ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC - KỸ THUẬT MÁY TÍNH

XÁC SUẤT THỐNG KÊ (MT2013)

Nhận diện hình ảnh quảng cáo thông qua các đặc trưng

GVHD: Nguyễn Tiến Dũng

SV:
Phan Thảo Vy - 2252930
Vũ Ngọc Thiên Thư - 2114963
Nguyễn Văn Thịnh - 2213301

Lớp: DT01 - Nhóm 4 - HK242

TP. HỒ CHÍ MINH, THÁNG 5/2025

Contents

Danh sách thành viên và khối lượng công việc

Họ và tên	MSSV	Nhiệm vụ	Phần trăm công việc
Phan Thảo Vy	2252930	Thống kê mô tả, hồi quy logistic	100%
Vũ Ngọc Thiên Thư	2114963	Thêm dữ liệu vào, xử lý tiền dữ liệu	100%
Nguyễn Văn Thịnh	2213301	Decision tree, Random Forest	100%

Contents

1 Cơ sở lý thuyết

1.1 Giới thiệu mô hình hồi quy tuyến tính bội

Hồi quy tuyến tính là một phương pháp để dự đoán giá trị biến phụ thuộc (Y) dựa trên giá trị của biến độc lập (X). Nó có thể sử dụng cho các trường hợp chúng ta muốn dự đoán một số lượng liên tục. Ví dụ dự đoán thời gian người đọc dừng lại một trang nào đó hay số người đã truy cập vào một website,... Thông qua việc thu thập dữ liệu thực tế, chúng ta ước lượng hàm hồi quy của tổng thể, đó là ước lượng các tham số của tổng thể.

Hồi quy tuyến tính bội là phần mở rộng của hồi quy tuyến tinh đơn. Nó được sử dụng khi chúng ta muốn dự đoán giá trị của một biến phản hồi dựa trên giá trị của hai hoặc nhiều biến giải thích.

Mô hình hồi quy tuyến tính bội có dạng tổng quát như sau:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + u$$

Trong đó:

Y: Biến phụ thuộc
X_i: Biến độc lập
β_i: Hệ số hồi quy riêng
β_0: Hệ số tự do (hệ số chặn)
u: Hạng nhiễu ngẫu nhiên

1.1.1 Hàm hồi quy tổng thể (PRF - Population Regression Function)

Với Y là biến phụ thuộc và $X_1, X_2, \ldots, X_n$ là biến độc lập, Y là ngẫu nhiên và có một phân phối xác suất nào đó. Ta có:

$$F(X_1, X_2, \ldots, X_n) = E(Y|X_1, X_2, \ldots, X_n)$$

là hàm hồi quy tổng thể của Y theo $X_1, X_2, \ldots, X_n$.

Nếu $F(X)$ tuyến tính, ta có hàm hồi quy tổng thể có dạng tương tự phương trình (1):

$$F(X_1, X_2, \ldots, X_n) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n + u$$

1.1.2 Hàm hồi quy mẫu (SRF - Sample Regression Function)

Vì không biết tổng thể nên không biết giá trị trung bình tổng thể của biến phụ thuộc là đúng ở mức độ nào, do vậy chúng ta phải dựa vào dữ liệu mẫu để ước lượng.

Giả sử đã có các mẫu ngẫu nhiên $(Y_1, X_{1,1}, X_{2,1}, \ldots, X_{n,1})$, $(Y_2, X_{1,2}, X_{2,2},, \ldots, X_{n,2}), \ldots,$ hàm hồi quy được xây dựng dựa trên mẫu này được gọi là **hàm hồi quy mẫu**.

Ta có hàm hồi quy mẫu tổng quát được viết dưới dạng như sau:

$$\hat{Y}_i = \hat{\beta}_1 + \hat{\beta}_2 x_{2,i} + \hat{\beta}_3 x_{3,i} + \cdots + \hat{\beta}_n x_{n,i} + \hat{u}_i \ (1)$$

Trong đó $\hat{\beta}_m$ là ước lượng của β_m, $\hat{u}_i$ là ước lượng của u_i. Chúng ta mong đợi $\hat{\beta}_m$ là ước lượng không chênh lệch của β_m, hơn nữa phải là một ước lượng hiệu quả.

Ước lượng SRF giúp ta ước lượng các tham số của F qua việc tìm các tham số của $\hat{F}$ và lấy giá trị quan sát của các tham số này làm giá trị xấp xỉ cho tham số của F.

1.1.3 Phương pháp bình phương nhỏ nhất (Ordinary Least Squares)

Các giả thiết của phương pháp bình phương nhỏ nhất cho mô hình hồi quy tuyến tính bội như sau:

- Hàm hồi quy là tuyến tính theo các tham số.
  Điều này có nghĩa là quá trình thực hành hồi quy trên thực tế được miêu tả bởi mối quan hệ dưới dạng: $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \beta_n x_n + u$ hoặc mối quan hệ thực tế có thể được viết lại, ví dụ như dưới dạng lấy logarit cả hai vế.

- Kỳ vọng của các yếu tố ngẫu nhiên: $u_i = 0$.
  Trung bình tổng thể sai số bằng 0. Nghĩa là có một số giá trị sai số mang dấu dương và một số sai số mang dấu âm. Do hàm xem là đường trung bình nên giả định các sai số ngẫu nhiên trên sẽ loại trừ lẫn nhau ở mức trung bình trong tổng thể.

- $Cov(u_i, u_j) = 0$: Các sai số độc lập với nhau.

- $Var(u_i) = \sigma^2$: Các sai số có phương sai bằng nhau.
  Tất cả các giá trị u được phân phối giống nhau với cùng phương sai σ^2 sao cho $Var(u_i) = E(u_i^2) = \sigma^2$.

- Các sai số có phân phối chuẩn.
  Điều này rất quan trọng khi phát sinh khoảng tin cậy và thực hiện kiểm định giả thuyết trong những phạm vị mẫu là nhỏ. Nhưng nếu phạm vi mẫu lớn hơn, điều này trở nên không còn quan trọng

1.1.4 Độ phù hợp của mô hình

Để có thể biết mô hình giải thích được như thế nào hay bao nhiêu % biến động của biến phụ thuộc, người ta sử dụng R^2.

Ta có:

- $\sum(y_i - \bar{y})^2$: TSS - Total Sum of Squares.
- $\sum(\hat{y}_i - \bar{y})^2$: ESS - Explained Sum of Squares.
- $\sum e_i^2$: RSS - Residual Sum of Squares.

Có thể viết lại thành: TSS = ESS + RSS.

Ý nghĩa của các thành phần:

- TSS là tổng bình phương của tất cả các sai lệch giữa các giá trị quan sát Y_i và giá trị trung bình.
- ESS là tổng bình phương của tất cả các sai lệch giữa các giá trị của biến phụ thuộc Y nhận được từ hàm hồi quy mẫu và giá trị trung bình của chúng. Phần này đo độ chính xác của hàm hồi quy.
- RSS là tổng bình phương của tất cả các sai lệch giữa các giá trị quan sát Y và các giá trị nhận được từ hàm hồi quy.

- TSS được chia thành 2 phần: một phần do ESS và một phần do RSS gây ra.

R^2 được xác định theo công thức:

$$R^2 = \frac{ESS}{TSS} = 1 - \frac{RSS}{TSS}$$

Tỷ số giữa tổng biến thiên được giải thích bởi mô hình cho tổng bình phương cần được giải thích gọi là hệ số xác định hay là trị thống kê *good of fit*. Từ định nghĩa R^2 chúng ta thấy R^2 đo tỷ lệ hay số của toàn bộ sai lệch Y với giá trị trung bình được giải thích bằng mô hình. Khi đó chúng ta sử dụng R^2 để đo sự phù hợp của hàm hồi quy:

- $0 \leqslant R^2 \leqslant 1$.
- R^2 cao nghĩa là mô hình ước lượng được giải thích được một mức độ cao biến động của biến phụ thuộc.
- Nếu $R^2 = 1$, nghĩa là đường hồi quy giải thích 100 sự thay đổi của Y.
- Nếu $R^2 = 0$, nghĩa là mô hình không đưa ra thông tin nào về sự thay đổi của biến phụ thuộc Y.

1.1.5 Khoảng tin cậy và kiểm định các hệ số hồi quy

1.1.5.a Ước lượng khoảng tin cậy đối với các hệ số hồi quy

Với các giả thiết OLS, u_i có phân phối $N(0, \sigma^2)$. Các hệ số ước lượng tuân theo phân phối chuẩn:

$$\hat{\beta}_j \sim N(\hat{\beta}_j, Se(\hat{\beta}_j))$$

Ước lượng phương sai sai số dựa vào các phần dư bình phương tối thiểu. Trong đó k là hệ số có trong phương trình hồi quy đa biến:

$$\hat{\sigma}^2 = \frac{\Sigma e_i^2}{n-k}$$

Ước lượng 2 phía, tìm được $t_{\frac{\alpha}{2}}(n-k)$ thỏa mãn:

$$P = \left(-t_{\frac{\alpha}{2}}(n-k)\right) \leq \frac{\hat{\beta}_j - \beta_j}{Se(\hat{\beta}_j)} \leq P\left(t_{\frac{\alpha}{2}}(n-k)\right) = 1 - \alpha$$

Khoảng tin cậy $1-\alpha$ của β_j là:

$$\left[\hat{\beta}_j - t_{\frac{\alpha}{2}}(n-k).(\hat{\beta}_j); \hat{\beta}_j + t_{\frac{\alpha}{2}}(n-k).(\hat{\beta}_j)\right]$$

1.1.5.b Kiểm định giả thuyết đối với $\hat{\beta}_j$

Kiểm định ý nghĩa thống kê của các hệ số hồi quy có ý nghĩa hay không: kiểm định rằng biến giải thích có thực sự ảnh hưởng đến biến phụ thuộc hay không. Nói cách khác là hệ số hồi quy có ý nghĩa thống kê hay không

Có thể đưa ra giả thuyết nào đó đối với β_j , chẳng hạn β_j^*. Nếu giả thuyết này đúng thì:

$$T = \frac{\hat{\beta}_j - \beta_j}{Se(\hat{\beta}_j)} \sim T(n-k)$$

Ta có bảng sau:

Loại giả thuyết	Giả thuyết H_0	Giả thuyết đối H_1	Miền bác bỏ
Hai phía	$\beta_1 = \beta_i^*$	$\beta_i \neq \beta_i^*$	$\|t\| > t_{\alpha/2;n-k}$
Phía Phải	$\beta_1 \leq \beta_i^*$	$\beta_i > \beta_i^*$	$t > t_{\alpha;n-k}$
Phía trái	$\beta_1 \geq \beta_i^*$	$\beta_i > \beta_i^*$	$t < -t_{\alpha;n-k}$

Bảng 1: Bảng tóm tắt giả thuyết và miền bác bỏ tương ứng

Ta có thể sử dụng giá trị $P-value$: $P-value <$ mức ý nghĩa thì bác bỏ giả thuyết H_0.

Kiểm định β_j:

- Giả thuyết H_0: $\beta_j = 0 \Leftrightarrow x_j$ không tác động.
- Giả thuyết $H_1 : \beta_j \neq 0 \Leftrightarrow x_j$ có tác động.
- $\beta_j < 0 \Leftrightarrow x_j$ có tác động ngược.
- $\beta_j > 0 \Leftrightarrow x_j$ có tác động thuận.

