'_6$ là giá trị ước đoán nằm trên đường quy hồi $\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4, \epsilon_5, \epsilon_6$ là phần dư.

Như vậy

$$\begin{aligned}\epsilon_1 &= y_1 - y'_1 \\ \epsilon_2 &= y_2 - y'_2 \\ \epsilon_3 &= y_3 - y'_3 \\ \epsilon_4 &= y_4 - y'_4 \\ \epsilon_5 &= y_5 - y'_5 \\ \epsilon_6 &= y_6 - y'_6\end{aligned}$$

Khi đó phương trình hồi quy tuyến tính được viết dưới dạng tổng quát như sau :

$$y' = \beta x_i + \alpha_i + \epsilon_i$$

Như vậy nếu phần dư ϵ_i càng nhỏ sự liên hệ giữa x, y càng lớn và ngược lại. Phần liên hệ còn được gọi là phần quy hồi. Mô hình hồi quy tuyến tính được mô tả như sau :

Dữ liệu= Hồi qui (Regression) + Phần dư (Residual)

B. Decision Tree

Cây quyết định (Decision Tree) là cây phân cấp có cấu trúc được dùng để phân lớp các đối tượng dựa vào dãy các luật. Các thuộc tính của đối tượng có thể thuộc các kiểu dữ liệu khác

nhau như nhị phân (Binary), định danh (Nominal), thứ tự (Ordinal), số lượng (Quantitative) trong khi đó thuộc tính phân lớp phải có kiểu dữ liệu là Binary hoặc Ordinal. Tóm lại, cho dữ liệu về các đối tượng gồm các thuộc tính cùng với lớp (classes) của nó, cây quyết định sẽ sinh ra các luật để dự đoán lớp của các dữ liệu chưa biết.

Dựa theo mô hình trên, ta thấy : Nếu trời nắng, độ ẩm bình thường thì khả năng các bạn nam đi chơi bóng sẽ cao. Còn nếu trời nắng, độ ẩm cao thì khả năng các bạn nam sẽ không đi chơi bóng

Các thuật toán cây quyết định phổ biến nhất bao gồm:

- Chi-squared Automatic Interaction Detection (CHAID)
- Classification và Regression Tree – CART
- Conditional Decision Trees
- C4.5 và C5.0
- Decision Stump
- Iterative Dichotomiser 3 (ID3)
- M5

Ví dụ 2: Bảng dữ liệu này mô tả mối quan hệ giữa thời tiết trong 14 ngày

id	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rainy	mild	high	weak	yes
5	rainy	cool	normal	weak	yes
6	rainy	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rainy	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rainy	mild	high	strong	no

Hình 4. Dữ liệu thời tiết

Các thuộc tính thời tiết :

- *Outlook* nhận một trong ba giá trị: *sunny, overcast, rainy*
- *Temperature* nhận một trong ba giá trị: *hot, cool, mild*
- *Humidity* nhận một trong hai giá trị: *high, normal*.
- *Wind* nhận một trong hai giá trị: *weak, strong*.

Đây có thể được coi là một bài toán dự đoán liệu đội bóng có chơi bóng không dựa trên các quan sát thời tiết. Ở đây, các quan sát đều ở dạng categorical. Cách dự đoán dưới đây tương đối đơn giản và khá chính xác, có thể không phải là cách ra quyết định tốt nhất:

- Nếu *outlook = sunny* và *humidity = high* thì *play = no*.
- Nếu *outlook = rainy* và *windy = true* thì *play = no*.
- Nếu *outlook = overcast* thì *play = yes*.
- Ngoài ra, nếu *humidity = normal* thì *play = yes*.
- Ngoài ra, *play = yes*.

Trong 14 giá trị đầu ra ở Bảng trên, có năm giá trị bằng no và chín giá trị bằng yes. Entropy tại root node của bài toán là:

$$H(S) = -\frac{5}{14} \log\left(\frac{5}{14}\right) - \frac{9}{14} \log\left(\frac{9}{14}\right) \approx 0.65$$

Tiếp theo, chúng ta tính tổng có trọng số entropy của các *child node* nếu chọn một trong các thuộc tính *outlook*, *temperature*, *humidity*, *wind*, *play* để phân chia dữ liệu.

