Dự báo bán hàng

Nguyễn Nhật Thiên Tân, Nguyễn Tiến Đạt, Lê Huy Quang, Trần Thành Hùng

Đại học Công nghệ Thông tin - Đại học Quốc gia Thành phố Hồ Chí Minh

{19520922,20521171, 20521804, 20521374}@gm.uit.edu.vn

Dự báo bán hàng (Sales Forecast) là sự ước đoán về lượng bán của doanh nghiệp, (tính bằng tiền hoặc theo đơn vị sản phẩm) có thể bán được trong một thời kì nhất định dưới một kế hoạch marketing đã được thông qua và dưới một tổ hợp các điều kiện kinh tế được giả định. Trong bài báo cáo này, nhóm chúng em sẽ thực hiện dự báo bán hàng dựa trên bộ dữ liệu do Walmart Recruiting cung cấp miễn phí trên Kaggle. Phương pháp tiếp cận của chúng em là thử nghiệm một số mô hình máy học LinearRegression, DecisionTree, Random Forest, SVM . Các độ đo đánh giá mô hình được sử dụng trong bài báo cáo này là RMSE, MAE, R2.

Keywords: Sales Forecast, Walmart, LinearRegression, DecisionTree, Random Forest, SVM.

I. GIỚI THIỆU BÀI TOÁN , BỐI CẢNH

Walmart tổ chức một số sự kiện giảm giá khuyến mại trong suốt cả năm. Nhưng có một số biến động trong doanh số bán hàng diễn ra trước các ngày lễ nổi bật, bốn ngày lễ lớn nhất, đó là Super Bowl, Ngày Lao động, Lễ Tạ ơn và Giáng sinh. Các tuần bao gồm các ngày lễ này được đánh giá có trọng số cao hơn năm lần so với các tuần không nghỉ lễ. Vì vậy, nhóm chúng em quyết đinh sẽ thử áp dụng một số thuật toán máy học để dự đoán doanh số bán hàng cho tuần tiếp theo. Dữ liệu bán hàng lịch sử cho 45 cửa hàng Walmart ở các khu vực khác nhau đều có sẵn.

Input: Doanh số bán hàng của 45 cửa hàng Walmart (từ 05-02-2010 đến 01-11-2012). Doanh số này được tính theo mỗi tuần.

Task: dự đoán doanh số bán hàng 1 tuần

E: Gồm tập train và test

P: tỉ lệ dự đoán doanh số bán hàng.

Output: Doanh số bán hàng mỗi tuần của 1 hay nhiều cửa hàng

Tại sao phải dự báo bán hàng:

- Dự báo doanh số là nền tảng để thiết lập và duy trì sản xuất
- Giúp quyết định số lượng hàng cần, thời gian, nhân lực, nguồn lực, vật liệu, máy móc, vật tư, v v.
- Nó ảnh hưởng đến vấn đề tài chính: vay bao nhiêu, dòng tiền thu về như thế nào,…

II. GIỚI THIỆU DATASET

Nguồn thu thập và cách thức thu thập :

- Bộ dữ liệu được sử dụng trong đồ án là một phần của bộ dữ liệu do Walmart Recruiting cung cấp miễn phí trên Kaggle.
- Đây là dữ liệu lịch sử bán hàng của 45 cửa hàng Walmart nằm ở các khu vực khác nhau.