1.1.6 Kiểm định mức độ ý nghĩa chung của mô hình (trường hợp đặc biệt của kiểm định WALD)

1.1.6.a Khái quát về kiểm định WALD

Giả sử chúng ta có 2 mô hình sau:

$$(U) : Y = \beta_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + u$$
$$(R) : Y = \beta_1 + \beta_2 X_2 + v$$

Mô hình U được gọi là mô hình không giới hạn (Unrestrict), và mô hình R được gọi là mô hình giới hạn (Restrict). Đó là do β_3 và $|beta_4$ buộc phải bằng 0 trong mô hình R. Ta có thể kiểm định giả thuyết liên kết $\beta_3 = \beta_4 = 0$ với giả thuyết đối là ít nhất một trong những hệ số này không bằng 0. Kiểm định giả thuyết liên kết này được gọi là kiểm định Wald, thủ tục như sau:

Đặt các mô hình giới hạn và không giới hạn là:

$$(U): Y = \beta_1 + \beta_2 X_2 + ... + \beta_m X_m + \beta_{m+1} X_{m+1} + ... + \beta_k X_k + u$$
$$(R): Y = \beta_1 + \beta_2 X_2 + ... + \beta_m X_m + v$$

Mô hình (R) có được bằng cách bỏ bớt một số biến ở mô hình (U), đó là: $X_{m+1}, X_{m+2}, ..., X_k$

Giả thuyết $H_0 : \beta_{m+1} = ... = \beta_k = 0$

Giả thuyết H_1 : Các tham số không đồng thời bằng 0

Lưu ý rằng (U) chứa k hệ số hồi quy chưa biết và (R) chứa m hệ số hồi quy chưa biết. Do đó, mô hình R có ít hơn thông số so với U. Câu hỏi chúng ta nêu ra là biến bị loại ra có ảnh hưởng ý nghĩa đối với Y hay không.

Trị thống kê kiểm định đối với giả thuyết là:

$$F_c = \frac{[RSS_R - RSS_U]/(k-m)}{RSS_U/(n-k)} \sim F(\alpha, k-m, n-k) = \frac{R_U^2 - R_R^2/(k-m)}{1-R_U^2/(n-k)}$$

Với R^2 là số đo độ thích hợp không hiệu chỉnh.

Với giả thuyết không, F_c có phân phối F với $k-m$ bậc tự do đối với tử số và $n-k$ bậc tự do đối với mẫu số.

Ta bác bỏ giả thuyết H_0 khi:

$$F_c > F(\alpha, k-m, n-k)$$

Hoặc giá trị $P - value$ của thống kế F nhỏ hơn mức ý nghĩa cho trước.

1.1.6.b Kiểm định ý nghĩa của mô hình

Trong mô hình hồi quy đa biến, giả thuyết "không" cho rằng mô hình không có ý nghĩa được hiểu là tất cả các hệ số hồi quy riêng đều bằng 0.

Ứng dụng kiểm định Wald (thường được gọi là kiểm định F) được tiến hành cụ thể như sau:

- Bước 1: Giả thuyết $H_0 : \beta_2 = \beta_3 = ... = \beta_k = 0$.

  Giả thuyết H_1 : có ít nhất một trong những giá trị β khác không.

- Bước 2: : Trước tiên hồi quy Y theo một số hạng không đổi và $X_2, X_3, ..., X_k$, sau đó tính tổng bình phương sai số RSS_U, RSS_R. Phân phối F là tỷ số của hai biến ngẫu nhiên phân phối khi bình phương độc lập. Điều này cho ta trị thống kê:

  $$F_c = \frac{[RSS_R - RSS_U]/(k-m)}{RSS_U/(n-k)} \sim F(\alpha, k-m, n-k)$$

  Vì $H_0 : \beta_2 = \beta_3 = ... = \beta_k = 0$, nhận thấy rằng trị thống kê kiểm định đối với giả thuyết này sẽ là:

  $$F_c = \frac{ESS/(k-1)}{RSS/(n-k)} \sim F(\alpha, k-1, n-k)$$

- Bước 3: Tra số liệu trong bảng F tương ứng với bậc tự do $(k-1)$ cho tử số và $(n-k)$ cho mẫu số, và với mức ý nghĩa α cho trước.

- Bước 4: Bác bỏ giả thuyết H_0 ở mức ý nghĩa α nếu $F_c > F(\alpha, k-1, n-k)$.

  Đối với phương pháp giá trị $P - value$, tính giá trị $p = P(F > F_c | H_0)$ và bác bỏ giả thuyết H_0 nếu p bé hơn mức ý nghĩa α.

1.2 Lý thuyết về ANOVA (Phân tích phương sai)

Phân tích phương sai (Analysis of Variance) hay còn gọi là kiểm định ANOVA là một kỹ thuật thống kê tham số được sử dụng để so sánh các bộ dữ liệu. Mục tiêu của phân tích phương sai là so sánh trung bình của nhiều nhóm (tổng thể) dựa trên các trị trung bình của các mẫu quan sát từ các nhóm này và thông qua kiểm định giả thuyết của kết luận về sự bằng nhau của các trung bình tổng thể này. Phân tích phương sai được sử dụng như một công cụ để xem xét ảnh hưởng của một yếu tố nguyên nhân (định tính) đến một yếu tố kết quả (định lượng).

1.2.1 Phân tích phương sai một yếu tố

Phân tích phương sai một yếu tố (One-way ANOVA) là phân tích ảnh hưởng của một yếu tố nguyên nhân (dạng biến định tính) ảnh hưởng đến một yếu tố kết quả (dạng biến định lượng) đang nghiên cứu.

1.2.1.a Trường hợp k tổng thể có phân phối bình thường và phương sai bằng nhau

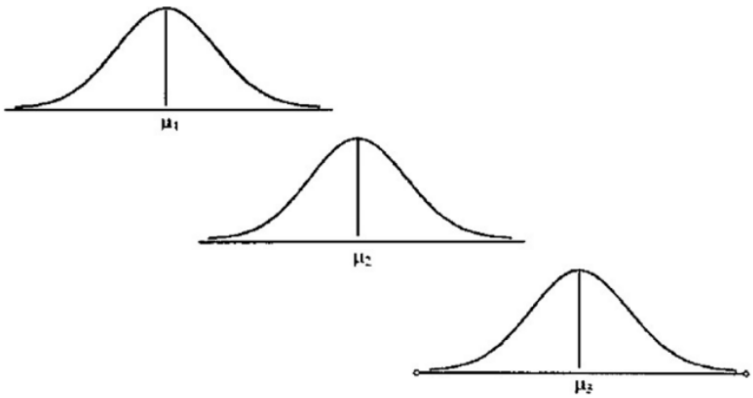
Giả sử rằng chúng ta muốn so sánh trung bình của k tổng thể (với ví dụ trên thì $k = 3$) dựa trên những mẫu ngẫu nhiên độc lập gồm $n_1, n_2, n_3, ..., n_k$ quan sát từ k tổng thể. Cần ghi nhớ ba giả định sau đây về các nhóm tổng thể được tiến hành phân tích ANOVA:

- Các tổng thể này có phân phối chuẩn.
- Các phương sai tổng thể bằng nhau.
- Các quan sát được lấy mẫu là độc lập nhau.

Nếu trung bình của các tổng thể được ký hiệu là $\mu_1, \mu_2, ..., \mu_k$ thì khi các giả định trên được đáp ứng, mô hình phân tích phương sai một yếu tố ảnh hưởng được mô tả dưới dạng kiểm định giả thuyết như sau: $H_0 = \mu_1 = \mu_2 = \mu_k$.

Giả thuyết H_0 cho rằng trung bình của k tổng thể đều bằng nhau (về mặt nghiên cứu liên hệ thì giả thuyết này cho rằng yếu tố nguyên nhân không có tác động gì đến vấn đề ta đang nghiên cứu). Và giả thuyết đối là H_1: Tồn tại ít nhất một cặp trung bình tổng thể khác nhau.

Hai giả định đầu tiên để tiến hành phân tích phương sai được mô tả như hình dưới đây, ta có thể thấy thấy ba tổng thể đều có phân phối chuẩn với mức độ phân tán tương đối giống nhau, nhưng ba vị trí chênh lệch của chúng cho thấy ba trị trung bình khác nhau. Rõ ràng nếu ta thực sự có các giá trị của 3 tổng thể và biểu diễn được phân phối của chúng như hình dưới thì ta có thể ngay lập tức kết luận bác bỏ H_0, hay 3 tổng thể này có trị trung bình khác nhau.



Hình 1. Mô hình phân phối của các tổng thể

Tuy vậy, ta chỉ có mẫu đại diện được quan sát, nên để kiểm định giả thuyết này, ta cần thực hiện các bước như sau:

Bước 1: Tính các trung bình mẫu của các nhóm (xem như đại diện của các tổng thể).

Trước hết ta xem cách tính các trung bình mẫu từ những quan sát của k mẫu ngẫu nhiên độc lập (ký hiệu $\overline{x_1}, \overline{x_2}, ...$) và trung bình chung của k mẫu quan sát (ký hiệu $\overline{x}$) từ trường hợp tổng quát như sau:

Tổng thể				
1	2	3	...	k
x_{11}	x_{21}	x_{31}	...	x_{k1}
x_{12}	x_{22}	x_{32}	...	x_{k2}
...	...	...	...	...

Bảng 2. Bảng số liệu tổng quát thực hiện phân tích phương sai

Tính trung bình mẫu của từng nhóm $\overline{x_1}, \overline{x_2}$ theo công thức:

$$\overline{x_i} = \frac{\sum_{j=1}^{n_i} x_i j, (i=1,2,3,...,k)}{n_i}$$

Và trung bình chung của k mẫu (trung bình chung của toàn bộ mẫu khảo sát):

$$\overline{x_i} = \frac{\sum_{i=1}^{k} n_i \overline{x_1}}{\sum_{i=1}^{k} n_i}$$

Bước 2: Tính các tổng các chênh lệch bình phương (hay gọi tắt là tổng bình phương). Tính tổng các chênh lệch bình phương trong nội bộ nhóm - SSW và tổng các chênh lệch bình phương giữa các nhóm - SSG.

Tổng các chênh lệch bình phương trong nội bộ nhóm (SSW) được tính bằng cách cộng các chênh lệch bình phương giữa các giá trị quan sát với trung bình mẫu của từng nhóm,

rồi sau đó lại tính tổng cộng kết quả tất cả các nhóm lại. SSW phản ánh phần biến thiên của yếu tố kết quả do ảnh hưởng của các yếu tố khác, chứ không phải do yếu tố nguyên nhân đang nghiên cứu (là yếu tố dùng để phân biệt các tổng thể/ nhóm đang so sánh)

Viết tổng quát theo công thức ta có:

$$\text{SSW} = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (x_{ij} - \overline{x_i})^2$$

Tổng các chênh lệch bình phương giữa các nhóm (SSG) được tính bằng cách cộng các chênh lệch được lấy bình phương giữa các trung bình mẫu của từng nhóm với trung bình chung của k nhóm (các chênh lệch này đều được nhân thêm với số quan sát tương ứng với từng nhóm). SSG phản ánh phần biến thiên của yếu tố kết quả do ảnh hưởng của yếu tố nguyên nhân đang nghiên cứu.

$$\text{SSG} = \sum_{i=1}^{k} n_i (\overline{x_i} - \overline{x})^2$$

Tổng các chênh lệch bình phương toàn bộ (SST) được tính bằng cách cộng tổng các chênh lệch đã lấy bình phương giữa từng giá trị quan sát của toàn bộ mẫu nghiên cứu (x_{ij}) với trung bình toàn bộ (x). SST phản ánh biến thiên của yếu tố kết quả do ảnh hưởng của tất cả các nguyên nhân.

$$\text{SST} = \sum_{i=1}^{k} \sum_{j=1}^{n} (x_{ij} - \overline{x})^2$$

Có thể dễ dàng chứng minh là tổng các chênh lệch bình phương toàn bộ bằng tổng cộng tổng các chênh lệch bình phương trong nội bộ các nhóm và tổng các chênh lệch bình phương giữa các nhóm.

$$\text{SST} = \text{SSW} + \text{SSG}$$

Như vậy công thức trên cho thấy, SST là toàn bộ biến thiên của yếu tố kết quả đã được phân tích thành hai phần: phần biến thiên do yếu tố đang nghiên cứu tạo ra (SSG) và phần biến thiên còn lại do các yếu tố khác không nghiên cứu ở đây tạo ra (SSW). Nếu phần biến thiên do yếu tố nguyên nhân đang xét tạo ra càng đáng kể so với phần biến thiên do các yếu tố khác không cét tạo ra, thì chúng ta càng có cơ sở để bác bỏ H_0 và kết luận là yếu tố nguyên nhân đang nghiên cứu ảnh hưởng có ý nghĩa đến yếu tố kết quả.

Bước 3: Tính các phương sai (là trung bình của các chênh lệch bình phương).

