



Đồ Án Giữa Kì

Môn học: Học Máy

Nhóm: 07

Hướng dẫn: Thầy Lê Anh Cường





Thành viên nhóm

MSSV	Tên thành viên	Email
522H0148	Phạm Đăng Thanh Trung	522H0148@student.tdtu.edu.vn
521H0064	Đinh Công Hưng	521H0064@student.tdtu.edu.vn

Nhiệm vụ

Câu 1: Giải Bài Toán Bằng Các Phương Pháp Học Máy

Câu 2: Overfitting và Giải Pháp

Câu 3: Feature Selection Using Correlation Analysis

Tập Dữ Liệu



Bank Marketing

Donated on 2/13/2012

The data is related with direct marketing campaigns (phone calls) of a Portuguese banking institution. The classification goal is to predict if the client will subscribe a term deposit (variable y).

Dataset Characteristics	Subject Area	Associated Tasks
Multivariate	Business	Classification
Feature Type	# Instances	# Features
Categorical, Integer	45211	16

Dataset Information

Additional Information
The data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed. ...

[SHOW MORE ▾](#)

Has Missing Values?
No

Các thuật toán sử dụng:

Classification (Phân Loại)

- K-Nearest Neighbors
- Logistic Regression
- Decision Tree

Regression (Hồi Quy)

- Linear Regression
- Random Forest Regression

Đặc Điểm Dữ Liệu

Thống kê dữ liệu:

	id	age	balance	day	duration	campaign	pdays	previous
count	4521.000000	4521.000000	4521.000000	4521.000000	4521.000000	4521.000000	4521.000000	4521.000000
mean	2260.000000	41.170095	1422.657819	15.915284	263.961292	2.793630	39.766645	0.542579
std	1305.244613	10.576211	3009.638142	8.247667	259.856633	3.109807	100.121124	1.693562
min	0.000000	19.000000	-3313.000000	1.000000	4.000000	1.000000	-1.000000	0.000000
25%	1130.000000	33.000000	69.000000	9.000000	104.000000	1.000000	-1.000000	0.000000
50%	2260.000000	39.000000	444.000000	16.000000	185.000000	2.000000	-1.000000	0.000000
75%	3390.000000	49.000000	1480.000000	21.000000	329.000000	3.000000	-1.000000	0.000000
max	4520.000000	87.000000	71188.000000	31.000000	3025.000000	50.000000	871.000000	25.000000

Kiểm tra kiểu dữ liệu:

```

Kieu du lieu cac cot:
   id          int64
   age         int64
   job        object
   marital    object
   education  object
   default    object
   balance    int64
   housing    object
   loan       object
   contact    object
   day        int64
   month      object
   duration   int64
   campaign   int64
   pdays      int64
   previous   int64
   poutcome   object
   y          object
dtype: object

           id      age     balance      day duration campaign \
y
no    2261.324500  40.998000  1403.211750  15.948750  226.347500  2.862250
yes   2249.831094  42.491363  1571.955854  15.658349  552.742802  2.266795

           pdays previous
y
no    36.006000  0.471250
yes   68.639155  1.090211

```

Đặc Điểm Dữ Liệu

Kiểm tra cột trong DF và thống kê y:

```

Các cột trong DataFrame: Index(['id', 'age', 'job', 'marital', 'education', 'default', 'balance',
   'housing', 'loan', 'contact', 'day', 'month', 'duration', 'campaign',
   'pdays', 'previous', 'poutcome', 'y'],
  dtype='object')

y
no    4000
yes   521
Name: count, dtype: int64

```

Kiểm tra kiểu có tồn tại null không:

```
Kiem tra xem co gia tri null trong moi cot hay khong:  
id           False  
age          False  
job          False  
marital      False  
education    False  
default      False  
balance      False  
housing      False  
loan          False  
contact      False  
day           False  
month         False  
duration     False  
campaign     False  
pdays        False  
previous     False  
poutcome     False  
y             False  
dtype: bool
```

Đặc Điểm Dữ Liệu

Kiểm tra xem y có tồn tại giá trị khác không:

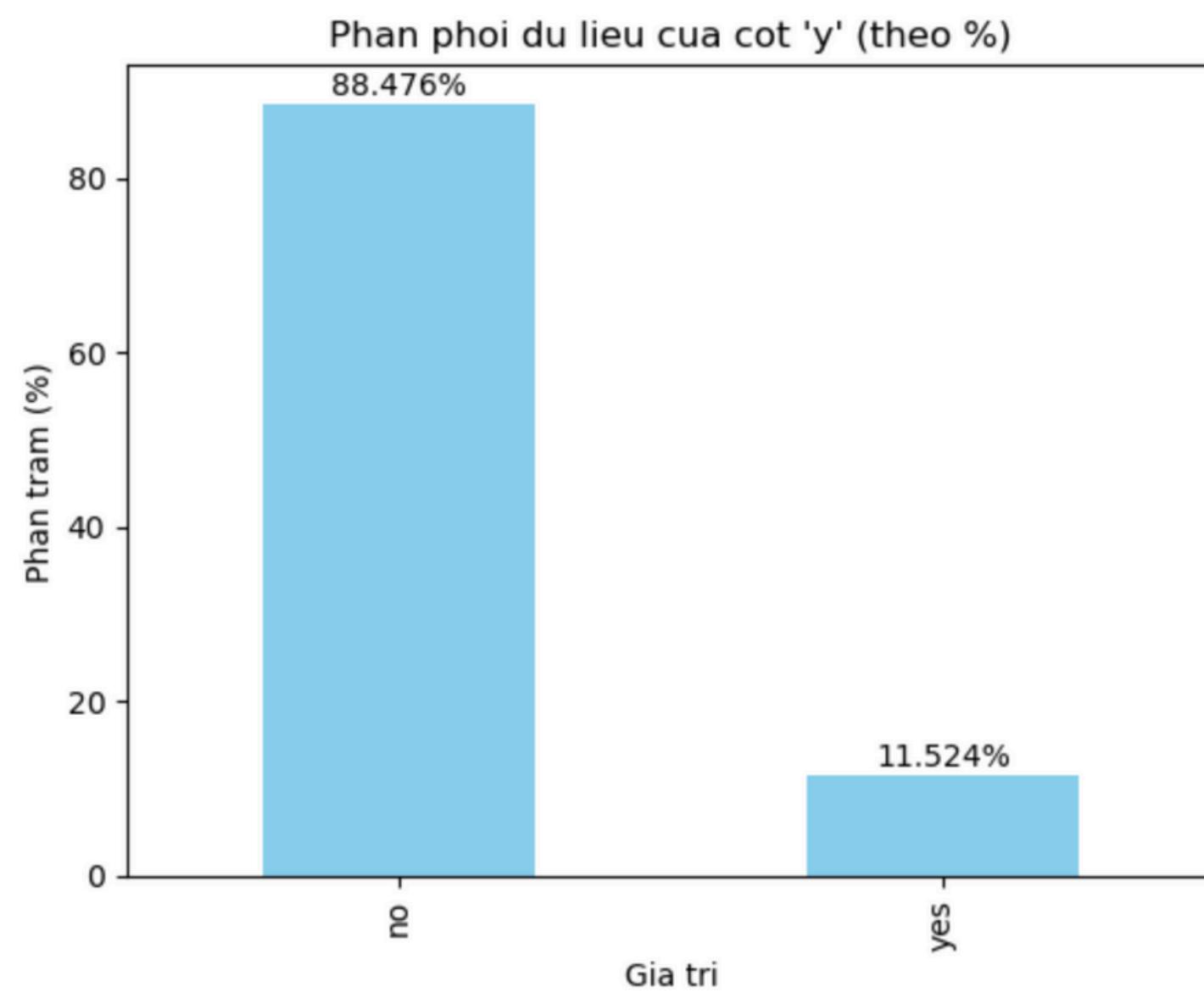
```
Cac gia tri unique trong cot 'y': ['no' 'yes']
```