Dataset gồm 6435 hàng và 8 cột (đặc trưng)

đây là dữ liệu lịch sử bao gồm doanh số bán hàng từ 2010-02-05 đến 2012-11-01, trong tệp WalmartStoresales. Trong tệp này bao gồm các thuộc tính:

Store – Số lượng cửa hàng

Date – tuần bán hàng được xác định bởi ngày đầu tiên trong tuần

Weekly_Sales – doanh số bán hàng cho bộ phân nhất định trong của hàng nhất định

Holiday_Flag - tuần này có phải tuần lễ đặc biệt gì không

Temperature - Nhiệt độ vào ngày bán

Fuel_Price - Giá nhiên liệu của khu vực đó

CPI – Chỉ số giá tiêu dùng (dùng để đo lường mức giá trung bình của giỏ hàng hóa và dịch vụ mà một người tiêu dùng điển hình)

Unemployment - Tỷ lệ thất nghiệp

Những tuần có ngày lễ đặc biệt trong dataset:

Super Bowl: 12-Feb-10, 11-Feb-11, 10-Feb-12, 8-Feb-13

Labour Day: 10-Sep-10, 9-Sep-11, 7-Sep-12, 6-Sep-13

Thanksgiving: 26-Nov-10, 25-Nov-11, 23-Nov-12, 29-Nov-13

Christmas: 31-Dec-10, 30-Dec-11, 28-Dec-12, 27-Dec-13

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment
0	1	05-02-2010	1643690.90	0	42.31	2.572	211.096358	8.106
1	1	12-02-2010	1641957.44	1	38.51	2.548	211.242170	8.106
2	1	19-02-2010	1611968.17	0	39.93	2.514	211.289143	8.106
3	1	26-02-2010	1409727.59	0	46.63	2.561	211.319643	8.106
4	1	05-03-2010	1554806.68	0	46.50	2.625	211.350143	8.106
5	1	12-03-2010	1439541.59	0	57.79	2.667	211.380643	8.106
6	1	19-03-2010	1472515.79	0	54.58	2.720	211.215635	8.106
7	1	26-03-2010	1404429.92	0	51.45	2.732	211.018042	8.106
8	1	02-04-2010	1594968.28	0	62.27	2.719	210.820450	7.808
9	1	09-04-2010	1545418.53	0	65.86	2.770	210.622857	7.808

10 dòng dữ liệu đầu tiên của dataset

III. PHƯƠNG PHÁP MÁY HỌC

A. *Tiền xử lý dữ liệu*

- Scale data

Scaling là biến đổi khoảng giá trị của dữ liệu về một khoảng đặc biệt như 0-100 hay 0-1, thường là 0-1. Trong một số thuật toán Machine Learning mà khoảng cách giữa các điểm dữ liệu là quan trong, như SVM hay KNN, thì việc scale dữ liệu là vô cùng quan trọng, vì mỗi thay đổi nhỏ của dữ liệu cũng mang đến kết quả khó đoán trước.

Trong bài báo cáo này nhóm chúng em sẽ scale data về khoảng 0-1 để dễ dàng thao tác với các mô hình máy học. Đồng thời cũng chuyển toàn bộ các dữ liệu số về dạng numeric để dễ dàng tính toán.

- Loại bỏ outlier

Outliers/anomalies (dữ liệu ngoại lai/dữ liệu bất thường) là một trong những thuật ngữ được sử dụng rất rộng rãi trong thế giới data và đặc biệt là data science. Xác định và loại bỏ outliers là một bước cực kỳ quan trọng trong quá trình xử lý dữ liệu. Việc xử lý các dữ liệu ngoại lai sẽ giúp tăng cao độ

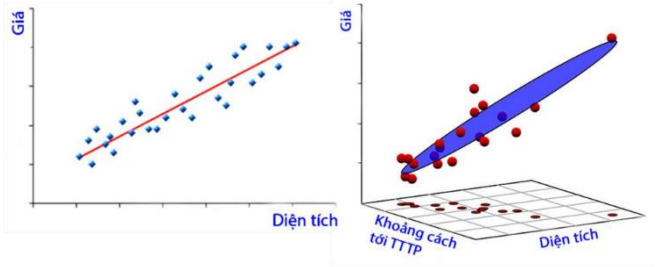
chính xác cho các mô hình dự đoán hay các báo cáo doanh nghiệp một cách đáng kể.
Sau khi loại bỏ các outliers thì dataset có số dòng giảm từ 6435 xuống 5953 dòng.