Các phương sai được tính bằng cách lấy các tổng chênh lệch bình phương chia cho bậc tự do tương ứng. Tính phương sai trong nội bộ nhóm (MSW) bằng cách lấy tổng các chênh lệch bình phương trong nội bộ các nhóm (SSW) chia cho bậc tự do tương ứng là $n \check{} k$ (n là số quan sát, k là số nhóm so sánh). MSW là ước lượng phần biến thiên của yếu tố kết quả do các yếu tố khác gây ra.

$$\text{MSW} = \frac{SSW}{n-k}$$

Tính phương sai giữa các nhóm (MSG) bằng cách lấy tổng các chênh lệch bình phương

giữa các nhóm chia cho bậc tự do tương ứng là $k-1$. MSG là ước lượng phần biến thiên của yếu tố kết quả do yếu tố nguyên nhân đang nghiên cứu gây ra.

$$\text{MSG} = \frac{SSG}{k-1}$$

Bước 4: Kiểm định giả thuyết. Giả thuyết về sự bằng nhau của k trung bình tổng thể được quyết đinh dựa trên tỉ số của hai phương sai: phương sai giữa các nhóm (MSG) và phương sai trong nội bộ nhóm (MSW). Tỉ số này gọi là tỉ số F vì nó tuân theo định luật Fisher – Snedecor với bậc tự do $k-1$ ở tử số và $n-k$ ở mẫu số.

$$\text{F} = \frac{MSG}{MSW}$$

Ta bác bỏ giả thuyết H_0 cho rằng trị trung bình của k tổng thể bằng nhau khi:

$$F > F_{(k-1;n-k;\alpha)}$$

$F > F_{(k-1;n-k;\alpha)}$ là giá trị giới hạn tra từ bảng Fisher với bậc tự do $k-1$ tra theo hàng đầu tiên và $n-k$ tra theo cột đầu tiên, và cần chọn bảng với mức ý nghĩa phù hợp.

Nguồn biến thiên	**Tổng chênh lệch bình phương**	**Bậc tự do**	**Phương sai**	**Tỉ số F**
Giữa các nhóm	SSG	k-1	MSG = $\frac{SSG}{k-1}$	$\frac{MSG}{MSW}$
Trong nội bộ cá nhóm	SSW	n-k	MSW = $\frac{SSW}{n-k}$	
Toàn bộ	SST	n-1		

Bảng 3. Bảng kết quả tổng quát của ANOVA

1.2.1.b Kiểm tra các giả định của phân tích phương sai

Chúng ta có thể kiểm tra nhanh các giả định này bằng đồ thị. Histogram là phương pháp tốt nhất để kiểm tra giả định về phân phối bình thường của dữ liệu nhưng nó đòi hỏi một số lượng quan sát khá lớn. Biểu đồ thân lá hay biểu đồ box and whiskers là một thay thế tốt trong tình huống số quan sát ít hơn. Nếu công cụ đồ thị cho thấy tập dữ liệu mẫu khá phù hợp với phân phối bình thường thì ta có thể xem giả định phân phối bình thường đã thỏa mãn.

Một phương pháp kiểm định tham số chắc chắn hơn cho giả định phương sai bằng nhau là kiểm định Levene về phương sai của các tổng thể. Kiểm định này xuất phát từ giả thuyết sau.

$$H_0 = \sigma_1^2 = \sigma_2^2 = \sigma_k^2$$

H_1: Không phải tất cả các phương sai bằng nhau.

Để quyết định chấp nhận hay bác bỏ H_0 ta tính toán giá trị kiểm định F theo công thức:

$$F_{max} = \frac{{S_{max}}^2}{{S_{min}}^2}$$

Trong đó ${S_{max}}^2$ là phương sai lớn nhất trong các nhóm nghiên cứu và ${S_{min}}^2$ là phương sai nhỏ nhất trong các nhóm nghiên cứu. Giá trị F tính được được đem so sánh với giá trị $F_{(k;df;\alpha)}$ tra được từ bảng phân phối Hartley F_{max}.

Quy tắc quyết định: $F_{(k;df;\alpha)}$ thì bác bỏ giả thuyết H_0 cho rằng phương sai bằng nhau và ngược lại.

Nếu ta không chắc chắn về các giả định hoặc nếu kết quả kiểm định cho thấy các giả định không được thỏa mãn thì một phương pháp kiểm định thay thế cho ANOVA là phương pháp kiểm định phi tham số Kruskal – Wallis sẽ được áp dụng.

1.2.1.c Phân tích sau ANOVA

Mục đích của phân tích phương sai là kiểm định giả thuyết H_0 rằng trung bình của tổng thể bằng nhau. Sau khi phân tích và kết luận, có hai trường hợp xảy ra là chấp thuận giả thuyết H_0 hoặc bác bỏ giả thuyết H_0.

Nếu chấp nhận giả thuyết H_0 thì phân tích kết thúc.

Nếu bác bỏ giả thuyết H_0, ta kết luận trung bình của các tổng thể không bằng nhau. Vì vậy, vấn đề tiếp theo là phân tích sâu hơn để xác minh nhóm (tổng thể) nào khác nhóm nào, nhóm nào có trung bình lớn hơn hay nhỏ hơn.

Có nhiều phương pháp để tiếp tục phân tích sâu ANOVA khi bác bỏ giả thuyết H_0. Trong phần này chỉ đề cập đến một phương pháp thông dụng đó là phương pháp Tukey, phương pháp này còn được gọi là kiểm định HSD (Honestly Significant Differences). Nội dung của phương pháp này là so sánh từng cặp các trung bình nhóm ở mức ý nghĩa nào đó cho tất cả các cặp kiểm định có thể để phát hiện ra những nhóm khác nhau. Nếu có k nhóm nghiên cứu và chúng ta so sánh tất cả các cặp nhóm thì số lượng cặp cần phải so sánh là tổ hợp chập 2 của k nhóm.

$$c_k^2 = \frac{k!}{2!(k-2)!} = \frac{k(k-1)}{2}$$

Giá trị giới hạn Tukey được tính theo công thức:

$$T = q_{\alpha,k,n-k}\sqrt{\frac{MSW}{n_i}}$$

Trong đó:

- $q_{\alpha,k,n-k}$ là giá trị tra bảng phân phối kiểm định Tukey ở mức ý nghĩa α, với bậc tự do k và $n-k$, với n là tổng số quan sát mẫu ($n = \sum ni$).

- MSW là phương sai trong nội bộ nhóm.
- n_i là số quan sát trong một nhóm (tổng thể), trong trường hợp mỗi nhóm có số quan sát n_i khác nhau, sử dụng giá trị n_i nhỏ nhất.

Tiêu chuẩn quyết định là bác bỏ giả thuyết H_0 khi độ lệch tuyệt đối giữa các cặp trung bình mẫu lớn hơn hay bằng T giới hạn.

Bên cạnh việc kiểm định để phát hiện ra những nhóm khác biệt, ta có thể tìm khoảng ước lượng cho chênh lệch giữa các nhóm có khác biệt có ý nghĩa thống kê. Ước lượng khoảng chênh lệch giữa hai trung bình nhóm có khác biệt tính theo công thức:

$$\mu_1 - \mu_2 = (\overline{x_1} - \overline{x_2} \pm (t_{n-k}, \frac{\alpha}{2}\sqrt{\frac{MSW}{n_i}}))$$

Trong đó, t là giá trị được tra từ bảng phân phối Student t với $n - k$ bậc tự do.

Phân tích phương sai với kiểm định F chỉ có thể áp dụng khi các nhóm so sánh có phân phối bình thường và phương sai bằng nhau. Trong trường hợp không thỏa điều kiện này, ta có thể chuyển đổi dữ liệu của yếu tố kết quả từ dạng định lượng về dạng định tính (dữ liệu thứ bậc) và áp dụng một kiểm định phi tham số phù hợp tên là Kruskal – Wallis.

1.2.1.d Phương pháp phân tích phương sai một yếu tố Kruskal - Wallis bằng thứ hạng

Khi các nhóm so sánh không thỏa mãn các điều kiện có phân phối chuẩn và phương sai bằng nhau, ta không thể sử dụng phương pháp ANOVA thông thường cũng như phương pháp Tukey. Ở đây ta sẽ đề xuất một phương pháp thay thế là phương pháp Kruskal - Wallis.

Trong phương pháp Kruskal - Wallis, mỗi quan sát trong tổng số N quan sát được thay thế bởi một số điểm để xếp hạng, với điểm thấp nhất có hạng 1, ..., điểm cao nhất có hạng N. Tổng của hạng sẽ được tính cho từng nhóm. Phương pháp Kruskal - Wallis giúp xác định rằng liệu tổng này có khác biệt đáng kể đến mức chúng không thể được lấy từ chung một nhóm.

Ta chứng minh được rằng nếu k nhóm được lấy từ chung 1 tổng thể, tức là giả thiết H_0 đúng, vậy hàm H được định nghĩa bởi dưới đây sẽ phân phối theo chi - bình phương (chi - square) với $df = k - 1$. Ta cần lưu ý rằng kích thước của các nhóm này không được quá nhỏ.

$$\mathrm{H} = \frac{12}{N(N+1)} \sum_{j=1}^{k} \frac{R_j^2}{n_j} - 3(N+1)$$

Trong đó:

- k: Số nhóm.

- n_j: Kích thước của nhóm thứ j.
- N: Tổng n_j, tổng kích thước các nhóm.
- R_j: Tổng hạng của nhóm thứ j.

Khi giá trị quan sát được của H lớn hơn hoặc bằng giá trị của chi - bình phương với mức ý nghĩa cho trước và giá trị quan sát của $df = k - 1$, vậy ta có thể bác bỏ giả thiết H_0.

1.3 Mô hình Random Forest

1.3.1 Giới thiệu chung

Random Forest hoạt động dựa trên hai nguyên lý chính: Bootstrap Aggregating (Bagging) và Tính ngẫu nhiên trong lựa chọn đặc trưng. Mô hình Random Forest là một kỹ thuật học máy mạnh mẽ và phổ biến, thuộc nhóm các phương pháp học tập không giám sát.

1.3.2 Nguyên lý hoạt động

- Tạo Các Cây Quyết Định:
  - Bootstrap Sampling: Mỗi cây trong rừng được tạo ra từ một mẫu ngẫu nhiên của dữ liệu huấn luyện. Phương pháp này gọi là "Bootstrap sampling". Điều này có nghĩa là một số mẫu có thể được chọn nhiều lần, trong khi một số khác có thể không được chọn.
  - Chọn Đặc Trưng Ngẫu Nhiên: Khi tạo từng cây, Random Forest chọn ngẫu nhiên một tập hợp các đặc trưng từ tổng thể. Việc này giúp giảm sự tương quan giữa các cây, làm cho mô hình mạnh mẽ hơn.
- Dự Đoán: Đối với từng mẫu mới, Random Forest sẽ để mỗi cây đưa ra dự đoán. Trong trường hợp phân loại, kết quả cuối cùng sẽ là kết quả mà nhiều cây nhất chọn (bỏ phiếu đa số). Đối với hồi quy, kết quả sẽ là giá trị trung bình của tất cả các cây.

1.3.3 Ưu điểm

- Chống Quá Khớp: Nhờ vào việc tổng hợp nhiều cây, Random Forest có khả năng giảm thiểu hiện tượng quá khớp (overfitting) mà một cây quyết định đơn lẻ có thể gặp phải.
- Độ Chính Xác Cao: Nhiều nghiên cứu cho thấy Random Forest thường cho kết quả chính xác hơn so với nhiều thuật toán khác, đặc biệt trong các bài toán phân loại.
- Xử Lý Dữ Liệu Thiếu: Mô hình này có khả năng xử lý tốt các giá trị thiếu trong dữ liệu mà không cần phải loại bỏ chúng.

1.3.4 Nhược điểm

- Thời Gian Huấn Luyện: Việc tạo ra nhiều cây có thể tốn thời gian và tài nguyên tính toán, đặc biệt với dữ liệu lớn.
- Khó Giải Thích: Mặc dù mỗi cây quyết định có thể dễ hiểu, nhưng khi kết hợp nhiều cây, sẽ khó khăn hơn để giải thích mô hình tổng thể và cách mà nó đưa ra quyết định.