Kiểm tra xem trong cột age có chứa giá trị số âm không:

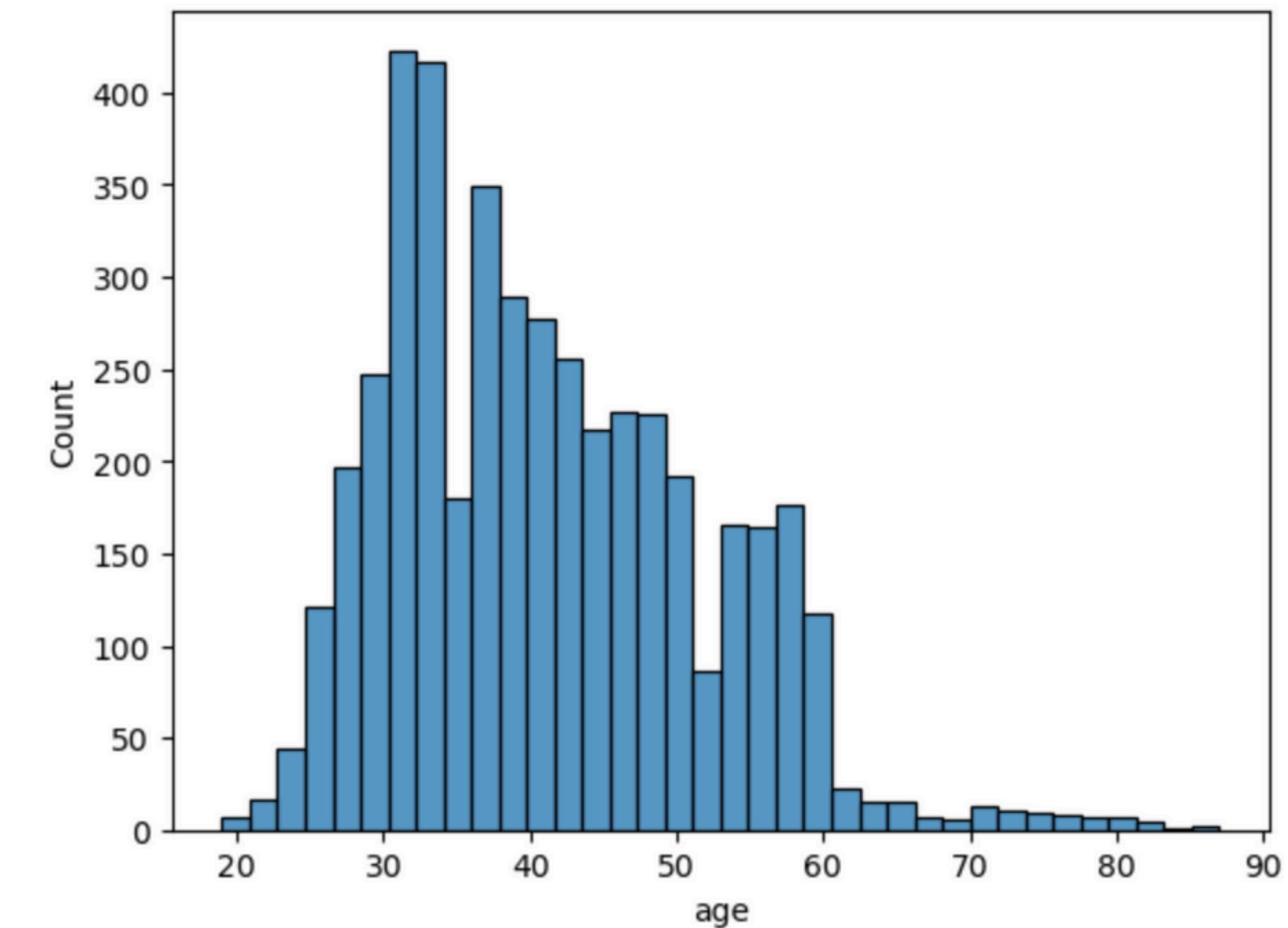
```
Du lieu vo li trong 'age':  
Empty DataFrame  
Columns: [id, age, job, marital, education, default, balance, housing, loan, contact, day, month, duration, campaign, pdays, previous,  
Index: []
```

Đặc Điểm Dữ Liệu

Vài biểu đồ phân phối dữ liệu:



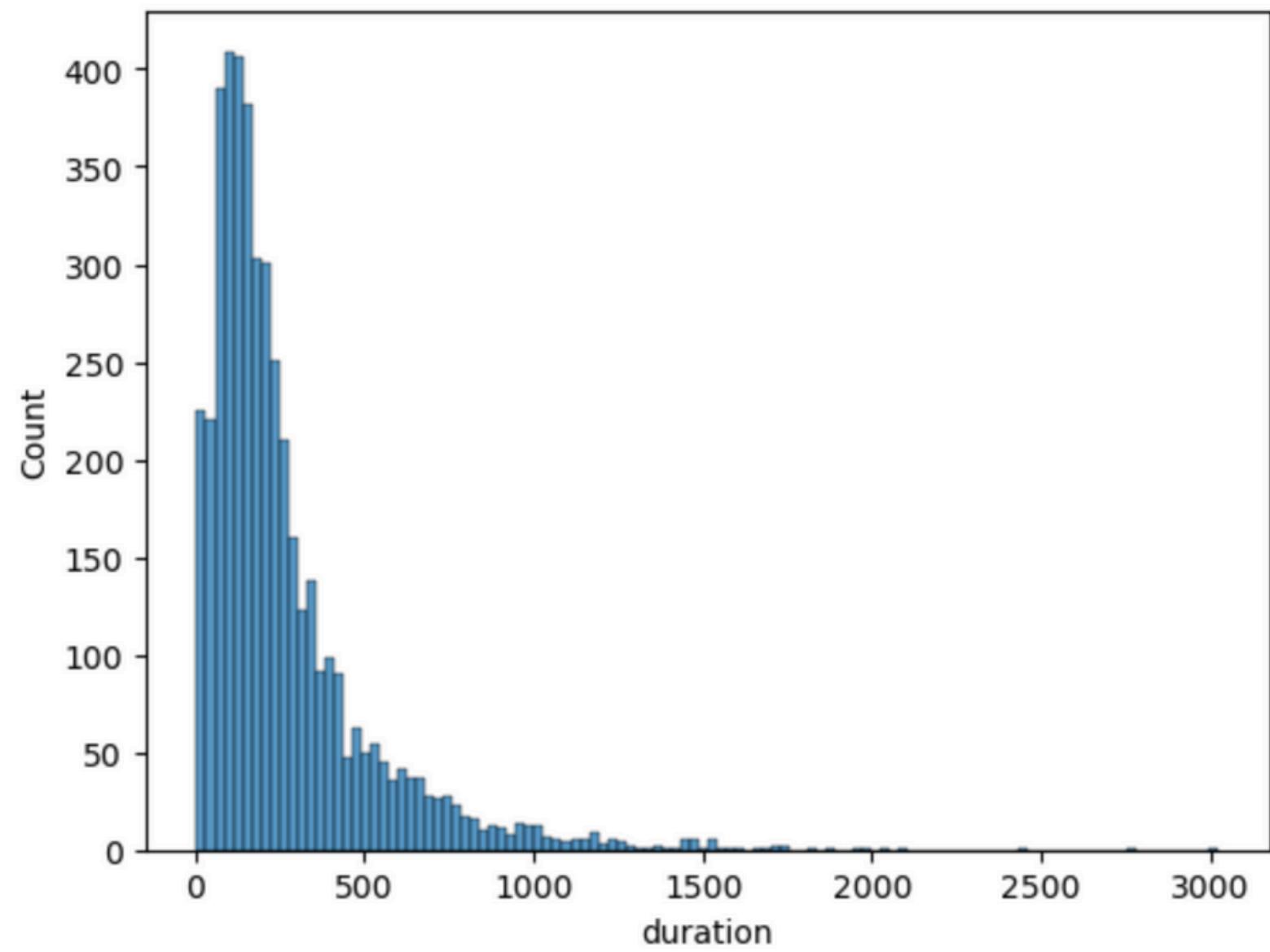
phân phối dữ liệu cột y



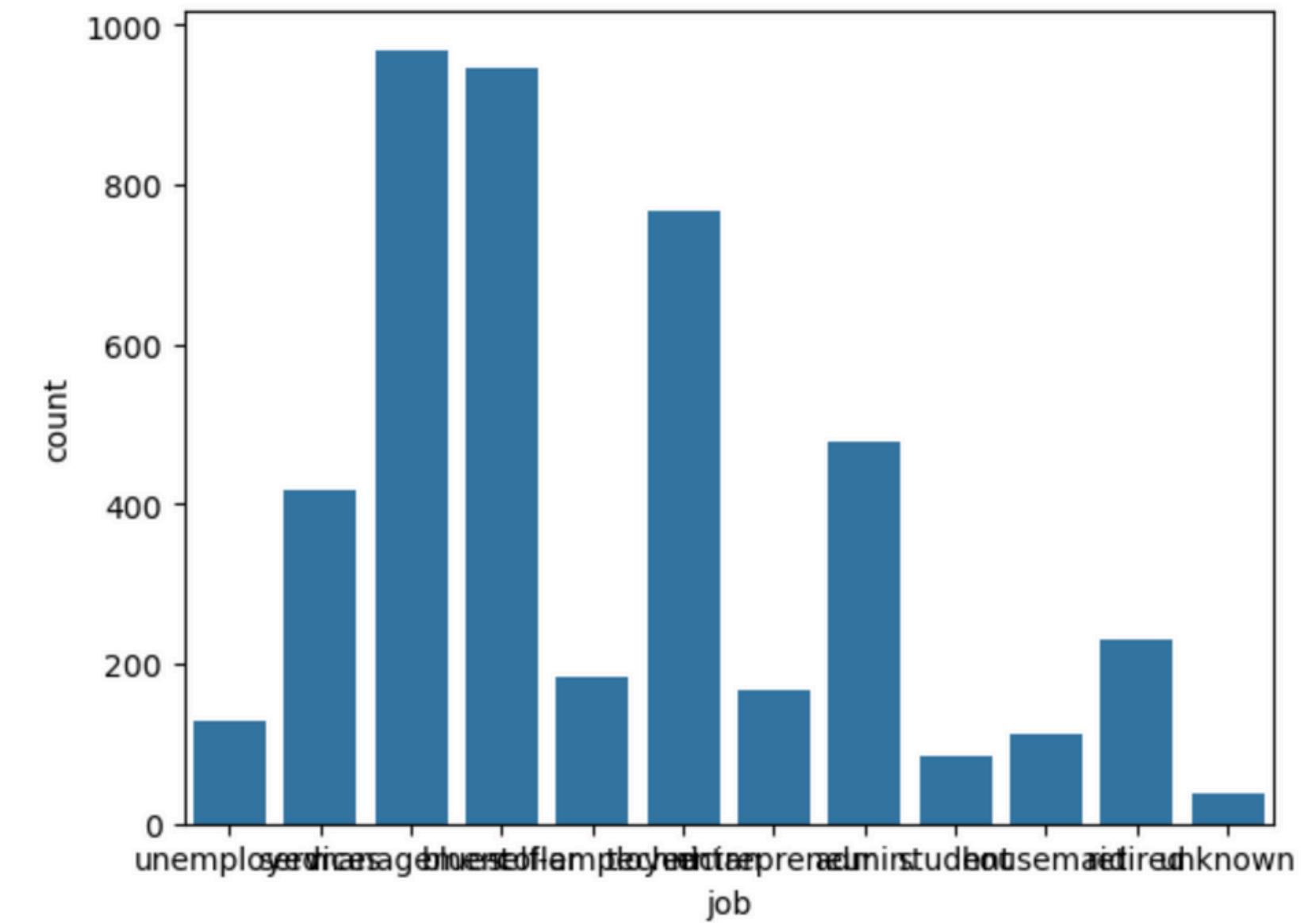
phân phối dữ liệu cột age

Đặc Điểm Dữ Liệu

Vài biểu đồ phân phối dữ liệu:

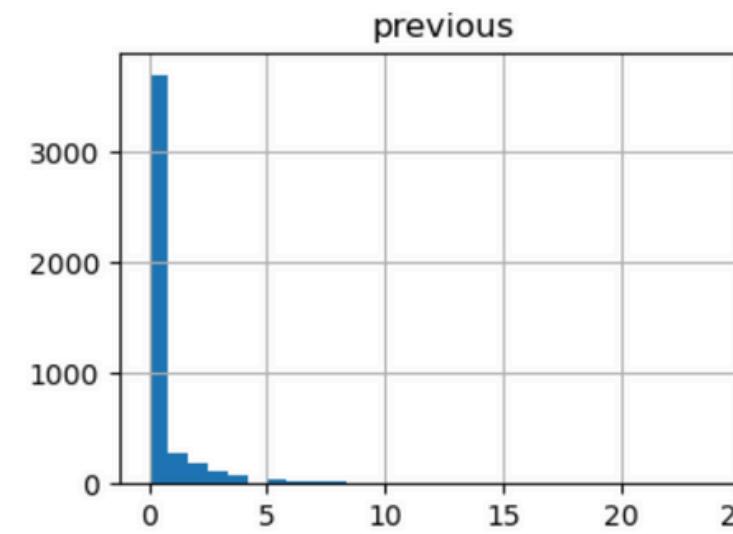
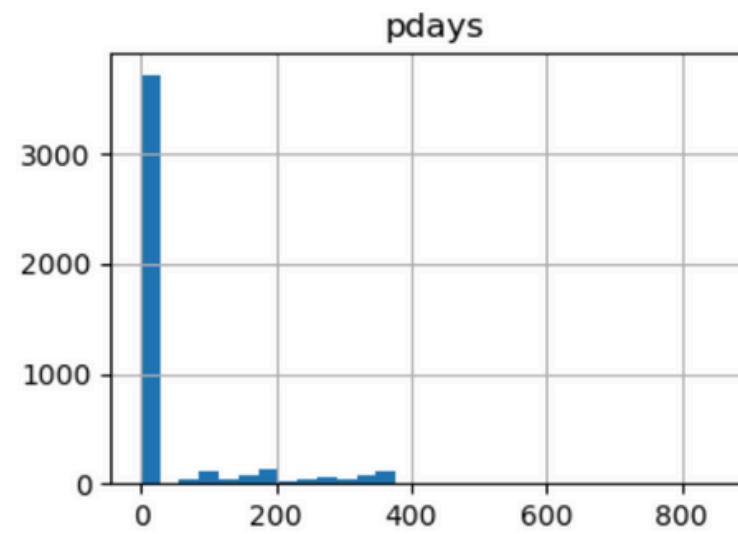
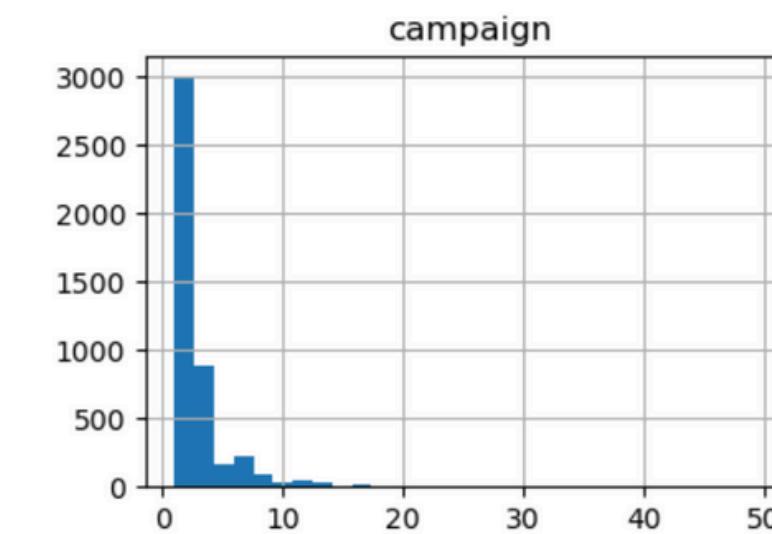
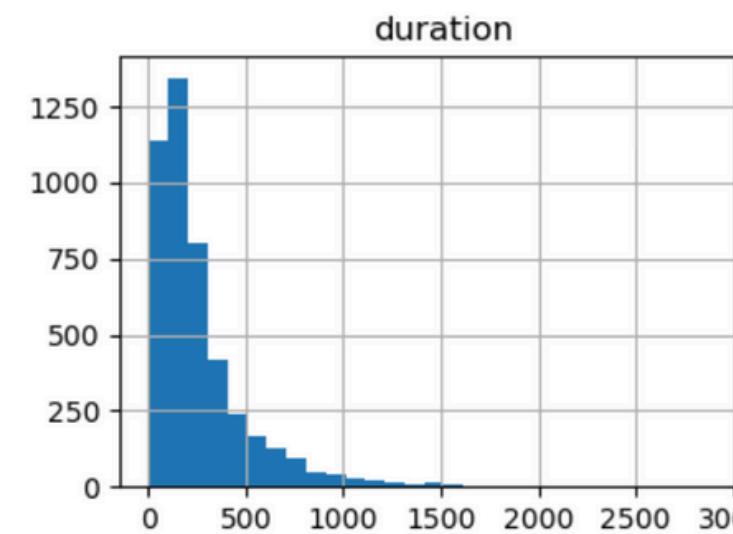
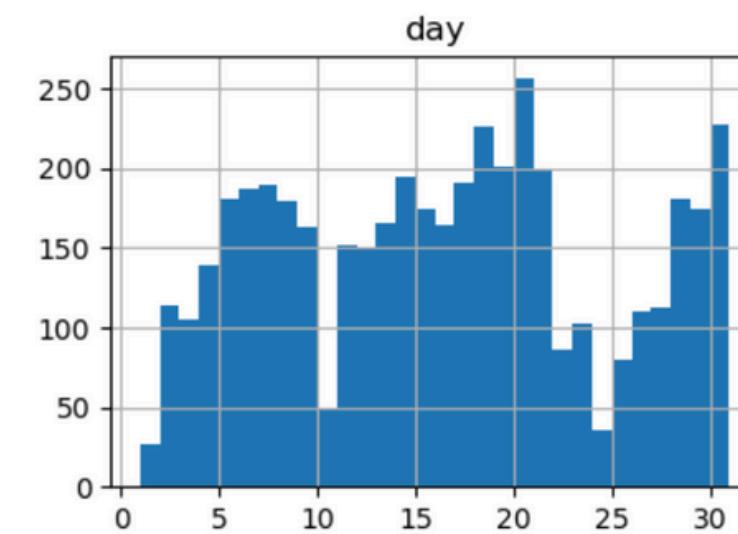
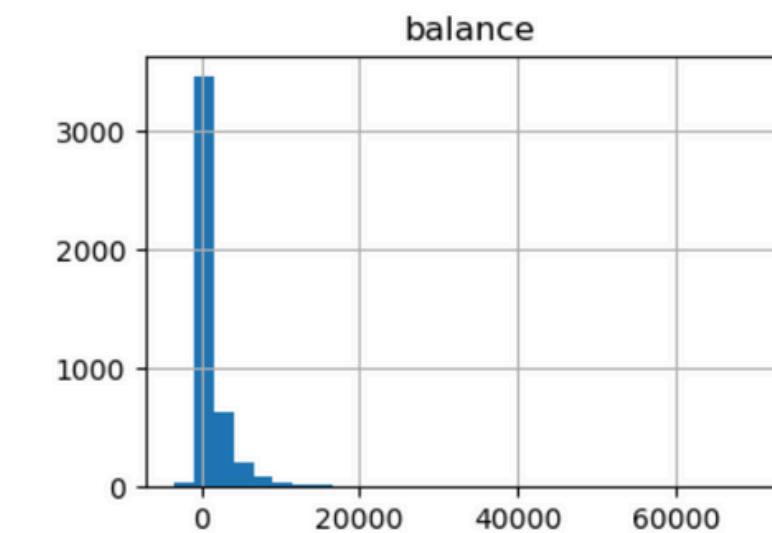
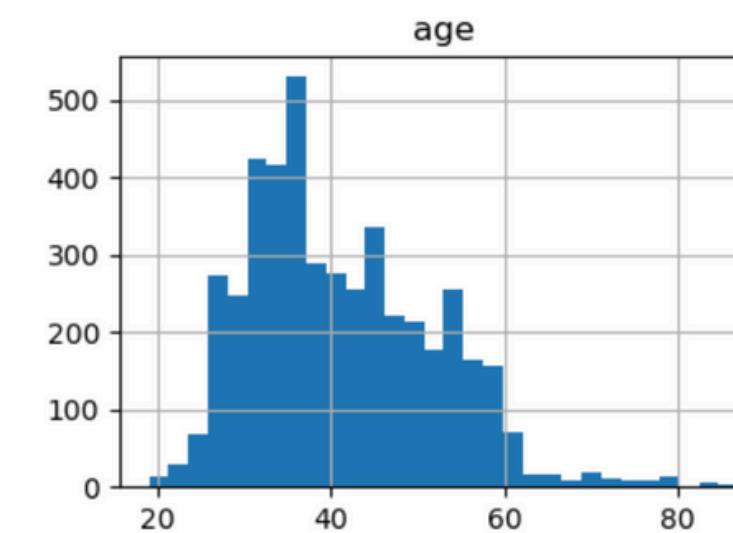
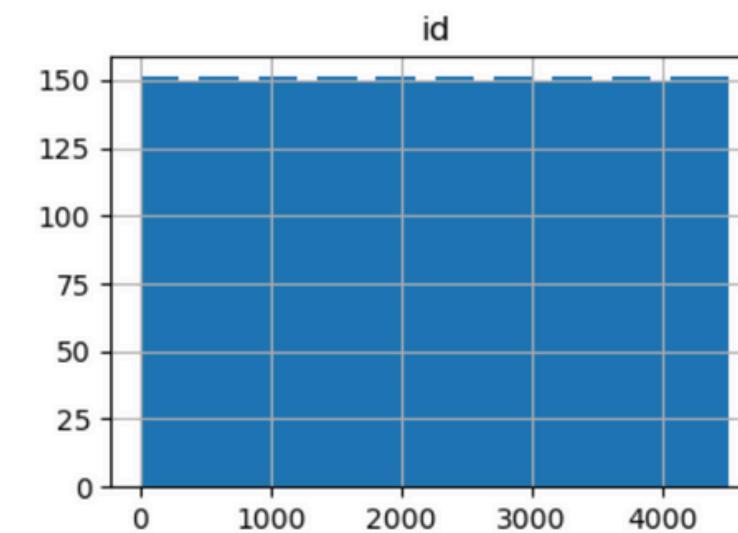