Xét thuộc tính *outlook*. Thuộc tính này có thể nhận một trong ba giá trị *sunny*, *overcast*, *rainy*. Mỗi một giá trị sẽ tương ứng với một *child node*. Gọi tập hợp các điểm trong mỗi *child node* này lần lượt là S_s , S_o , S_r với tương ứng m_s , m_o , m_r . Sắp xếp lại bảng ban đầu theo thuộc tính *outlook* ta đạt được ba bảng nhỏ sau đây .

id	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
11	sunny	mild	normal	strong	yes

id	outlook	temperature	humidity	wind	play
3	overcast	hot	high	weak	yes
7	overcast	cool	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes

id	outlook	temperature	humidity	wind	play
4	rainy	mild	high	weak	yes
5	rainy	cool	normal	weak	yes
6	rainy	cool	normal	strong	no
10	rainy	mild	normal	weak	yes
14	rainy	mild	high	strong	no

Hình 5. Bảng phân loại xét theo thuộc tính *outlook*

Quan sát nhanh ta thấy rằng *child node* ứng với *outlook* = *overcast* sẽ có entropy bằng 0 vì tất cả $m_o = 4$ output đều là yes. Hai *child node* còn lại với $m_s = m_r = 5$ có entropy khá cao vì tần suất output bằng *yes* hoặc *no* là xấp xỉ nhau. Tuy nhiên, hai *child node* này có thể được phân chia tiếp dựa trên hai thuộc tính *humidity* và *wind*.

Ta thấy rằng :

$$H(S_s) = -\frac{2}{5} \log\left(\frac{2}{5}\right) - \frac{3}{5} \log\left(\frac{3}{5}\right) \approx 0.673$$

$$H(S_o) = 0$$

$$H(S_r) = -\frac{3}{5} \log\left(\frac{2}{5}\right) - \frac{3}{5} \log\left(\frac{3}{5}\right) \approx 0.673$$

$$H(\text{outlook}, S) = \frac{5}{14} H(S_s) + \frac{4}{14} H(S_o) + \frac{5}{14} H(S_r) \approx 0.48$$

Xét thuộc tính *temperature*, ta có bảng sau :

id	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
13	overcast	hot	normal	weak	yes

id	outlook	temperature	humidity	wind	play
4	rainy	mild	high	weak	yes
8	sunny	mild	high	weak	no
10	rainy	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
14	rainy	mild	high	strong	no

id	outlook	temperature	humidity	wind	play
5	rainy	cool	normal	weak	yes
6	rainy	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
9	sunny	cool	normal	weak	yes

Hình 6. Bảng phân loại xét theo thuộc tính *temperature*

Gọi S_h , S_m , S_c là ba tập con tương ứng với *temperature* bằng *hot*, *mild*, *cool*. Chúng ta có thể tính được

$$H(S_h) = -\frac{2}{4} \log\left(\frac{2}{4}\right) - \frac{2}{4} \log\left(\frac{2}{4}\right) \approx 0.693$$

$$H(S_m) = -\frac{4}{6} \log\left(\frac{4}{6}\right) - \frac{2}{6} \log\left(\frac{2}{6}\right) \approx 0.637$$

$$H(S_c) = -\frac{3}{4} \log\left(\frac{3}{4}\right) - \frac{1}{4} \log\left(\frac{1}{4}\right) \approx 0.562$$

$$H(\text{temperature}, S) = \frac{4}{14} H(S_h) + \frac{6}{14} H(S_m) + \frac{4}{14} H(S_c) \approx 0.631$$

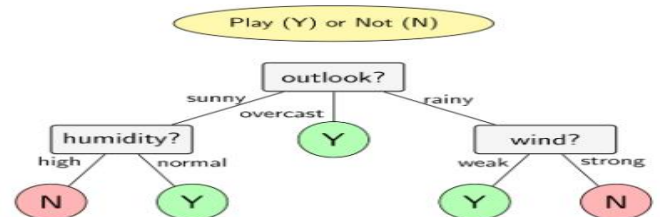
Tính toán tương tự với 2 thuộc tính còn lại

$$H(\text{humidity}, S) \approx 0.547, \quad H(\text{wind}, S) \approx 0.618$$

Như vậy, thuộc tính cần chọn ở bước đầu tiên là *outlook* vì $H(\text{outlook}, S)$ đạt giá trị nhỏ nhất (information gain là lớn nhất).

Sau bước phân chia đầu tiên này, ta nhận được ba *child node* với các phần tử như trong hình 5. *Child node* thứ hai không cần phân chia tiếp vì nó đã tinh khiết. Với *child node* thứ nhất, ứng với *outlook* = *sunny*, kết quả tính được bằng ID3 sẽ cho chúng ta thuộc tính *humidity* vì tổng trọng số của entropy sau bước này sẽ bằng 0 với output bằng *yes* khi và chỉ khi *humidity* = *normal*. Tương tự, *child node* ứng với *outlook* = *wind* sẽ được tiếp tục phân chia bởi thuộc tính *wind* với output bằng *yes* khi và chỉ khi *wind* = *weak*.