B. Các phương pháp học máy

LINEAR REGRESSION

Linear Regression là một kĩ thuật thường được dùng trong các mô hình phân tích và dự đoán. Mô hình được nhà toán học Carl Friedrich Gauss phát minh vào đầu thế kỉ 19. Mô hình chỉ ra quan hệ tuyến tính giữa một biến phụ thuộc vào một hay nhiều biến độc lập. Có hai mô hình Linear Regression: Simple Linear Regression và Multiple Linear Regression. Trong Linear Regression, giá trị của biến phụ thuộc và biến độc lập phải là giá trị liên tục (số thực).

Mục tiêu của hồi quy tuyến tính là tạo ra một đường thẳng, mặt phẳng hoặc siêu mặt phẳng sao cho nó nằm càng gần với các điểm dữ liệu càng tốt.



MINH HỌA BÀI TOÁN SỬ DỤNG HỒI QUY TUYẾN TÍNH

Mô hình hồi quy tuyến tính đơn giản có dạng:

$$y = w_0 + w_1 x + \varepsilon$$

Trong đó:

+ Biến x được gọi biến độc lập.

+ Biến y được gọi là biến phụ thuộc (biến phụ thuộc vào biến độc lập) + $w0$ và $w1$ được gọi là các tham số của mô hình.

+ Các tham số và không được biết trước và sẽ được ước lượng dựa vào dữ liệu

Để tìm được giá trị cho $w0$ và $w1$, ngoài cách sử dụng công thức thống kê truyền thống, chúng ta có thể sử dụng thuật toán tối ưu Gradient Descent để lựa chọn đường khớp nhất với dữ liệu thông qua việc xác định cực tiểu của biểu thức hàm loss (loss function). Một trong những hàm loss thông dụng nhất là hàm mất mát L2 , với công thức là:

$$L(w) = \frac{1}{2N} \|y - Xw\|_2^2$$

Trong đó:

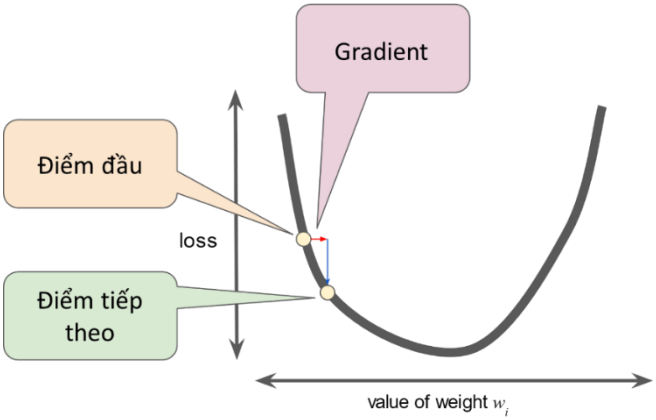
+ N là số điểm dữ liệu trong bộ dữ liệu

+ n là số biến độc lập

+ y là vector cột $N \times 1$ chứa giá trị thực của biến phụ thuộc.

+ X là ma trận biến độc lập $N \times n$ mà mỗi hàng của nó là tập hợp các biến độc lập của một điểm dữ liệu.

+ w là vector hệ số $n \times 1$



TỐI ƯU HÓA HÀM LOSS

Để tìm được vector hệ số w thỏa hàm mất mát có giá trị nhỏ nhất, w được cập nhật theo công thức:

$$w := w - \alpha \nabla_w L(w)$$

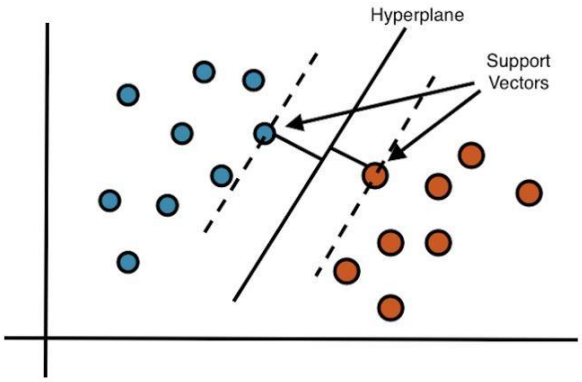
với $\nabla_w L(w) = \frac{1}{-} X^T (Xw - y)_N$, α là learning rate