phân phối dữ liệu cột duration



phân phối dữ liệu cột job

Đặc Điểm Dữ Liệu

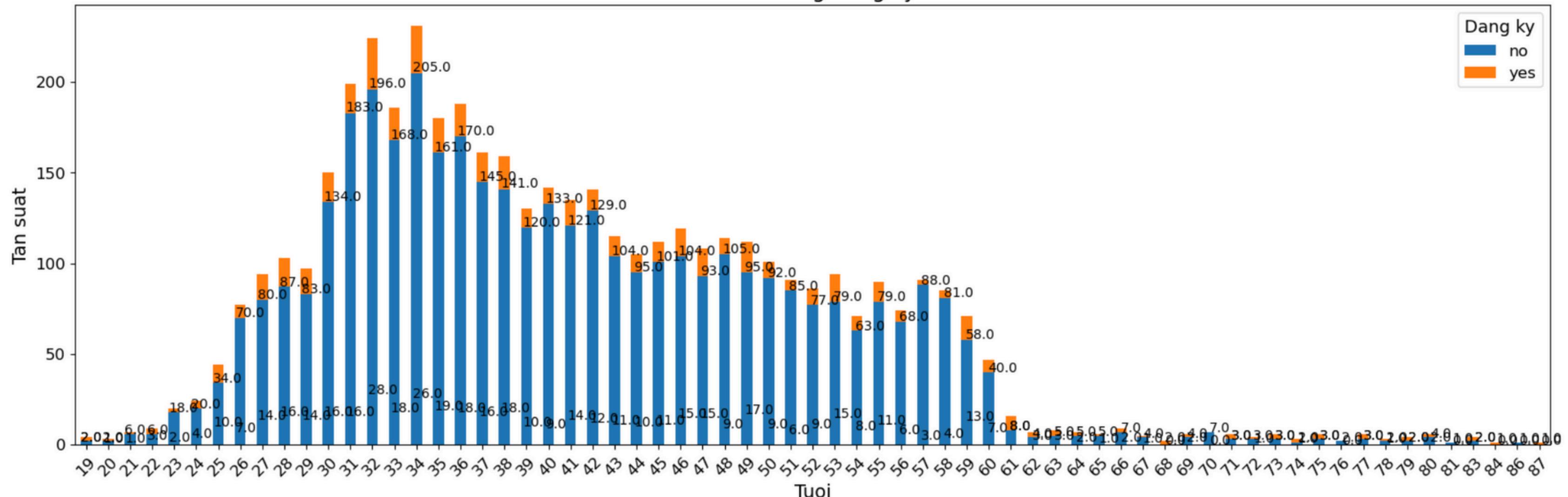


phân phối đặc trưng số

Đặc Điểm Dữ Liệu

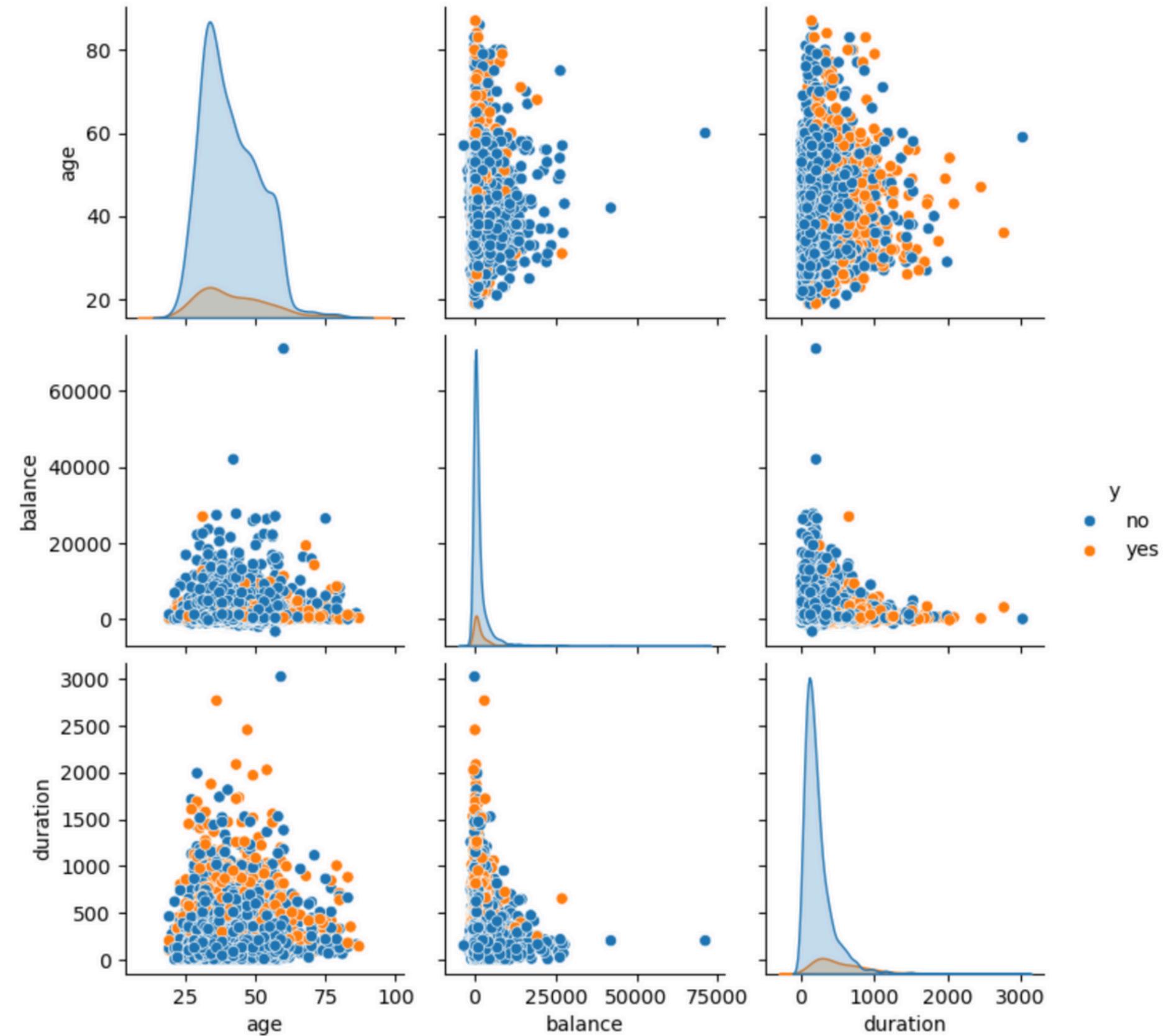
Vài biểu đồ phân phối dữ liệu:

Tần suất khách hàng đăng ký theo tuổi



Tần suất khách hàng theo tuổi

Đặc Điểm Dữ Liệu



Mối quan hệ các đặc trưng

Dữ liệu ban đầu:

X shape: (4521, 7)

y shape: (4521,)

	age	default	balance	duration	campaign	pdays	previous
0	30	no	1787	79	1	-1	0
1	33	no	4789	220	1	339	4
2	35	no	1350	185	1	330	1
3	30	no	1476	199	4	-1	0
4	59	no	0	226	1	-1	0
...
4516	33	no	-333	329	5	-1	0
4517	57	yes	-3313	153	1	-1	0
4518	57	no	295	151	11	-1	0
4519	28	no	1137	129	4	211	3
4520	44	no	1136	345	2	249	7

[4521 rows x 7 columns]

	y
0	no
1	no
2	no
3	no
4	no
...	...
4516	no
4517	no
4518	no
4519	no
4520	no

Name: y, Length: 4521, dtype: object

Tiền Xử Lí Dữ Liệu

Tiến hành chuẩn hoá dữ liệu:

X shape: (4521, 7)

y shape: (4521, 1)

	age	balance	duration	campaign	pdays	previous	default_yes
0	30.0	1787.0	79.0	1	-1	0	False
1	33.0	4789.0	220.0	1	339	4	False
2	35.0	1350.0	185.0	1	330	1	False
3	30.0	1476.0	199.0	4	-1	0	False
4	59.0	0.0	226.0	1	-1	0	False
...
4516	33.0	-333.0	329.0	5	-1	0	False
4517	57.0	-3313.0	153.0	1	-1	0	True
4518	57.0	295.0	151.0	11	-1	0	False
4519	28.0	1137.0	129.0	4	211	3	False
4520	44.0	1136.0	345.0	2	249	7	False