1.4 Mô hình Decision Tree

1.4.1 Giới thiệu chung

Decision Tree (cây quyết định) là một mô hình học máy (machine learning) được sử dụng phổ biến trong các bài toán phân loại (classification) và hồi quy (regression). Về bản chất, cây quyết định là một cấu trúc dạng cây, trong đó mỗi nút trong (internal node) đại diện cho một thuộc tính kiểm tra, mỗi nhánh (branch) tương ứng với kết quả của phép kiểm tra đó, và mỗi nút lá (leaf node) biểu thị một nhãn hoặc giá trị dự đoán.

1.4.2 Nguyên lý hoạt động

Các giải thuật dùng để xây dựng decision tree được xây dựng dựa trên nguyên lý chia để trị (divide-and-conquer), lặp đi lặp lại việc phân tách tập dữ liệu theo thuộc tính sao cho thông tin đạt được là tối đa tại mỗi bước. Một tiêu chí phổ biến để lựa chọn thuộc tính phân tách là Entropy và Information Gain, hoặc Gini Impurity trong thuật toán CART. Quá trình xây dựng cây kết thúc khi dữ liệu tại nút đạt mức thuần nhất, hoặc khi không còn thuộc tính nào có thể phân tách tiếp.

1.4.2.a Cấu trúc

Chúng ta có nhiều cách nhìn nhận về cấu trúc của decision tree nhưng nhìn chung cấu trúc sẽ gồm các thành phần sau:

- Nút gốc (Root Node): Chứa toàn bộ tập dữ liệu ban đầu.
- Nút quyết định (Internal Nodes): Thực hiện các phép kiểm tra trên các thuộc tính.
- Nút lá (Leaf Nodes): Đại diện cho kết quả phân loại (ví dụ: 0 hoặc 1).

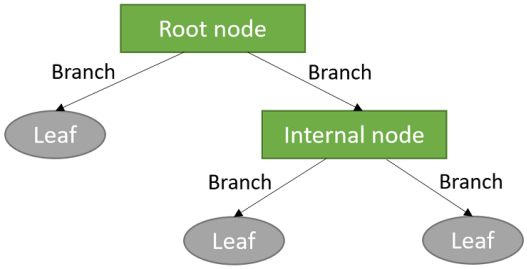


Figure 1: Cấu trúc Decision Tree (Nguồn: https://statlect.com/machine-learning/decision-tree)

1.4.3 Thuật toán xây dựng

1.4.3.a Khái niệm chính

Decision Tree là một mô hình sử dụng các thuật toán đệ quy để phân chia dữ liệu thành các tập con dựa trên giá trị của đặc trưng, nhằm tối ưu hóa độ đo "impurity" (độ không thuần khiết) như chỉ số Gini, entropy (cho bài toán phân loại) hoặc phương sai (cho bài toán hồi quy). Mỗi nút trong cây đại diện cho một quyết định dựa trên đặc trưng, và các nhánh dẫn đến các tập con dữ liệu thuần khiết hơn, hỗ trợ dự đoán chính xác.

1.4.3.1 Độ đo impurity phổ biến

Entropy và Information Gain Entropy là khái niệm quan trọng trong vật lý nhằm để thể hiện mức độ hỗn loạn. Trong machine learning thì Entropy sẽ biểu thị mức độ hỗn loạn hay sự không thuần khiết của các nhãn lớp trong một tập mẫu dữ liệu.

Công thức: Giả sử một tập hợp dữ liệu S có k lớp, entropy được tính như sau:

$$H(S) = -\sum_{i=1}^{k} p_i \log_2(p_i)$$

Trong đó:

- p_i: Tỷ lệ các mẫu thuộc lớp i trong tập S
- k: số lượng các lớp trong tập dữ liệu
- S: là tập dữ liệu

Information Gain:

$$IG(S, A) = H(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} H(S_v)$$

Trong đó:

- Values(A): Tập các giá trị của thuộc tính A
- S_v: Tập con của S với giá trị v của thuộc tính A

Gini Impurity:

$$G(S) = 1 - \sum_{i=1}^{k} p_i^2$$

Trong đó:

- p_i: Tỷ lệ mẫu thuộc lớp i
- k: Số lớp trong tập dữ liệu

1.4.4 Một số thuật toán phổ biến

- **ID3**
  - *Mã giả*:
    1. Tính Information Gain cho mỗi thuộc tính
    2. Chọn thuộc tính có IG lớn nhất làm nút gốc
    3. Lặp lại đệ quy cho các tập con
  - *Đặc trưng*:
    - Chỉ xử lý dữ liệu rời rạc
    - Dễ bị overfitting do không có cắt tỉa
- **C4.5**
  - *Cải tiến từ ID3*:
    - Xử lý dữ liệu liên tục bằng ngưỡng
    - Hỗ trợ giá trị thiếu (missing values)
    - Dùng Gain Ratio thay vì Information Gain
- **CART**
  - *Đặc trưng chính*:
    - Dùng Gini Impurity cho phân loại
    - Hỗ trợ hồi quy (MSE)
    - Xây dựng cây nhị phân
  - *Mã giả tỉa cây*:
    1. Tính cost-complexity $R_\alpha(T)$
    2. Chọn subtree có α tối ưu

1.4.5 Điều kiện dừng

Theo Scikit-learn documentation (Pedregosa et al., 2011), quá trình xây dựng cây dừng khi:

- Đạt max_depth (độ sâu tối đa)
- Số mẫu trong nút $<$ min_samples_split

1.5 Ưu nhược điểm và thách thức

1.5.1 Ưu nhược điểm của thuật toán

- Ưu điểm:
  - Dễ diễn giải (interpretability)
  - Không cần chuẩn hóa dữ liệu
- Nhược điểm:
  - Nhạy cảm với nhiễu (noise)
  - Dễ overfitting nếu không kiểm soát độ phức tạp

1.5.2 Thách thức

- Việc dữ liệu với nhiều đặc trưng phức tạp mức độ entropy không thuần nhất lớn dẫn tới việc có thể đệ quy vô hạn
- Độ sâu quá lớn cũng dẫn tới việc overfitting dễ xảy ra

1.5.3 Khắc phục Overfitting

- Pruning: Giảm độ phức tạp bằng cách cắt tỉa cây
- Ensemble Methods (Random Forest - Breiman, 2001): Kết hợp nhiều cây để giảm phương sai

2 Tổng quan dữ liệu

Bộ dữ liệu "Internet Advertisements Data Set" là một tập dữ liệu cổ điển trong lĩnh vực học máy, đặc biệt hữu ích cho các bài toán phân loại nhị phân. Mục tiêu chính của bộ dữ liệu này là phân biệt giữa các hình ảnh quảng cáo ("ad") và không phải quảng cáo ("nonad") trên các trang web.
Thông tin chung: thu thập và chuẩn hóa từ UCI Machine Learning Repository. Tại Kaggle, data sẽ được lưu với dạng là một tệp add.csv.
Số lượng mẫu: 3.279 dòng dữ liệu, mỗi dòng đại diện cho một hình ảnh trên trang web.
Số lượng đặc trưng: 1.559 đặc trưng, bao gồm:
- 3 đặc trưng liên tục (continuous features) liên quan đến hình học của hình ảnh .

- Height: Chiều cao của hình ảnh, tính bằng pixel.
- Width: Chiều rộng của hình ảnh, tính bằng pixel.
- Ratio: Tỷ lệ khung hình của hình ảnh, thường được tính bằng cách lấy chiều rộng chia cho chiều cao.

- 1.555 đặc trưng nhị phân (binary features) biểu thị sự xuất hiện của các cụm từ trong URL, văn bản liên kết (anchor text), văn bản thay thế (alt text) và các từ xung quanh liên kết.
- 1 đặc trưng là dùng để phân loại.

- Target: Biến nhị phân cho biết bản ghi có phải là quảng cáo (1) hay không (0).

Đây là một nguồn tài nguyên quý báu dành thường dùng để:

- Huấn luyện và đánh giá các mô hình phân loại nhị phân.
- Nghiên cứu về phát hiện quảng cáo tự động trên các trang web.
- Thực hành xử lý dữ liệu có số lượng đặc trưng lớn và không đồng nhất.

3 Hoạt động

3.1 Tiền xử lý dữ liệu

3.1.1 Đọc dữ liệu và xử lý dữ liệu cột

Dữ liệu được đọc từ file add.csv, sử dụng thư viện dplyr để tiến hành xử lý. Ở dòng đầu tiên được loại bỏ vì mang tính ý nghĩa chỉ số. Các cột từ 4 tới 1558 được gỡ bỏ vì các cột khi phân tích không liên quan. Sau đó tiến hành đổi tên các cột tương ứng từ [1] đến [4] thành [height]; [width]; [ratio] và [target].

```
library(dplyr)
add <- read.csv("C:/Users/Admin/Downloads/add.csv")
head(add, 5)
add <- add[, -1]
add <- add[, -c(4:1558)]
colnames(add)[1] <- "height"
colnames(add)[2] <- "width"
colnames(add)[3] <- "ratio"
colnames(add)[4] <- "target"
head(add)

```

```
> add <- add[, -c(4:1558)]
> colnames(add)[1] <- "height"
> colnames(add)[2] <- "width"
> colnames(add)[3] <- "ratio"
> colnames(add)[4] <- "target"
> head(add)
  height width  ratio target
1    125   125      1    ad.
2     57   468 8.2105    ad.
3     33   230 6.9696    ad.
4     60   468    7.8    ad.
5     60   468    7.8    ad.
6     60   468    7.8    ad.
```

Figure 2: 6 dòng đầu tiên của dữ liệu

3.1.2 Chuyển đổi biến và kiểu dữ liệu

- Biến mục tiêu `target` được mã hóa thành 0 (nonad) và 1 (ad.).
- Các cột `height`, `width`, `ratio` được chuyển sang kiểu số thực.

```
add$target <- ifelse(add$target == "ad.", 1, 0)
table(add$target)
head(add)

```

```
> # Thay thế giá trị của biến mục tiêu 'target' từ "ad." thành 1 và "nonad" thành 0
> add$target <- ifelse(add$target == "ad.", 1, 0)
> table(add$target)

   0    1
2820  459
```

Figure 3: Tần số của các label trong cột target

```
> head(add)
  height width  ratio target
1    125   125      1      1
2     57   468 8.2105      1
3     33   230 6.9696      1
4     60   468    7.8      1
5     60   468    7.8      1
6     60   468    7.8      1
```

Figure 4: 6 phần tử đầu sau khi encoding

3.1.3 Xử lý giá trị thiếu

- Các giá trị dấu hỏi được thay bằng `NA`.