4.1.2. Support Vector Machine

SVM (Support Vector Machine) là một thuật toán học máy có giám sát

(Supervised Learning) được sử dụng rất phổ biến ngày nay trong các bài toán phân lớp (Classification) hay hồi quy (Regression). SVM được đề xuất bởi Vladimir N.

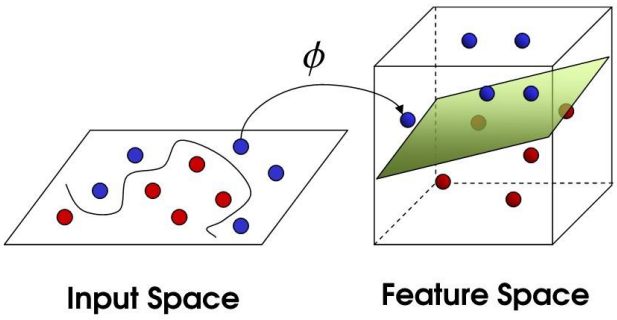
Vapnik và các đồng nhiệp của ông vào năm 1963 tại Nga và sau đó trở nên phổ biến trong những năm 90 nhờ ứng dụng giải quyết các bài toán phi tuyến tính (nonlinear) bằng phương pháp Kernel Trick. Trong thuật toán này, chúng ta vẽ đồ thị dữ liệu là các điểm trong n chiều (n là số lượng các thuộc tính). Sau đó ta xây dựng một siêu phẳng (hyperplane) có (n - 1) chiều trong không gian n chiều đó sao cho siêu phẳng này phân loại các lớp một cách tối ưu nhất.



Ý tưởng của thuật toán Support Vector Machine

Bốn thành phần cấu thành một mô hình SVM gồm có: Kernel, Regularization, Gamma và Margin.

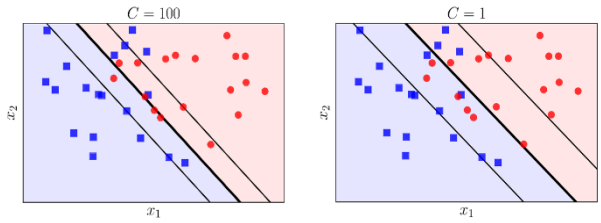
Kernel là một hàm ánh xạ dữ liệu từ không gian ít nhiều hơn sang không gian nhiều chiều hơn. Đây là kỹ thuật quan trọng trong SVM. Có 3 loại kernel thường gặp: kernel dạng tuyến tính (linear kernel), kernel dạng đa thức (polynomial kernel), kernel dạng lũy thừa (exponential kernel). Việc sử dụng Kernel function khiến cho các phương pháp chuyển không gian trở nên linh hoạt, từ đó có thể tìm ra mặt phẳng tối ưu nhất để phân chia dữ liệu.



Biến đổi không gian 2 chiều sang 3 chiều nhờ Kernel

Kernel dạng đa thức và dạng lũy thừa tính toán đường phân cách ở những chiều không gian cao hơn và được gọi là kernel trick.

Tham số Regularization (còn gọi là tham số C) điều chỉnh việc có nên bỏ qua các điểm dữ liệu bất thường trong quá trình tối ưu mô hình SVM. Nếu tham số này có giá trị lớn, siêu phẳng được chọn sao khoảng cách giữa nó tới các điểm dữ liệu của 2 lớp sẽ có giá trị nhỏ (small-margin). Ngược lại, khi tham số này có giá trị nhỏ, siêu phẳng sẽ được xây dựng sao cho khoảng cách với các điểm dữ liệu của 2 lớp có giá trị lớn (large-margin), kể cả khi siêu phẳng này sẽ phân loại sai nhiều điểm dữ liệu hơn.