[4521 rows x 7 columns]

	y
0	True
1	True
2	True
3	True
4	True
...	...
4516	True
4517	True
4518	True
4519	True
4520	True

[4521 rows x 1 columns]



Logistic Regression

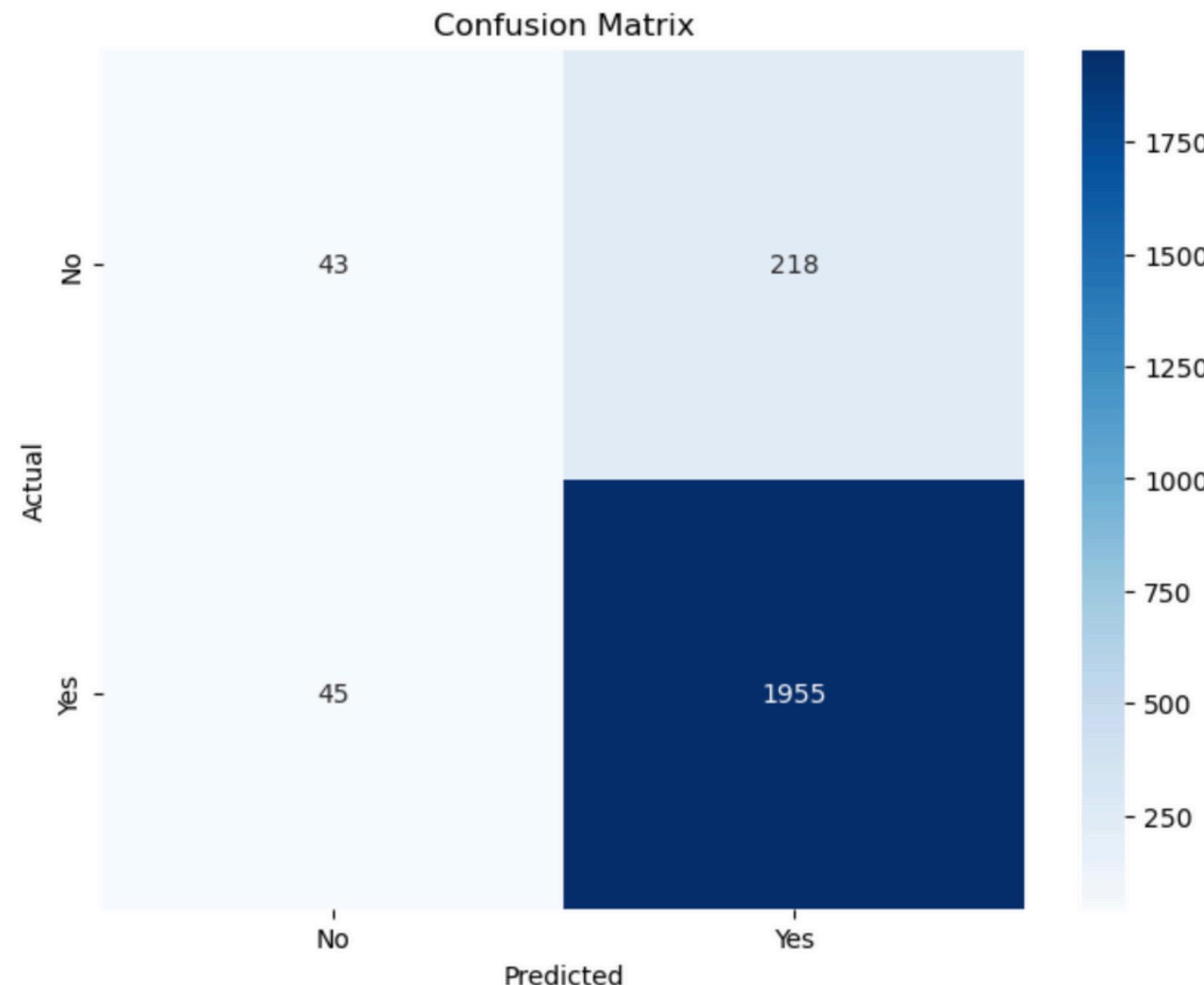
Khái niệm: Logistic Regression là một thuật toán phân loại tuyến tính được sử dụng để dự đoán xác suất của một nhãn phân loại.

Các bước thực hiện:

1. Khởi tạo mô hình: Sử dụng LogisticRegression với random_state=0 và max_iter=500.
2. Huấn luyện mô hình: Huấn luyện trên tập huấn luyện (X_train, y_train).
3. Dự đoán: Dự đoán trên tập kiểm tra (X_test).
4. Tính toán các chỉ số: Accuracy, Precision, Recall, F1 Score.
5. Lưu kết quả: Tạo DataFrame để lưu trữ các chỉ số đánh giá.

Kết Quả:

Logistic Regression



	Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression		0.88368	0.899678	0.9775	0.936976



K-Nearest Neighbors (KNN)

Khái niệm: K-Nearest Neighbors (KNN) là một thuật toán phân loại dựa trên khoảng cách giữa các điểm dữ liệu.

Các bước thực hiện:

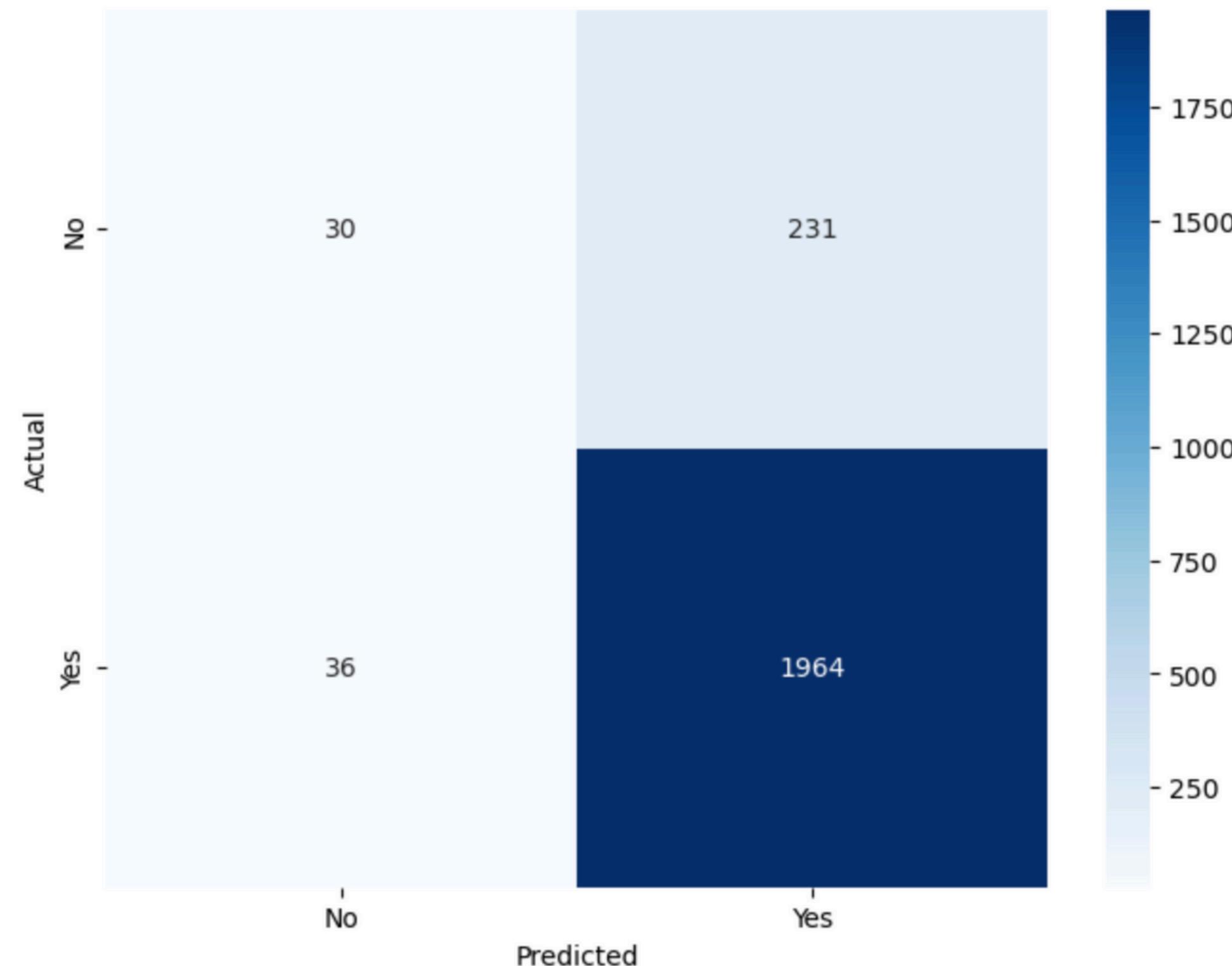
1. Khởi tạo mô hình: Sử dụng KNeighborsClassifier với n_neighbors=15, metric='minkowski', và p=2.
2. Huấn luyện mô hình: Huấn luyện trên tập huấn luyện (X_train và y_train).
3. Dự đoán: Huấn luyện trên tập huấn luyện (X_train và y_train).
4. Tính toán các chỉ số: Accuracy, Precision, Recall, F1 Score.
5. Lưu kết quả: Tạo DataFrame để lưu trữ các chỉ số đánh giá.