Như vậy, câu quyết định cho bài toán dựa trên ID3 sẽ có dạng như hình 7



Hình 7. Decision tree cho bài toán ví dụ sử dụng thuật toán ID3

C. Ridge Regression:

Hồi quy dạng sóng (Ridge Regression) là mô hình hồi quy phân tích mối quan hệ giữa các biến độc lập và biến phụ thuộc sử dụng phương pháp Regularization, điều chỉnh mô hình sao cho giảm thiểu các vấn đề Overfitting, tối ưu hay kiểm soát mức độ phức tạp của mô hình để cân đối giữa Biased và Variance qua đó giảm sai số của mô hình. Công thức tổng quát của mô hình :

$$\sum_{i=1}^M (y_i - \hat{y}_i)^2 = \sum_{i=1}^M \left(y_i - \sum_{j=0}^p w_j \times x_{ij} \right)^2 + \lambda \sum_{j=0}^p w_j^2$$

Hệ số lamda còn là tham số Regularization, hay tham số Penalty, hay tham số Shrinkage, là số luôn dương, là giá trị mà ở đó phương trình tuyến tính sẽ được “tinh chỉnh” sao cho sai số của mô hình được giảm thiểu tối đa, nghĩa là giá trị lambda nào mà mô hình đạt MSE (Mean Square Error) sẽ được chọn. W_i là hệ số β của phương trình hồi quy tuyến tính.

D. ĐỘ ĐO :

Về các độ đo sử dụng để đánh giá gồm có RMSE, R2 và Explained variance.

R2 là một thống kê được sử dụng trong bối cảnh của các mô hình thống kê có mục đích chính là dự đoán các kết quả trong tương lai hoặc kiểm tra các giả thuyết, trên cơ sở các thông tin liên quan khác. Nó cung cấp một thước đo về mức độ nhân rộng của các kết quả quan sát được của mô hình, dựa trên tỷ lệ của tổng biến động của các kết quả được mô hình giải thích.

RMSE là một biện pháp thường được sử dụng trong những khác biệt giữa các giá trị (mẫu hoặc các giá trị dân) được dự đoán bởi một mô hình hay một ước lượng và các giá trị quan sát được. RMSD đại diện cho căn bậc hai của thời điểm mẫu thứ hai về sự khác biệt giữa các giá trị dự đoán và giá trị quan sát hoặc giá trị trung bình bậc hai của những khác biệt này. Các độ lệch này được gọi là phần dư khi

Explained variation được giải thích là sự biến đổi đo lường tỷ lệ mà một mô hình toán học chiếm sự thay đổi (độ phân tán) của một tập dữ liệu nhất định

IV. PHƯƠNG PHÁP MÁY HỌC

Ridge Regression

Parameter				Độ đo
alpha	solver	Fit_intercept	normalize	R2
1e-5	svd	True	False	0.764190
1e-5	svd	False	False	0.707459
1e-4	svd	True	False	0.864381
1e-4	svd	False	False	0.708019
1e-5	cholesky	True	False	0.864261
100	cholesky	True	True	0.045210
1	cholesky	True	True	0.681229
1e-5	cholesky	True	True	0.764430
1e-5	lsqr	True	False	0.568257
100	lsqr	True	False	0.568056
100	lsqr	False	True	0.313041
1	sag	True	False	0.585690
1e-2	sag	False	False	0.334022
1e-2	sag	True	True	0.763847
1e-5	sag	True	False	0.585682

Linear Regression

Parameter			Độ đo
copy_X	Fit_intercept	normalize	R2
False	False	False	0.764190
False	False	True	0.700457
False	True	False	0.865588
False	True	True	0.703215
True	False	False	0.700457
True	False	True	0.700457
True	True	False	0.865588
True	True	True	0.703215

Decision Tree Regression

Parameter		Độ đo
criterion	Max_depth	R2
Friendman_mse	1	0.76403015
Friendman_mse	2	0.87769822
Friendman_mse	4	0.95279453
Friendman_mse	10	0.98426804
Friendman_mse	15	0.98511187
mse	1	0.7640301
mse	2	0.87769822
mse	4	0.95279453
mse	10	0.98400363
mse	15	0.98432895
mae	1	0.76086172
mae	2	0.86933782
mae	4	0.93787722
mae	10	0.97829171
mae	15	0.98013488

Chúng tôi sử dụng một vài tham số phổ biến nhất cho từng mô hình để tiến hành xem xét độ đo R2 khi thay đổi giá trị của các tham số đó, các tham số còn lại để giá trị mặc định. Việc thay đổi tham số và kết quả tính toán độ đo R2 của từng mô hình được thể hiện trong 3 bảng ở trên.

Đối với mô hình *Ridge Regression* sau khi sử dụng lớp *GridSearch* (tìm kiếm theo lưới) của *Scikit_learn* ta có được bảng giá trị của các parameter làm cho độ đo R2 lớn nhất và giá trị nhỏ nhất thu được khi ta làm thí nghiệm với các tham số.