Hình 4.1.2.3 Tác dụng của tham số C trong SVM

Gamma

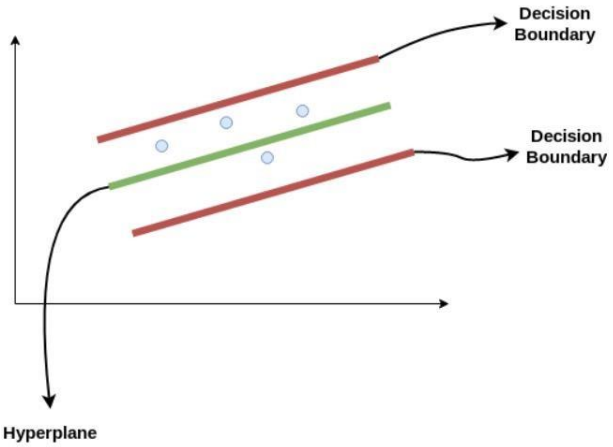
Tham số gamma xác định việc sử dụng bao nhiêu điểm dữ liệu cho việc xây dựng siêu phẳng phân cách. Với giá trị gamma nhỏ, các điểm dữ liệu nằm xa đường phân cách sẽ được sử dụng trong việc tính toán đường phân cách. Ngược lại, với giá trị gamma lớn, chỉ những điểm nằm gần đường phân cách mới được sử dụng để tính toán.

Margin

Margin trong SVM là khoảng cách giữa siêu phẳng phân cách với các điểm dữ liệu gần nó nhất. Khoảng cách này đối với các điểm dữ liệu gần nhất của cả 2 lớp càng lớn thì mô hình càng phân loại chính xác.

Support Vector Regression

Support Vector Regression sử dụng nguyên lý của SVM, nhưng được áp dụng vào bài toán hồi quy.

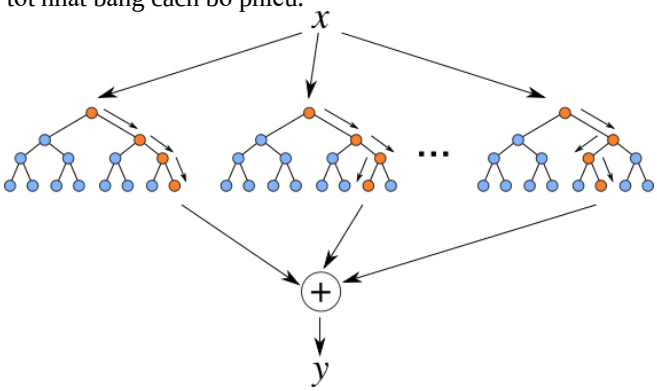


Hình 4.1.2.4 Các vector supports trong SVR

Để tối thiểu hóa giá trị hàm mất mát, SVR đi tìm đường thẳng (mặt phẳng hay siêu phẳng) cho margin tối đa, đồng thời chấp nhận một phần lỗi.

4.1.3. Random Forest

Random Forest là thuật toán học có giám sát (supervised learning), nằm trong họ thuật toán cây quyết định (decision tree). Nó có thể được sử dụng cho cả phân lớp và hồi quy. Random forest tạo ra cây quyết định trên các mẫu dữ liệu được chọn ngẫu nhiên, được dự đoán từ mỗi cây và chọn giải pháp tốt nhất bằng cách bỏ phiếu.