Kết Quả:

K-Nearest Neighbors (KNN)

Confusion Matrix



Model	Accuracy	Precision	Recall	F1 Score
KNN	0.881911	0.894761	0.982	0.936353

Decision Tree

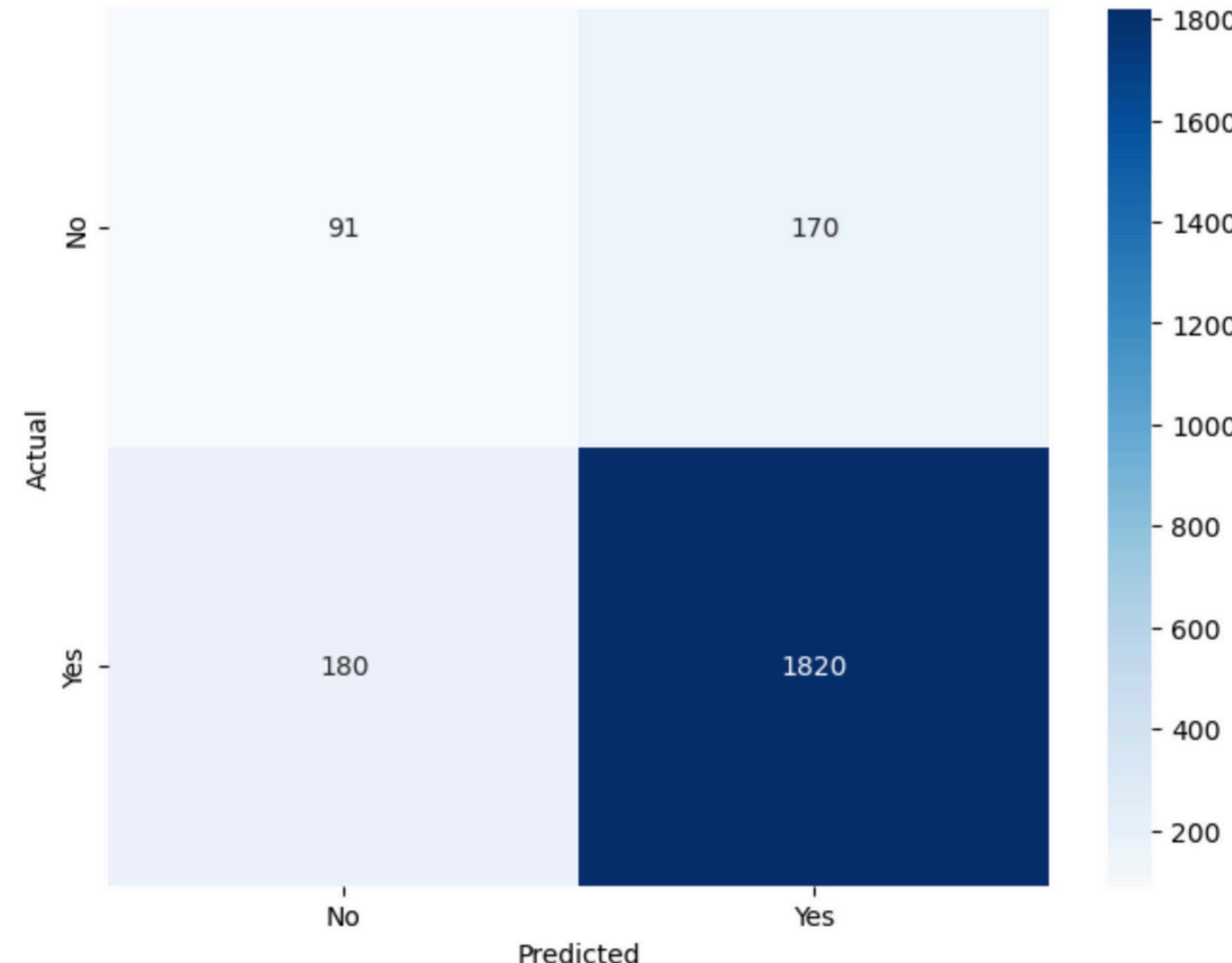
Khái niệm: Decision Tree là một thuật toán phân loại dựa trên cấu trúc cây.

Các bước thực hiện:

1. Khởi tạo mô hình: Sử dụng DecisionTreeClassifier với criterion='entropy' và random_state=0.
2. Huấn luyện mô hình: Huấn luyện trên tập huấn luyện (X_{train} và y_{train}).
3. Dự đoán: Dự đoán trên tập kiểm tra (X_{test}).
4. Tính toán các chỉ số: Accuracy, Precision, Recall, F1 Score.
5. Lưu kết quả: Tạo DataFrame để lưu trữ các chỉ số đánh giá.