- Tính tổng số giá trị thiếu và tỷ lệ phần trăm theo từng cột.

```
    add[add == "    ?"] <- NA
    add[add == "     ?"] <- NA
    na_counts <- colSums(is.na(add))
    na_percentage <- colMeans(is.na(add)) * 100
    na_summary <- data.frame(
        NA_Count = na_counts,
        NA_Percent = round(na_percentage, 2)
    )
    print(na_summary)

```

```
> # Thống kê số lượng giá trị NA theo từng cột
> na_counts <- colSums(is.na(add))
>
> # Tính tỷ lệ phần trăm NA theo cột
> na_percentage <- colMeans(is.na(add)) * 100
>
> na_summary <- data.frame(
+   NA_Count = na_counts,
+   NA_Percent = round(na_percentage, 2)
+ )
> print(na_summary)
       NA_Count NA_Percent
height      903      27.54
width       901      27.48
ratio       910      27.75
target        0       0.00
>
```

Figure 5: Bảng tổng số phần tử khiếm khuyết và tỉ lệ phần trăm trong mỗi đặc trưng

3.1.4 Loại bỏ hàng trong cột width và height có NA

```
add <- add[!(is.na(add$height) | is.na(add$width)), ]
na_counts <- colSums(is.na(add))
na_percentage <- colMeans(is.na(add)) * 100
na_summary <- data.frame(
  NA_Count = na_counts,
  NA_Percent = round(na_percentage, 2)
)
print(na_summary)

```

	NA_Count	NA_Percent
height	0	0
width	0	0
ratio	0	0
target	0	0

Figure 6: Số phần tử khiếm khuyết NA có trong cột width và height sau khi xử lý

3.1.5 Tìm và xử lý Outliers

- Kiểm tra số hàng hiện có trước khi xử lý outliers.

```
n_rows_before <- nrow(add)
print(n_rows_before)

```

```
> # --- Get the number of rows before outlier removal ---
> n_rows_before <- nrow(add)
> print(n_rows_before)
[1] 2369
```

Figure 7: Số hàng trước khi xử lý outliers

- Viết hàm `replace_outliers` để thay các giá trị ngoại lai bằng `NA` theo quy tắc IQR.

```
  replace_outliers <- function(x) {
    q1 <- quantile(x, 0.25, na.rm = TRUE)
    q3 <- quantile(x, 0.75, na.rm = TRUE)
    iqr <- q3 - q1
    lower_bound <- q1 - 1.5 * iqr
    upper_bound <- q3 + 1.5 * iqr
    x[x < lower_bound | x > upper_bound] <- NA
    return(x)
  }
  df1[c("height", "width","ratio")] <- lapply(
    df1[c("height", "width","ratio")],
    replace_outliers
  )

```

- Sử dụng lapply cho từng cột cụ thể và xem số dòng đã xóa

```
 add[c("height", "width","ratio")] <- lapply(add[c("height", "width","ratio")],
   replace_outliers)
 na_counts <- colSums(is.na(add))

  add <- add[!(is.na(add$height) | is.na(add$width) | is.na(add$ratio)), ]
```

```
n_rows_after <- nrow(add)
rows_removed <- n_rows_before - n_rows_after
print(paste("Tong so dong da xoa do outliers:", rows_removed))
```

```
> print(paste("Tổng số dòng đã xóa do outliers:", rows_removed))
[1] "Tổng số dòng đã xóa do outliers: 368"
```

Figure 8: Số dòng đã xóa sau khi xử lý outliers

3.2 Thống kê mô tả

3.2.1 Thống kê các giá trị mô tả

Ta viết function để tính các thông số như trung Bình, độ lệch chuẩn, giá trị nhỏ nhất, giá trị lớn nhất, phân vị 1, trung vị, phân vị 75.

```
df1=add[, c("height", "width", "ratio")]
summary_stats <- function(x) {
  c(
    count = sum(!is.na(x)),
    mean  = mean(x, na.rm = TRUE),
    std   = sd(x, na.rm = TRUE),
    min   = min(x, na.rm = TRUE),
    Q1 = quantile(x, 0.25, na.rm = TRUE),
    median = median(x, na.rm = TRUE),
    Q3 = quantile(x, 0.75, na.rm = TRUE),
    max   = max(x, na.rm = TRUE)
  )
}
stats <- sapply(df1, summary_stats)
print(stats)
}
```

	height	width	ratio
count	2001.00000	2001.00000	2001.000000
mean	57.08346	110.15292	2.906046
std	41.86233	61.04871	2.418271
min	1.00000	1.00000	0.208300
Q1.25%	24.00000	71.00000	1.000000
median	43.00000	100.00000	1.754900
Q3.75%	82.00000	140.00000	4.000000
max	172.00000	337.00000	11.363600

Figure 9: Thống kê mô tả dữ liệu

- Số lượng mẫu (count): Dữ liệu có 2001 quan sát cho cả ba biến: height, width, và ratio.
- Giá trị trung bình (mean): Chiều cao (height) trung bình là 57.08, chiều rộng (width) trung bình là 110.15, tỉ số (ratio) trung bình là 2.91.
- Độ lệch chuẩn (std): Độ lệch chuẩn của height là 41.86, cho thấy dữ liệu phân tán khá lớn quanh giá trị trung bình. width có độ lệch chuẩn 61.05, cũng thể hiện sự phân tán cao. ratio có độ lệch chuẩn 2.42, mức độ phân tán vừa phải.
- Giá trị nhỏ nhất (min) và lớn nhất (max):
  - height: từ 1 đến 172.
  - width: từ 1 đến 337.
  - ratio: từ 0.21 đến 11.36.

  Điều này cho thấy dữ liệu có sự chênh lệch lớn giữa các giá trị nhỏ nhất và lớn nhất.
- Các phân vị (Q1, median, Q3):
  - Với height, 25% dữ liệu có height nhỏ hơn hoặc bằng 24, 50% nhỏ hơn hoặc bằng 43, và 75% nhỏ hơn hoặc bằng 82.
  - Với width, các giá trị tương ứng là 71 (Q1), 100 (median), và 140 (Q3).
  - Với ratio, 25% nhỏ hơn 1.00, 50% nhỏ hơn 1.75, và 75% nhỏ hơn 4.00.
- Dữ liệu có phân phối lệch phải (right-skewed) do các giá trị trung bình lớn hơn trung vị.
- Độ lệch chuẩn lớn cho thấy dữ liệu có sự phân tán mạnh.

3.2.2 Vẽ đồ thị histogram

Ta tiến hành vẽ đồ thị histogram của biến height, width, ratio

```
hist(df1$height,
     main = "Histogram of Height",
     xlab = "Height",
     col = "pink",
     border = "white")

hist(df1$width,
     main = "Histogram of Width",
     xlab = "Width",
     col = "purple",
     border = "white")

hist(df1$ratio,
     main = "Histogram of Ratio",
     xlab = "Ratio",
     col = "lightgreen",
     border = "white")
```

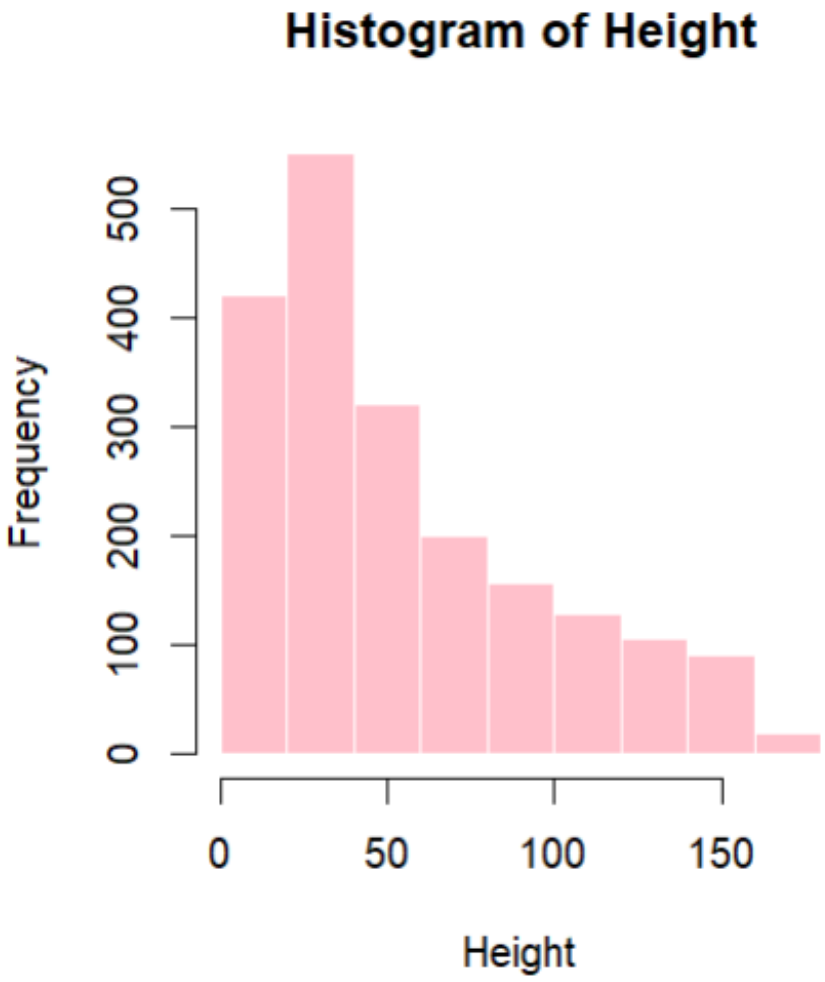


Figure 10: Đồ thị histogram theo biến height

- Phân phối lệch phải: Dữ liệu chiều cao (Height) có phân phối lệch phải rõ rệt, nghĩa là phần lớn các giá trị tập trung ở mức thấp, trong khi số lượng giá trị lớn giảm dần và kéo dài về phía bên phải.
- Tần suất cao ở giá trị nhỏ: Số lượng mẫu có Height nhỏ (dưới 50) chiếm đa số trong tập dữ liệu.
- Có thể có ngoại lai: Một số giá trị Height rất lớn xuất hiện với tần suất thấp, đây có thể là các giá trị ngoại lai.
- Không phân phối chuẩn: Dữ liệu không tuân theo phân phối chuẩn mà nghiêng về phía các giá trị nhỏ.

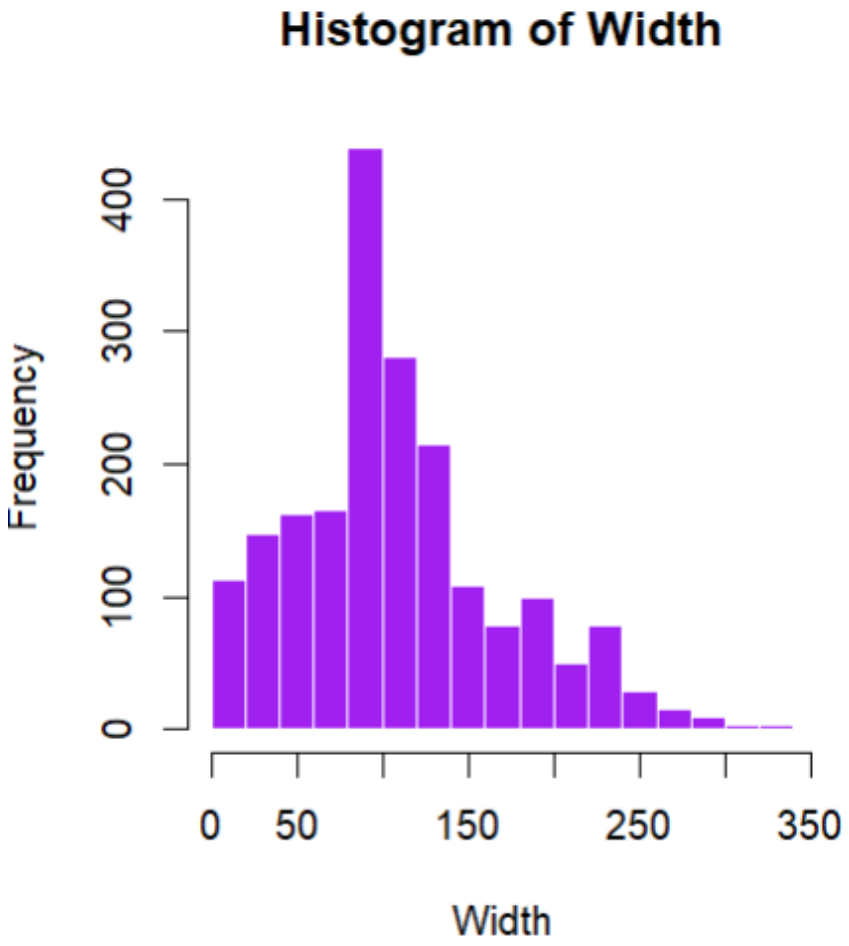


Figure 11: Đồ thị histogram theo biến width

- Phân phối lệch phải: Dữ liệu chiều rộng (Width) có phân phối lệch phải, tức là phần lớn các giá trị Width tập trung ở mức thấp và trung bình, trong khi số lượng giá trị Width lớn giảm dần về phía bên phải.
- Tần suất cao ở khoảng 70-100: Số lượng mẫu có Width trong khoảng 70-100 là cao nhất, thể hiện qua cột cao nhất trên biểu đồ.
- Độ phân tán lớn: Dữ liệu Width trải dài từ giá trị nhỏ (gần 0) đến hơn 300, cho thấy độ phân tán lớn.
- Có thể có ngoại lai: Một số giá trị Width rất lớn xuất hiện với tần suất thấp ở phía bên phải, đây có thể là các giá trị ngoại lai.
- Không phân phối chuẩn: Dữ liệu không tuân theo phân phối chuẩn mà nghiêng về phía các giá trị nhỏ và trung bình.