Hình 4.1.3.1 Ý tưởng của thuật toán Random Forest

Random Forest là mô hình gồm nhiều mô hình con (ensemble learning) rất hiệu quả cho các bài toán, nhất là các bài toán phân loại vì nó huy động cùng lúc hàng trăm mô hình nhỏ hơn bên trong với quy luật khác nhau để đưa ra quyết định cuối cùng. Mỗi mô hình con có thể mạnh yếu khác nhau, nhưng theo nguyên tắc wisdom of the crowd, ta sẽ có cơ hội phân loại chính xác hơn so với khi sử dụng bất kì một mô hình đơn lẻ nào. Như tên gọi của nó, Random Forest (RF) dựa trên cơ sở :

- Random = Tính ngẫu nhiên;

- Forest = Nhiều cây quyết định (decision tree).

Cây quyết định là một cây phân cấp có cấu trúc được dùng để phân lớp các đối tượng dựa vào dãy các luật (series of rules). Ngoài Random forest , các mô hình mạnh mẽ đang được sử dụng phổ biến như Gradient Boosting (boosting method) và
XGBoost (boosting method) đều được xây dựng dựa trên thuật toán cây quyết định. Có 2 loại cây quyết định: Cây quyết định biến phân loại và cây quyết định biến liên tục. Mục tiêu là tạo ra một mô hình dự đoán giá trị của biến mục tiêu (target variable) bằng cách học các quy tắc quyết định đơn giản được suy ra từ các đặc trưng dữ liệu (data features). Có 4 thuật toán cây quyết định phổ biến: ID3, CART, Chi-Sq, Reduction in Variance. Trong thư viện sklearn, cây quyết định được cài đặt mặc định dựa trên thuật toán CART với chỉ số đo lường độ nhiễu loạn của thông tin là Gini impurity:

$$G(p) = 1 - \sum_{i=1}^{C} (pi)2$$

Với C là số lớp cần phân loại, $pi = n\_Ni$, ni là số lượng phần tử ở lớp thứ i. Còn N là tổng số lượng phần tử ở node đó.

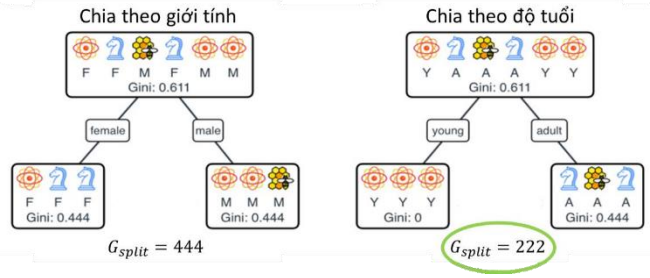
Để có thể tìm ra cách phân chia tối ưu nhất, người ta xác định thuộc tính có hệ số $Gsplit$ nhỏ nhất theo công thức:

$$Gsplit = \sum_{i=1}^{n} \frac{N}{N} G(i)$$

Trong đó:

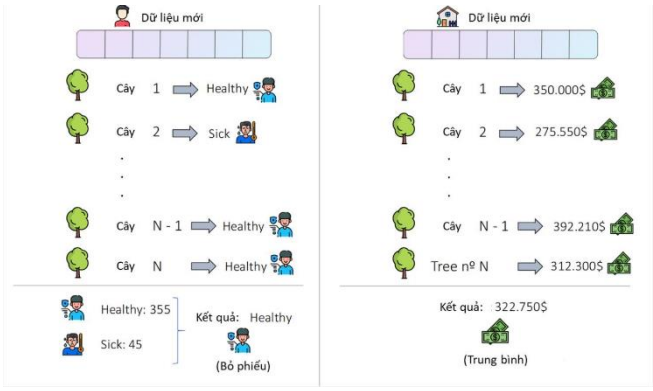
+ Ni là số điểm dữ liệu có trong node của nhánh được phân

+ N là số điểm dữ liệu có trong node được dùng để phân nhánh



Hình 4.1.3.2 Lựa chọn hệ số gini split nhỏ nhất

Random forests tạo ra cây quyết định trên các mẫu dữ liệu được chọn ngẫu nhiên, được dự đoán từ mỗi cây và chọn giải pháp tốt nhất bằng cách bỏ phiếu hoặc tính giá trị trung bình.