Parameter				Độ Đo
solver	alpha	fit_intercept	normalize	R2
svd	1e-5	True	False	0.865381
lsqr	100	False	True	0.313041

Tương tự khi sử dụng *GridSearchCV* (tìm kiếm theo lưới) cho mô hình *Linear Regression* ta thu được bảng sau:

Parameter			Độ đo
Copy_X	fit_intercept	normalize	R2
True	True	False	0.8655892195469982
True	False	False	0.7004573264852672

Và cuối cùng cho mô hình Decision tree Regression

Parameter			Độ đo
Max_depth	criterion	splitter	R2
1	mae	best	0.7608617275016
20	friedman_mse	best	0.9840174040153

Khi tăng giá trị của tham số max_depth (chiều sâu tối đa của cây) từ 1 đến vô cùng độ đo R2 có xu hướng tăng từ 0.76 đến giới hạn là 0.986. Khi cho max_depth lớn hơn 3 mô hình có xu hướng bị overfitting vì vậy chúng tôi chọn giá trị cho tham số max_depth là 2.

Qua các thí nghiệm trên tham số của từng mô hình chúng tôi chọn ra các tham số thích hợp để áp dụng lên mô hình máy học tránh các giá trị gây ra hiện tượng overfitting và underfitting. Kết quả các độ đo của 3 phương pháp máy học được ghi trong bảng bên dưới.

	RMSE	R2	Explained Variance
Linear Regression	1415.427	0.86558840	0.86558922
Decision Tree	1395.545	0.86933782	0.86933782
Ridge Regression	1416.513	0.86538192	0.86538530

Nhìn chung, mô hình Decision Tree có thể cho ra độ chính xác cao nhất, tối đa lên đến 98%, tiếp theo là Linear Regression là 87% và cuối cùng là Ridge Regression 86%. Tùy thuộc vào tham số truyền vào mà ta có thể cân nhắc lựa chọn 1 mô hình phù hợp nhất.

V. PHÂN TÍCH LỖI, HƯỚNG PHÁT TRIỂN

A. Phân tích lỗi

Vì giá bitcoin luôn biến động theo nhiều yếu tố bao gồm cả yếu tố thời gian như ngày, tháng, năm. Việc mã hóa thuộc tính date bằng thư viện LabelBinarizer sẽ làm mất đi sự tương quan của ngày, tháng, năm với giá bitcoin dẫn đến sai sót khi dự đoán giá bitcoin ở các thời gian cụ thể khác nhau. Vì thế để tăng sự phụ thuộc của giá bitcoin vào yếu tố thời gian cụ thể hơn thì việc chia nhỏ thuộc tính date sẵn có của mô hình ra thành 3 thuộc tính ngày, tháng, năm riêng lẻ sẽ giúp tăng hiệu quả của mô hình.

B. Hướng phát triển.

Để phát triển mô hình chúng tôi sẽ cân nhắc sử dụng thêm các mô hình máy học hiện đại và phù hợp hơn cho mô hình của mình. Áp dụng thêm các phương pháp xử lý dữ liệu mới để nâng cao hiệu quả của mô hình. Đồng thời biến mô hình chạy đang chạy bằng các dòng lệnh này thành một ứng dụng để tiện lợi cho việc sử dụng và cập nhập giá bitcoin dự đoán hằng ngày cho người dùng.

VI. KẾT LUẬN:

Nhìn chung mô hình máy học đã hoàn thiện và chạy ổn định với độ chính xác cao khoảng 0.86. Tuy nhiên bộ dữ liệu còn ít chưa bao quát được các thời điểm gần đây, mô hình máy học được sử dụng còn khá đơn giản nằm ở mức đơn cơ bản. Vì thế việc đưa mô hình vào sử dụng cho việc dự đoán giá bitcoin chuyên nghi hiện tại vẫn chưa thể đáp ứng được

VII. TÀI LIỆU THAM KHẢO:

- [1] Girshick, R. (2015). Fast r-cnn. In Proceedings of the IEEE international conference on computer vision (pp.1440-1448).
- [2] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).
- [3] Ian H. Witten Eibe Frank Mark A. Hall(2011) Data Mining: Practical Machine Learning Tools and Techniques(pp.11-14)
- [4] https://scikit-learn.org/stable/modules/model_evaluation.html
- [5] <http://ml4vn.blogspot.com/2017/08/linear-models-mo-hinh-tuyen-tinh.html>
- [6] <https://machinelearningmastery.com/hyperparameter-optimization-with-random-search-and-grid-search>
- [7] <https://vi.wikipedia.org/wiki/Bitcoin>
- [8] <https://www.phamduytung.com/blog/2019-04-19-deep-learning-view>