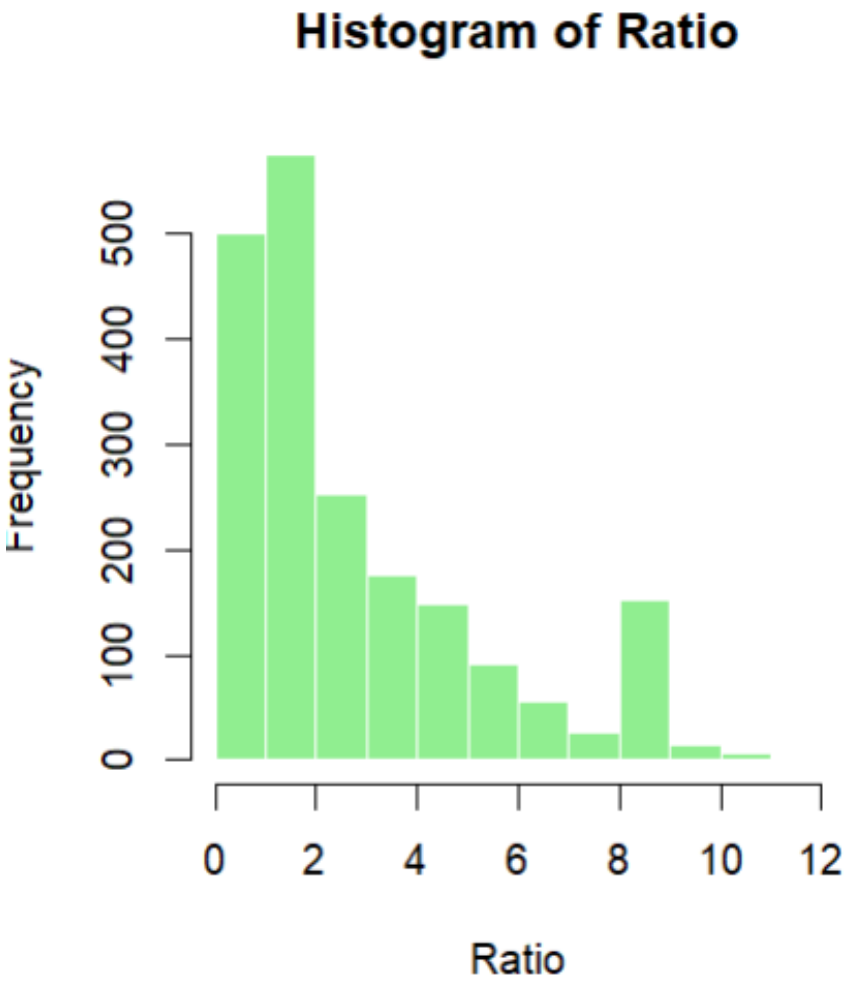


Figure 12: Đồ thị histogram theo biến ratio

- Phân phối lệch phải: Dữ liệu của biến Ratio có phân phối lệch phải rất rõ rệt. Phần lớn các giá trị Ratio tập trung ở mức thấp (dưới 2), tần suất giảm dần khi Ratio tăng lên.
- Tần suất cao ở giá trị nhỏ: Số lượng mẫu có Ratio nhỏ (dưới 2) chiếm đa số trong tập dữ liệu, thể hiện qua hai cột đầu tiên rất cao.
- Đuôi dài về phía phải: Có một số giá trị Ratio lớn (trên 6, thậm chí trên 10) xuất hiện với tần suất thấp, tạo thành một "đuôi dài" về phía bên phải. Đây có thể là các giá trị ngoại lai.
- Không phân phối chuẩn: Dữ liệu không tuân theo phân phối chuẩn mà nghiêng mạnh về phía các giá trị nhỏ.

3.2.3 Vẽ đồ thị boxplot

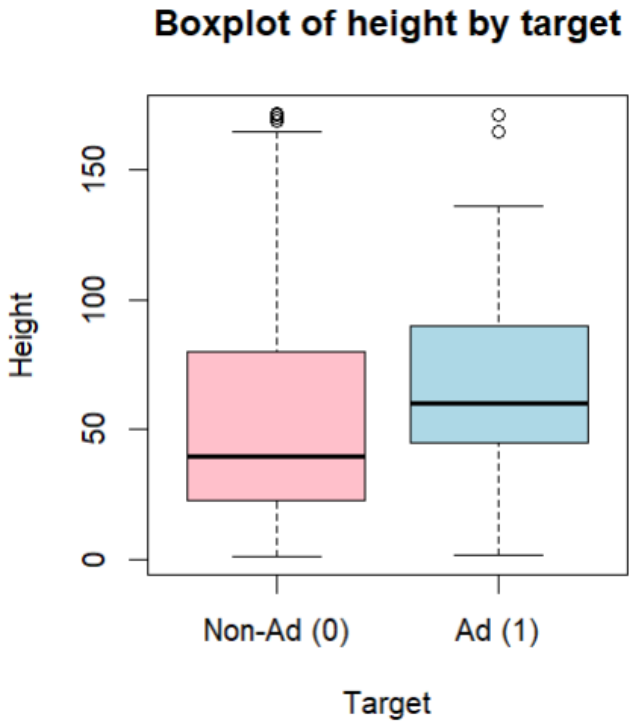


Figure 13: Box plot của height theo target

Sự khác biệt giữa hai nhóm:

- Nhóm "Ad (1)" có giá trị height trung bình và trung vị cao hơn so với nhóm "Non-Ad (0)". Điều này cho thấy các quảng cáo (Ad) thường có chiều cao lớn hơn so với các đối tượng không phải quảng cáo (Non-Ad).
- Độ phân tán: Cả hai nhóm đều có độ phân tán lớn, thể hiện qua chiều dài của hộp và râu (whiskers). Tuy nhiên, nhóm "Ad (1)" có xu hướng phân tán về phía các giá trị height lớn hơn.
- Ngoại lai (outliers): Cả hai nhóm đều xuất hiện các giá trị ngoại lai (các vòng tròn nhỏ phía trên), đặc biệt là ở phía giá trị height lớn.
- Vị trí trung vị: Đường trung vị (đường đen đậm trong hộp) của nhóm "Ad (1)" cao hơn rõ rệt so với nhóm "Non-Ad (0)", cho thấy sự khác biệt về chiều cao giữa hai nhóm.

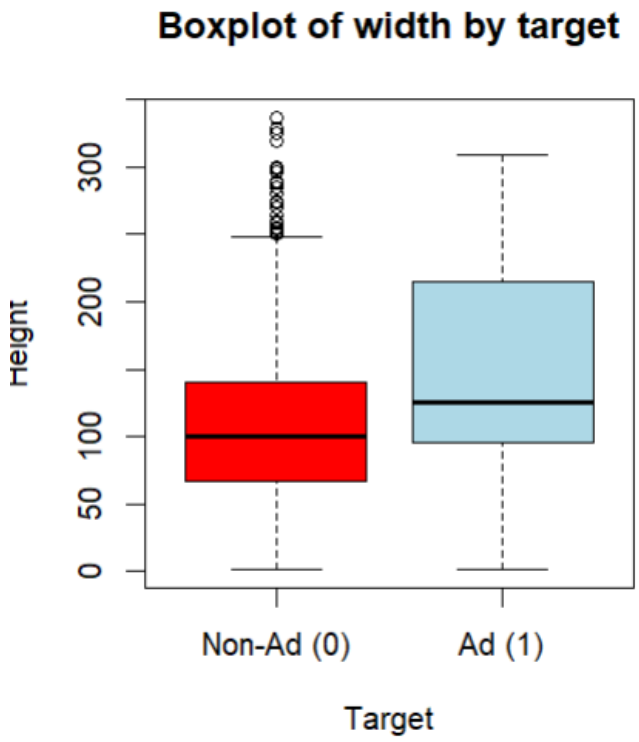


Figure 14: Box plot của width theo target

Sự khác biệt giữa hai nhóm:

- Nhóm quảng cáo (Ad) có giá trị width trung vị và giá trị lớn hơn rõ rệt so với nhóm không quảng cáo (Non-Ad). Điều này cho thấy các quảng cáo thường có chiều rộng lớn hơn các đối tượng không phải quảng cáo.
- Độ phân tán: Độ phân tán của width ở nhóm Ad cũng lớn hơn, thể hiện qua chiều dài của hộp và râu. Nhóm Ad có nhiều giá trị width cao, trong khi nhóm Non-Ad chủ yếu tập trung ở các giá trị width thấp và trung bình.
- Ngoại lai: Nhóm Non-Ad xuất hiện nhiều giá trị ngoại lai (outliers) ở phía width lớn, trong khi nhóm Ad có ít ngoại lai hơn. Điều này cho thấy trong nhóm không quảng cáo vẫn tồn tại một số trường hợp có width rất lớn, nhưng đây là các trường hợp hiếm gặp.
- Vị trí trung vị: Đường trung vị của nhóm Ad cao hơn rõ rệt so với nhóm Non-Ad, cho thấy sự khác biệt về chiều rộng giữa hai nhóm là khá rõ ràng.

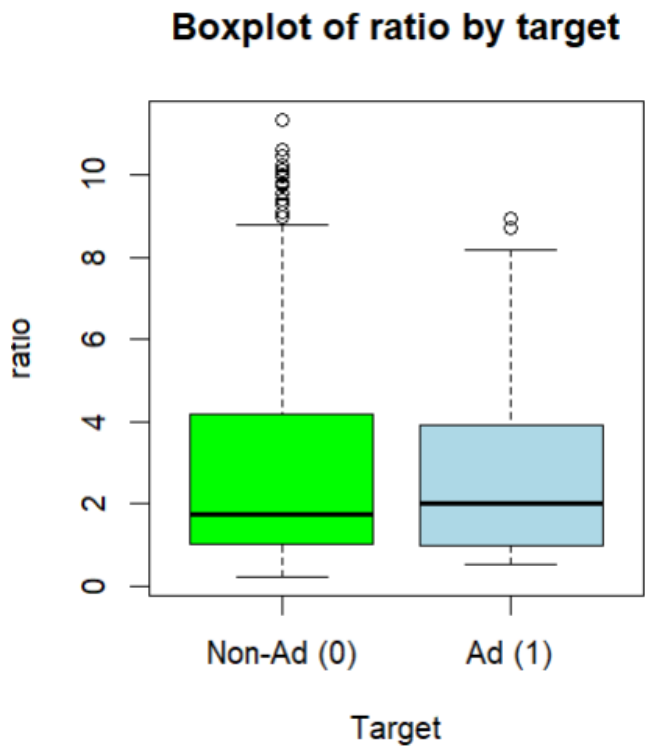


Figure 15: Box plot của ratio theo target

Sự khác biệt giữa hai nhóm:

- Sự khác biệt về trung vị: Trung vị của ratio ở nhóm Non-Ad (0) cao hơn so với nhóm Ad (1). Điều này cho thấy các đối tượng không phải quảng cáo thường có tỷ lệ chiều rộng/chiều cao lớn hơn so với quảng cáo.
- Độ phân tán và phạm vi giá trị: Nhóm Non-Ad có độ phân tán ratio lớn hơn, thể hiện qua chiều dài của hộp và râu. Khoảng giá trị ratio của nhóm này cũng rộng hơn, với nhiều trường hợp có ratio rất cao. Trong khi đó, nhóm Ad có phân bố ratio tập trung hơn ở các giá trị thấp và trung bình.
- Ngoại lai: Nhóm Non-Ad xuất hiện nhiều giá trị ngoại lai (outliers) ở phía ratio lớn, cho thấy có một số trường hợp đặc biệt với tỷ lệ rất cao. Nhóm Ad cũng có một số ngoại lai nhưng số lượng và giá trị nhỏ hơn.
- Ý nghĩa phân biệt: Sự khác biệt về phân bố ratio giữa hai nhóm là khá rõ rệt, tuy nhiên mức độ tách biệt không mạnh như ở các biến height hoặc width. Dù vậy, ratio vẫn là một đặc trưng hữu ích để hỗ trợ phân biệt giữa quảng cáo và không quảng cáo, đặc biệt khi kết hợp với các biến khác.