Hình 4.1.3.3 Kết quả cuối cùng của thuật toán Random Forest

Random Forest có thể làm việc được với dữ liệu thiếu giá trị, Khi Forest có nhiều cây hơn, chúng ta có thể tránh được việc Overfitting với tập dữ liệu

Mục 4: Kết quả thực nghiệm và đánh giá

Độ đo đánh giá:

Độ đo được nhóm tụi em lựa chọn làm thang đánh giá cho bài toán này là căn của sai số toàn phương trung bình, viết tắt RMSE (Root mean squared error). Đây là một độ đo phổ biến cho các bài toán hồi quy. RMSE cho biết mức độ phân tán các giá trị dự đoán từ các giá trị thực tế. Công thức tính RMSE là:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

RMSE là một đánh giá tốt cho các sai số và rất nhạy với phương sai của biến phụ thuộc. Giá trị RMSE sẽ thấp nếu biến phụ thuộc có phương sai nhỏ, và giá trị RMSE sẽ cao trong tình huống ngược lại.

Do vậy, khi nhìn vào chỉ số RMSE, các giá trị ngoại lệ (outliers) dễ dàng nhận biết có xuất hiện trong bộ dữ liệu. Trong các bài toán mà giá trị dự đoán lên xuống thất thường như bài toán này, độ đô thích hợp để đánh giá hiệu năng của các mô hình là chỉ số RMSE.

Ở độ đo sai số tuyệt đối trung bình (MAE) có công thức $MAE = \frac{1}{n}\sum_{j=1}^{n} |y_i - \hat{y}_i|$, trong đó tất cả các sự chênh lệch giá dự đoán – giá thực tế riêng lẻ có trọng số bằng nhau. Khác với MAE, khi có sự khác nhau quá lớn giữa giá mô hình đưa ra và giá thực tế, con số chênh lệch này sẽ được bình phương (theo công thức RMSE). Đây là một độ đo lí tưởng cho các mô hình đặt mục tiêu đưa ra giá trị sát với thực tế nhất có thể (tương tự như bài toán này).

Đồng thời nhóm chúng em cũng sử dụng thêm độ đo R^2 để xác định độ phù hợp của mô hình với bài toán này.

Công thức tính hệ số R bình phương xuất phát từ ý tưởng: toàn bộ sự biến thiên của biến phụ thuộc được chia làm hai phần: phần biến thiên do hồi quy và phần biến thiên không do hồi quy(còn gọi là phần dư).

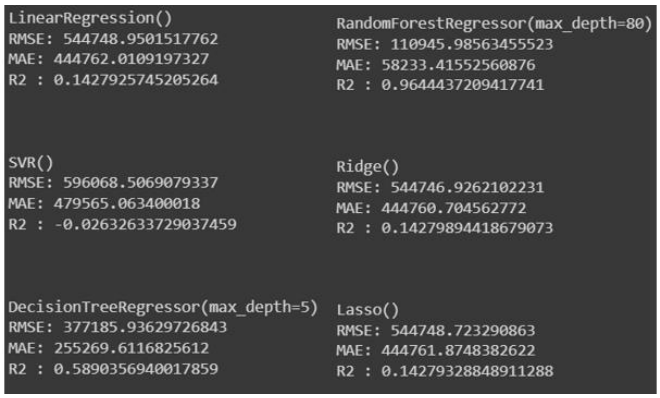
$$R^2 = 1 - (ESS/TSS)$$

Trong đó:
Regression Sum of Squares(RSS): tổng các độ lệch bình phương giải thích từ hồi quy