3.3 Thống kê suy diễn

3.3.1 Hồi quy Logistic

3.3.1.a Mục tiêu của mô hình

Ta sẽ xây dựng mô hình với mục đích đánh giá mức độ ảnh hưởng của các biến độc lập (height, width) lên biến phụ thuộc là biến target. Từ đó ta có thể dự báo xem đó có phải là quảng cáo hay không với các thông số cụ thể.

3.3.1.b Thực hiện mô hình

Ta sẽ xây dựng mô hình hồi quy tuyến tính đa tham số (MH1) bao gồm:

- Biến phụ thuộc: target.
- Biến dự báo (biến độc lập): height, width.

Chúng ta tiến hành đặt hạt số ngẫu nhiên để đảm bảo kết quả tái lập được. Sau đó, chuyển cột "target" thành kiểu phân loại (factor) để xử lý đúng khi mô hình phân tích dữ liệu.

```
set.seed(123)
add$target = as.factor(add$target)
```

Sau đó chúng ta chia tệp dữ liệu thành hai phần: train(70%) và test(30%)

```
sample_id <- sample(1:nrow(add),  size = 0.7 * nrow(add))
 train <- add[sample_id, ]
 test <- add[-sample_id, ]
```

Ta sử dụng lệnh sau để xây dựng mô hình hồi quy (dựa trên các biến phụ thuộc và biến độc lập nêu trên) và ước lượng các hệ số có trong mô hình và hiển thị kết quả:

```
 model_lg <- glm(target ~ height + width, data = train, family = binomial)
 summary(model_lg)
```

```
Call:
glm(formula = target ~ height + width, family = binomial, data = train)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.525593   0.239460 -14.723  < 2e-16 ***
height       0.006079   0.002316   2.624 0.008678 **
width        0.005884   0.001619   3.635 0.000278 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 795.12  on 1399  degrees of freedom
Residual deviance: 763.38  on 1397  degrees of freedom
AIC: 769.38

Number of Fisher Scoring iterations: 5
```

Figure 16: Kết quả của mô hình

Trong đó:

- Residuals: Sai số hồi quy, thặng dư, phần dư ϵi (sự chênh lệch giữa y thực nghiệm và y dự báo).
- Esitmate: Là giá trị ước lượng của các hệ số hồi quy (hệ số chặn và các biến độc lập). Ví dụ: Hệ số width là 0.0003003 nghĩa là khi width tăng 1 đơn vị, target tăng trung bình 0.0003003 đơn vị (giữ các biến khác không đổi).
- Std.Error: Sai số chuẩn của ước lượng hệ số, cho biết mức độ không chắc chắn của ước lượng.
- z-value: Giá trị kiểm định z (Estimate / Std. Error). Giá trị z càng lớn (dương hoặc âm), hệ số càng có ý nghĩa thống kê.
- Pr($> \|z\|$) là giá trị p-value, cho biết xác suất để hệ số nhận được giá trị như vậy (hoặc lớn hơn) nếu hệ số thực sự bằng 0. Nếu p-value nhỏ hơn 0.05, hệ số được coi là có ý nghĩa thống kê.. Tiếp theo ta cần kiểm định các hệ số hồi quy:
  - Giả thuyết H0: Hệ số hồi quy không có ý nghĩa thống kê ($\hat{\beta}_i = 0$).
  - Giả thuyết H1: Hệ số hồi quy có ý nghĩa thống kê ($\hat{\beta}_i \neq 0$)
- Null deviance: Độ lệch chuẩn của mô hình chỉ có hệ số chặn.
- Residual deviance: Độ lệch chuẩn của mô hình đã bao gồm các biến độc lập.
- AIC: Chỉ số thông tin Akaike, dùng để so sánh mức độ phù hợp của các mô hình (thấp hơn là tốt hơn).
- Number of Fisher Scoring iterations: Số vòng lặp Fisher Scoring đã thực hiện để hội tụ mô hình.

Mô hình hồi quy logistic này sử dụng hai biến độc lập là height và width để dự báo xác suất xảy ra biến mục tiêu. Kết quả cho thấy cả hai biến đều có ý nghĩa thống kê, với giá trị p nhỏ hơn 0.05, chứng tỏ chúng thực sự ảnh hưởng đến xác suất của biến mục tiêu.

Cụ thể, hệ số của hai biến đều dương, nghĩa là khi chiều cao hoặc chiều rộng tăng lên thì xác suất xảy ra biến mục tiêu cũng có xu hướng tăng. Mô hình đã cải thiện so với mô hình chỉ có hệ số chặn, thể hiện qua việc độ lệch chuẩn giảm từ 795.12 xuống còn 763.38 sau khi thêm các biến độc lập. Giá trị AIC là 769.38, cho thấy mô hình có mức độ phù hợp nhất định. Ngoài ra, mô hình hội tụ nhanh chỉ sau 5 vòng lặp nên không gặp vấn đề về tối ưu hóa.

Chúng ta tiến hành đánh giá mô hình với tệp test.

```
prob <- predict(model_lg, newdata = test, type = "response")
pred <- ifelse(prob >= 0.5, 1, 0)
cat("Accuracy:", mean(pred == as.numeric(as.character(test$target))), "\n")
```

```
Accuracy: 0.9168053
```

Figure 17: Kết quả accuracy của mô hình

```
roc_obj <- roc(test$target, prob)
cat("AUC:", auc(roc_obj), "\n")
plot(roc_obj, main = "ROC Curve")
```

```
AUC: 0.7392377
```

Figure 18: Kết quả AUC

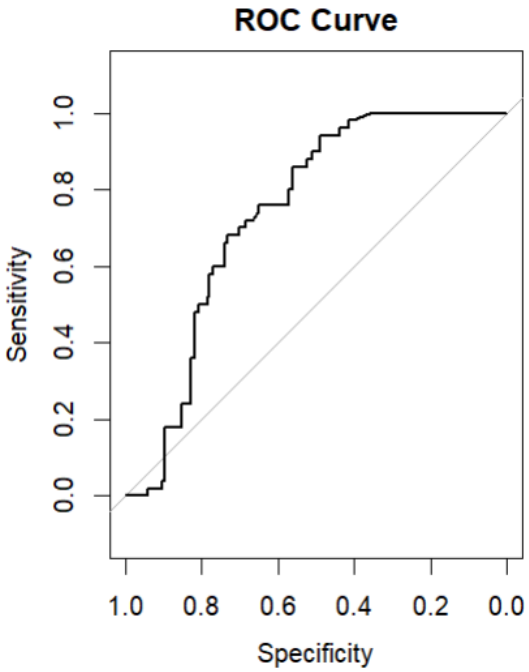


Figure 19: Kết quả ROC

- Dưới đây là nhận xét về kết quả đánh giá mô hình: Accuracy = 0.9168
- Mô hình dự đoán đúng khoảng 91.7% các trường hợp trên tập kiểm tra. Đây là một độ chính xác khá cao, cho thấy mô hình phân biệt tốt giữa hai nhóm target trong dữ liệu.
- Giá trị AUC (Area Under the Curve) là 0.7392, cho thấy mô hình có khả năng phân biệt giữa hai lớp tốt hơn ngẫu nhiên (AUC = 0.5), nhưng chưa đạt mức xuất sắc (AUC > 0.8). Điều này nghĩa là mô hình có hiệu quả phân loại khá, nhưng vẫn còn dư địa để cải thiện.
- Đường cong ROC nằm phía trên đường chéo, thể hiện mô hình có khả năng phân biệt hai lớp. Tuy nhiên, đường cong chưa sát với góc trên bên trái, cho thấy mô hình chưa hoàn hảo.

3.3.2 Decision Tree

Mục tiêu của việc áp dụng mô hình Decision Tree trong nghiên cứu này là:

- Xây dựng mô hình dự đoán khả năng một mẫu dữ liệu thuộc nhóm ad hay non-ad dựa trên các đặc trưng hình ảnh (height, width, ratio)
- Giải thích các yếu tố ảnh hưởng đến quyết định phân loại thông qua cấu trúc cây trực quan

3.3.2.1 Thiết lập môi trường và chia dữ liệu

- Sử dụng gói `rpart` đvà `rpart.plot` trong R. Với `rpart` để xây dựng decision tree model và được trực quan hóa bằng hàm `rpart.plot()`.
- Dung `set.seed(123)` để đảm bảo kết quả chia dữ liệu ngẫu nhiên sẽ giống nhau mỗi lần chạy code
- Sau khi tiền xử lý thì chia dữ liệu thành tập train (70%) và test (30%).

```
library(rpart)
library(rpart.plot)

set.seed(123)
sample_id <- sample(1:nrow(df1), size = 0.7 * nrow(df1))
train <- df1[sample_id, ]
test <- df1[-sample_id, ]
```

3.3.2.2 Training mô hình

- Phương pháp CART (Classification and Regression Tree)
- Kiểm soát độ phức tạp bằng tham số cp (complexity parameter)

```
# Basic model
tree_model <- rpart(target ~ height + width + ratio ,
                    data = train,
                    method = "class",
                    cp = 0.01)
```

Lưu ý: Nếu như mô hình phức tạp hơn thì sử dụng các kĩ thuật kiểm soát như về độ sâu (maxdepth), số nút tối thiểu (minsplit),...

```
Model with tighter control
control <- rpart.control(minsplit = 5,
                         cp = 0.01,
                         maxdepth = 3)
tree_model <- rpart(target ~ height + width + ratio,
                    data = train,
                    method = "class",
                    control = control)
```

3.3.2.3 Đánh giá mô hình

- Dự đoán trên tập test

```
# Predict
# Get predictions from decision tree model
```

```r
pred_tree_class <- predict(tree_model, newdata = test, type = "class")
# Probability of class 1
pred_tree_prob <- predict(tree_model, newdata = test)[,2]

result_df <- data.frame(TrueLabel = test$target,
                        PredictedClass = pred_tree_class,
                        PredictedProb = pred_tree_prob)
print(result_df)
```

pred_tree_class là một vector chứa các giá trị dự đoán nhãn lớp cho tập test Ví dụ giá trị của vector:

Figure 20: Giá trị của vector
vector pred_tree_class

pred_tree_prob là một vector chứa các giá trị xác suất tương ứng với khả năng mỗi mẫu thuộc lớp 1.

Figure 21: Giá trị của vector
pred_tree_prob

	TrueLabel	PredictedClass	PredictedProb
8	1	1	0.92307692
15	1	0	0.03800475
29	1	1	0.92307692
41	1	0	0.03800475
42	1	0	0.03800475
43	1	0	0.03800475
49	1	0	0.12962963
56	1	0	0.12962963
77	1	1	0.92307692
85	1	0	0.12962963
86	1	0	0.12962963
95	1	1	0.92307692
98	1	0	0.03800475
101	1	0	0.03800475
105	1	0	0.03800475
114	1	0	0.12962963
117	1	0	0.12962963
135	1	0	0.12962963
139	1	0	0.12962963
142	1	0	0.12962963
161	1	0	0.03800475
189	1	0	0.12962963
204	1	0	0.12962963

Figure 22: Tương quan giữa label và xác suất dự đoán trên tập test

- Đánh giá qua confusion matrix và các metrics

```r
# Confusion matrix
confusion_matrix <- table(Predicted = pred_tree_class, Actual = test$target)
print(confusion_matrix)
```

Table 1: Confusion Matrix

Predicted / Actual	**Non-Ad (0)**	**Ad (1)**
Non-Ad (0)	550	43
Ad (1)	1	7

- Acurancy

```r
accuracy_tree <- mean(pred_tree_class == test$target)
cat("Accuracy of decision tree:", round(accuracy_tree, 4), "\n")
```

```
Accuracy of decision tree: 0.9268
```

Figure 23: Accuracy của decision tree

Kết quả: mô hình dự đoán tổng thể với độ chính xác với 92,68% cho thấy mô hình hoạt động cực kỳ hiệu quả trong việc phân loại quảng cáo.