Residual Sum of Squares(ESS): tổng các độ lệch bình phương phần dư

Total Sum of Squares(TSS): tổng các độ lệch bình phương toàn bộ

Giá trị R bình phương dao động từ 0 đến 1. R bình phương càng gần 1 thì mô hình đã xây dựng càng phù hợp với bộ dữ liệu dùng chạy hồi quy. R bình phương càng gần 0 thì mô hình đã xây dựng càng kém phù hợp với bộ dữ liệu dùng chạy hồi quy. Trường hợp đặt biệt, phương trình hồi quy đơn biến (chỉ có 1 biến độc lập) thì R2 chính là bình phương của hệ số tương quan r giữa hai biến đó.
Kết quả thử nghiệm và đánh giá:



Nhận xét:RMSE và MAE cao vì RMSE,MAE dễ bị ảnh hưởng bởi phạm vi của biến phụ thuộc của bạn. Vì biến phụ thuộc có phạm vị rộng nên RMSE sẽ cao. Mô hình Ridge() và Lasso() cho ra RMSE và MAE giống nhau và cao hơn nhiều so với DecisionTreeRegressor và RandomForestRegressor() DecisionTreeRegressor và RandomForestRegressor() có R2(phần trăm biến phụ thuộc có thể dự đoán) cao hơn nhiều so với Ridge() và Lasso Qua 4 mô hình RandomForestRegressor tốt nhất cả về RMSE(mức độ phân tán các giá trị dự đoán so với các giá trị thực tế), và R2(phần trăm biến phụ thuộc có thể dự đoán)

IV. CÁC THỬ NGHIỆM TINH CHỈNH MÔ HÌNH

Sau khi huấn luyện và lưu lại kết quả đánh giá trên các mô hình cơ sở ở bước trước đó, nhóm em sử dụng kĩ thuật Tìm kiếm theo lưới (Grid search) để chọn ra các tổ hợp siêu tham số (hyperparmeters) mang lại kết quả cao khi đánh giá trên tập test.

	DecisionTree Regressor(max_depth=12)	RandomForestRegressor(max_depth=80, n_estimators=200)	SVR(C=1000)	Lasso(alpha=10)	Ridge(alpha=10)
RMSE	143649.2492	112111.23620	596024.184289948	544746.682609	544729.4893259
MAE	72202.68432	58265.243500797566	479544.105723318	444760.6487918	444749.3304475
R2	0.940392646	0.963692913213149	0.02617371128047	0.142799710834	0.142853819857

Nhìn chung thì sau bước hiệu chỉnh siêu tham số kết quả đánh giá RMSE trên các mô hình máy học trên đều có giảm xuống. Riêng mô hình DecisionTree có sự cải thiện rõ rệt nhất giữa kết quả trước qua sau khi lựa chọn siêu tham số.

V. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trong báo cáo này, nhóm em thử nghiệm giải quyết bài toán dự đoán giá cổ phiếu sử dụng 6 mô hình: Linear Regression, Support Vector Machine, Random Forest, Decision Tree, Lasso và Ridge. Thông qua một vài thao tác hiệu chỉnh siêu tham số của các mô hình, kết quả đánh giá đã được cải thiện so với kết quả trên mô hình cơ sở mà nhóm em đề ra. Kết quả thu được sau khi tiến hành thực nghiệm và tinh chỉnh mô hình cho thấy Random Forest là mô hình phù hợp nhất với bài toán dự báo bán hàng này.
Trong những nghiên cứu tương lai, nhóm em sẽ thử những phương pháp tiếp cận khác trong ngay từ bước xử lí định dạng input, output của bài toán, cho đến bước xây dựng mô hình và đánh giá kết quả. Đầu vào của mô hình có thể được thay thế bằng dataset có nhiều dữ liệu hơn và mới hơn chẳng hạn như dataset liên tục trong 5 năm gần đây nhất (2017 – 2021). Đồng thời sử dụng các thang đo khác như: sai số toàn phương trung bình (MSE - Mean squared errors) hoặc sai số phần trăm tuyệt đối trung bình (MAPE - Mean absolute percentage error) để đánh giá tốt hơn giữa các mô hình.

VI. TÀI LIỆU THAM KHẢO