- ROC và AUC

```r
roc_tree <- roc(test$target, pred_tree_prob)
auc_tree <- auc(roc_tree)

plot(roc_tree, col = "darkgreen", lwd = 2,
     main = "ROC Curve - Decision Tree")
cat("AUC:", auc_tree)
```

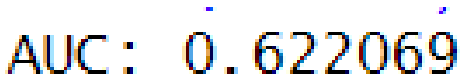


Figure 24: AUC của decision tree

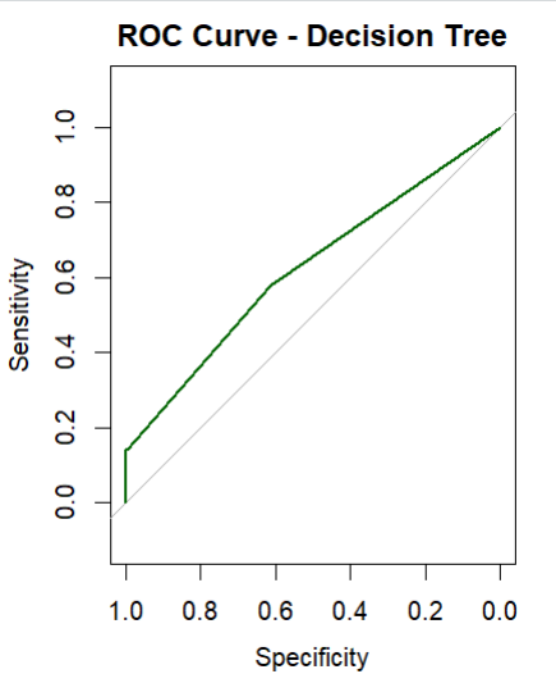


Figure 25: ROC của decision tree

Kết quả: đường cong khá góc cạnh với một góc nhọn, điều này điển hình cho các mô hình cây quyết định thực hiện các quyết định phân loại rời rạc thay vì cung cấp ước tính xác suất mượt mà. Mặt khác giá trị AUC chỉ đạt khoảng 0.64 cho thấy hiệu suất dự đoán khá thấp.

3.3.2.4 Kết luận và nhận xét

- **Ưu điểm:**
  - Khả năng phân loại tootsL với độ chính xác là 92.68%.
  - Khả năng xử lý dữ liệu phức tạp: Decision Tree đã xử lý tốt các đặc trưng về kích thước (height, width, ratio), có thể do khả năng phát hiện mối quan hệ phi tuyến giữa các biến này.

- **Nhược điểm:** Mặc dù mô hình cây quyết định đạt accuracy khá cao (92.68%), nhưng giá trị AUC chỉ là 0.622. Điều này cho thấy mô hình dự đoán đúng nhiều trường hợp, nhưng khả năng phân biệt giữa hai lớp (0 và 1) chưa thực sự tốt.

Kết luận: cần điều chỉnh ngưỡng phân loại áp dụng kỹ thuật sampling để cân bằng dữ liệu hoặc thử nghiệm với tham số complexity parameter (cp) khác. Đánh giá lại khả năng overfitting bằng áp dụng validation set hoặc cross-validation. Dùng kĩ thuật pruning để giảm nguy cơ overftting.

3.3.3 Random Forest

3.3.3.a Mục tiêu mô hình

Ta sẽ xây dựng mô hình với mục đích đánh giá mức độ ảnh hưởng của các biến độc lập (height, width) lên biến phụ thuộc là biến target. Từ đó ta có thể dự báo xem đó có phải là quảng cáo hay không với các thông số cụ thể.

3.3.3.b Thực hiện mô hình

Tiếp theo chúng ta sẽ tạo ra model và huấn luyện model với training dataset.

```
rf_model <- randomForest(target ~ height + width, data = train ,  ntree = 500, mtry
  = 2, importance = TRUE)
```

In kết quả của mô hình để có cái nhìn tổng quan.

```
print(rf_model)
```

```
        OOB estimate of  error rate: 4.79%
Confusion matrix:
     0  1 class.error
0 1254 31  0.02412451
1   36 79  0.31304348
```

Figure 26: Hình confusion matrix

Mặc dù OOB error tổng thể (ước lượng tỷ lệ mẫu bị phân loại sai khi sử dụng các cây không chứa mẫu đó trong quá trình huấn luyện) chỉ khoảng 4.79%. Tỷ lệ lỗi này khá thấp, nghĩa là mô hình dự đoán tốt trên dữ liệu chưa từng "thấy", nhưng đây chủ yếu là do hầu hết các mẫu thuộc lớp 0 (lớp chiếm đa số) được dự đoán rất chính xác.
Ngược lại, mô hình dự đoán rất kém đối với lớp 1 với tỷ lệ lỗi hơn 30%. Điều này cho thấy rằng mô hình gặp khó khăn trong việc nhận diện đúng các trường hợp thuộc lớp 1. Nguyên nhân là do dữ liệu mất cân bằng, tức số lượng mẫu của lớp 1 quá ít so với lớp 0, hoặc các đặc trưng hiện có (height và width) không đủ phân biệt giữa lớp 0 và lớp 1.

Chúng ta sẽ thực hiện dự đoán dựa trên model đã được huấn luyện với training dataset với mục đích xem model này khi được thực hiện với tập dữ liệu mới (test dataset) sẽ dự đoán được đúng bao nhiêu phần trăm. Chúng ta sẽ cho mô hình thực hiện dự đoán với test dataset:

```
pred_class <- predict(rf_model, newdata = test, type = "response")
pred_proba <- predict(rf_model, newdata = test, type = "prob")
```

Tiếp theo, chúng ta sẽ đánh giá mô hình xem sau khi được huấn luyện với training dataset thì với test dataset sẽ dự đoán đúng được bao nhiêu phần trăm.

```
results <- data.frame(
target = test$target,
pred_class = pred_class,
```

```
    prob_0 = pred_proba[,1],
    prob_1 = pred_proba[,2]
    accuracy <- mean(results$pred_class == results$target)
)
```

```
[1] 0.9750416
```

Figure 27: Kết quả accuracy

Như vậy, sau khi mô hình được huấn luyện và sau đó được thực hiện với bộ dữ liệu mới thì độ chính xác của mô hình này là 97.5%. Con số này cho thấy đây là mô hình có thể tin tưởng được.
Chúng ta hiển thị tầm quan trọng của các biến.

```
    importance(rf_model)
```

	0	1	MeanDecreaseAccuracy	MeanDecreaseGini
height	57.22917	111.6402	96.01031	87.20209
width	55.91082	111.1958	93.52556	102.08717

Figure 28: Enter Caption

Trong đó:

- MeanDecreaseAccuracy: Độ giảm accuracy trung bình khi loại bỏ từng biến.
- MeanDecreaseGini: Độ giảm chỉ số Gini khi loại bỏ từng biến (liên quan đến mức độ "tách biệt" các lớp).

Trung bình height của nhóm Non-Ad là 57.23, còn nhóm Ad là 111.64.
Trung bình width của nhóm Non-Ad là 55.91, còn nhóm Ad là 111.20.
Biến height và width đều có giá trị MeanDecreaseAccuracy và MeanDecreaseGini khá cao, cho thấy cả hai đều đóng vai trò quan trọng trong việc phân loại.
So sánh giữa hai biến: height có MeanDecreaseAccuracy (96.01) cao hơn width (93.53), nghĩa là khi loại height ra khỏi mô hình thì độ chính xác giảm nhiều hơn so với khi loại width. Tuy nhiên, width lại có MeanDecreaseGini (102.09) cao hơn height (87.20), cho thấy width đóng góp nhiều hơn vào việc giảm độ hỗn loạn (Gini impurity) trong các cây quyết định.

```
    roc_rf <- roc(as.numeric(as.character(test$target)), results$prob_1)
    auc_rf <- auc(roc_rf)
    cat("AUC (Random Forest):", auc_rf, "\n")
    plot(roc_rf, main = "ROC Curve - Random Forest", col = "blue", lwd = 2)
)
```

```
AUC (Random Forest): 0.9482759
```

Figure 29: AUC của random forest

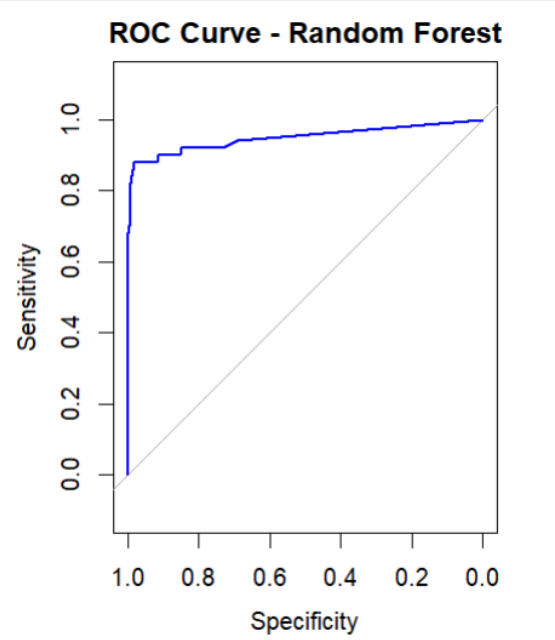


Figure 30: ROC của random forest

Giá trị AUC gần 1 cho thấy mô hình Random Forest có khả năng phân biệt giữa hai lớp (quảng cáo và không quảng cáo) rất tốt. Biểu đồ ROC của mô hình Random Forest cho thấy đường cong nằm sát phía trên bên trái, chứng tỏ mô hình có khả năng phân biệt hai lớp rất tốt. Độ nhạy (Sensitivity) và độ đặc hiệu (Specificity) đều cao ở nhiều ngưỡng khác nhau. Điều này phù hợp với giá trị AUC cao (gần 0.95), cho thấy mô hình Random Forest dự đoán chính xác và ổn định, là lựa chọn rất hiệu quả cho bài toán phân loại này.

3.3.3.c Nhận xét

Random Forest là mô hình phân loại thích hợp để dự đoán xác suất thuộc về một trong các lớp (trong trường hợp này là 0 hoặc 1), dữ liệu không hoàn toàn tuyến tính, cho ra kết quả dự đoán tương đối tốt, qua đó có thể tin tưởng được.

3.3.3.d Ưu điểm

- Độ chính xác cao: Random Forest tổng hợp dự đoán từ nhiều cây quyết định, giúp cải thiện độ chính xác và giảm hiện tượng overfitting so với một cây đơn lẻ.

- Khả năng xử lý phi tuyến: Vì mỗi cây quyết định có biên phân cách phi tuyến (piecewise), mô hình Random Forest có thể học được các mối quan hệ phức tạp giữa các biến và biến mục tiêu, phù hợp với dữ liệu không tuyến tính.
- Phản ánh tầm quan trọng của biến (Feature Importance): Mô hình cung cấp các chỉ số như MeanDecreaseAccuracy và MeanDecreaseGini, giúp đánh giá được mức độ quan trọng của từng biến đầu vào trong việc dự đoán.
- Độ ổn định: Nhờ cơ chế bootstrap và tính ngẫu nhiên trong việc chọn biến phân chia ở mỗi nút, mô hình có xu hướng ổn định và kém nhạy cảm với các trường hợp nhiễu (noise) trong dữ liệu.

3.3.3.e Nhược điểm

- Khó giải thích (giải thích trực quan): So với cây quyết định đơn lẻ, Random Forest là tập hợp của rất nhiều cây, nên mô hình trở nên "black-box" và khó diễn giải các quyết định cụ thể.
- Yêu cầu tính toán cao: Việc huấn luyện một số lượng lớn cây (ntree thường lên đến hàng trăm) đòi hỏi nhiều tài nguyên tính toán và thời gian, đặc biệt khi kích thước dữ liệu lớn.

Tài liệu tham khảo

- Quinlan, J. R. (1986). Induction of Decision Trees. Machine Learning
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning. Springer.
- Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. Cambridge University Press.
- Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). Classification and Regression Trees. CRC Press.
- Wikipedia contributors. (n.d.). Decision tree. Wikipedia. Retrieved May 9, 2025.