Decision Tree

Confusion Matrix

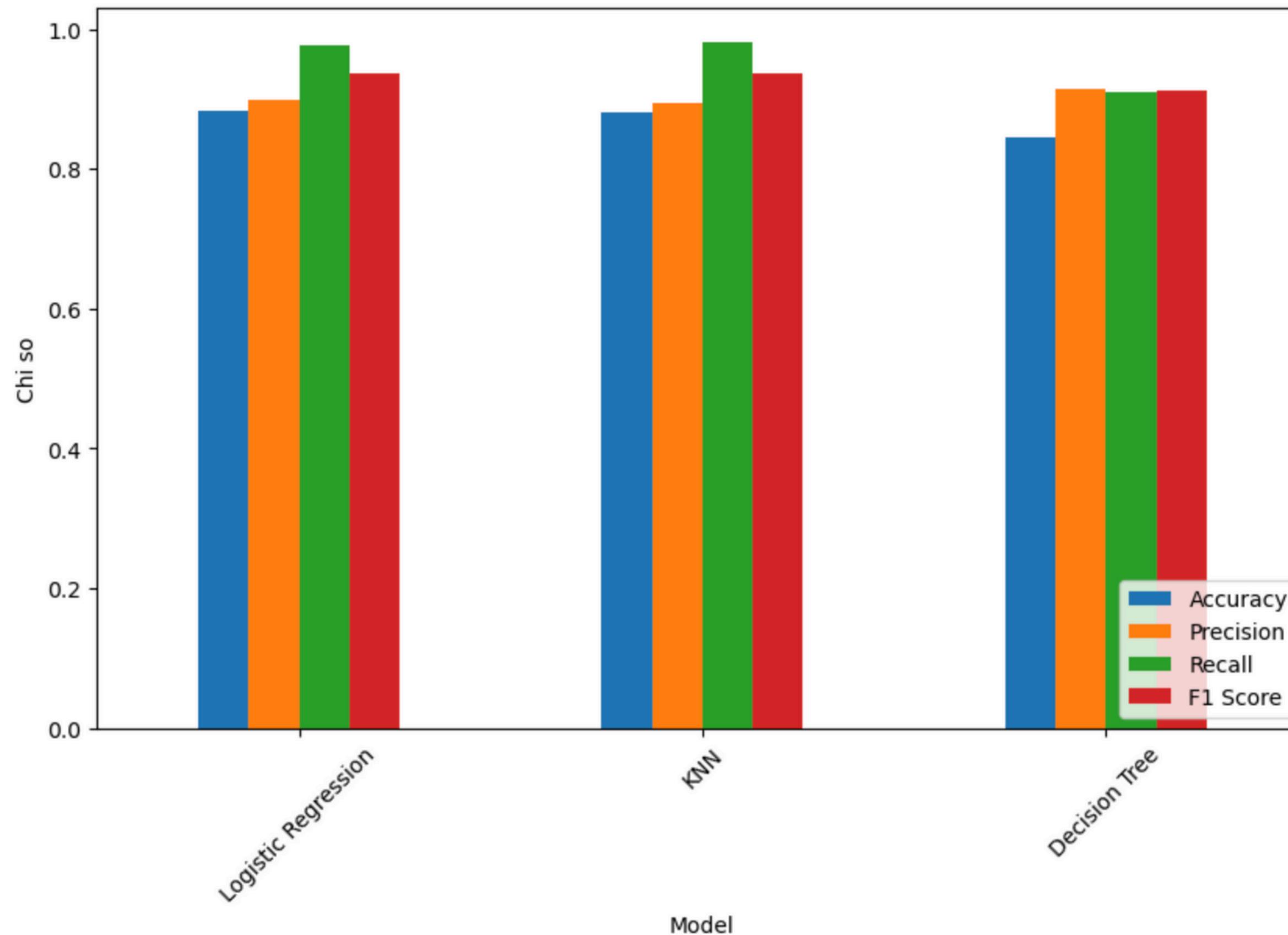


Kết Quả:

Model	Accuracy	Precision	Recall	F1 Score
Decision Tree	0.845201	0.914573	0.91	0.912281

So Sánh Phân Loại

So sánh các mô hình Classification





Linear Regression

Khái niệm: Linear Regression là một thuật toán hồi quy tuyến tính được sử dụng để dự đoán giá trị liên tục dựa trên mối quan hệ tuyến tính giữa các đặc trưng và biến mục tiêu.

Các bước thực hiện:

1. Khởi tạo mô hình: Sử dụng LinearRegression.
2. Huấn luyện mô hình: Huấn luyện trên tập huấn luyện (X_{train} và y_{train})..
3. Dự đoán: Dự đoán trên tập kiểm tra (X_{test}).
4. Tính toán các chỉ số: Accuracy, Precision, Recall, F1 Score.
5. Lưu kết quả: Tạo DataFrame để lưu trữ các chỉ số đánh giá.

Kết Quả:

Linear Regression RMSE: 1.2567440725831558

Linear Regression R²: 0.44970478858589935



Random Forest Regression

Khái niệm: Random Forest Regression là một thuật toán hồi quy dựa trên nhiều cây quyết định. Nó sử dụng một tập hợp các cây quyết định để dự đoán giá trị liên tục, giúp cải thiện độ chính xác và giảm overfitting.

Các bước thực hiện:

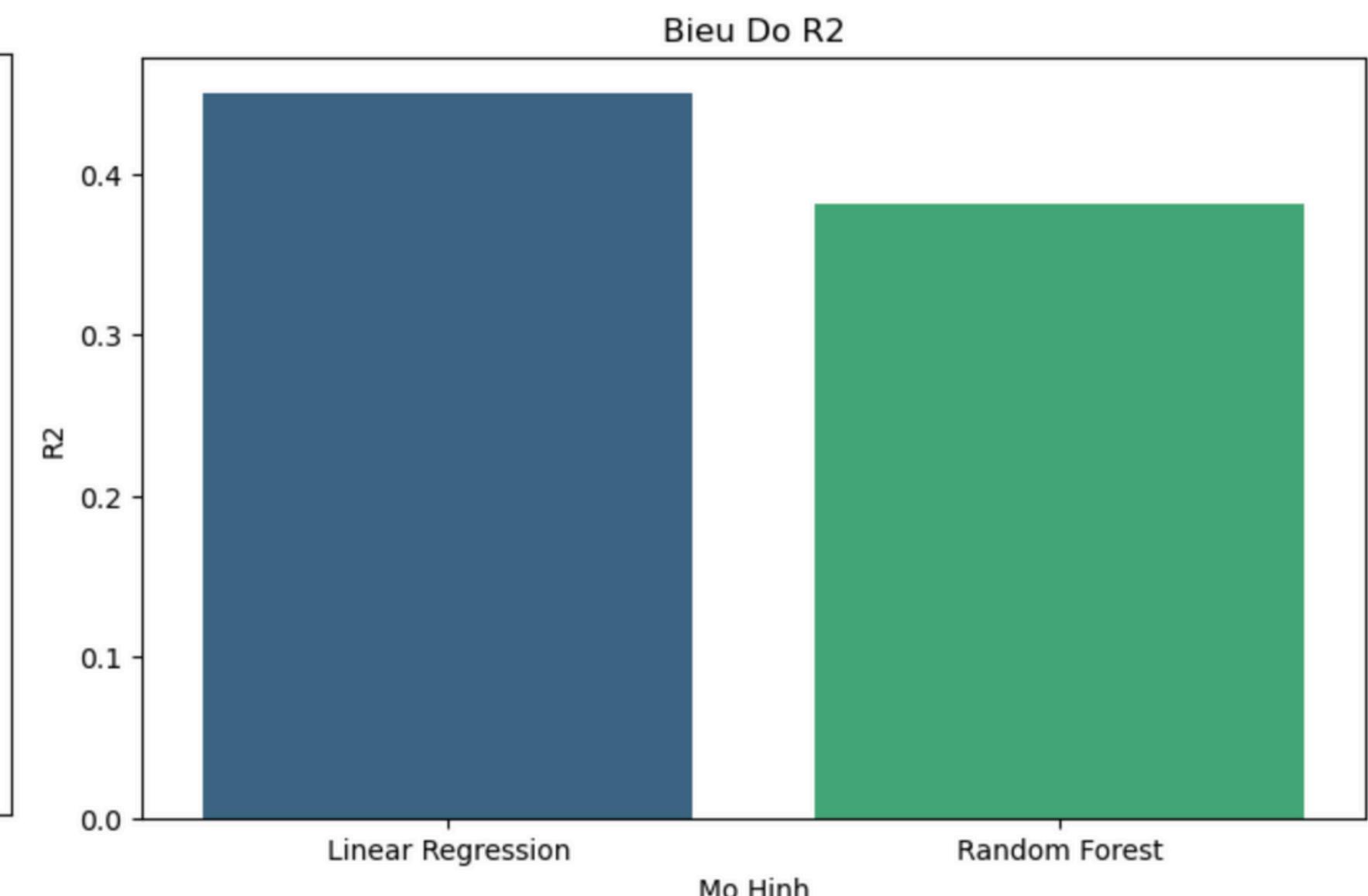
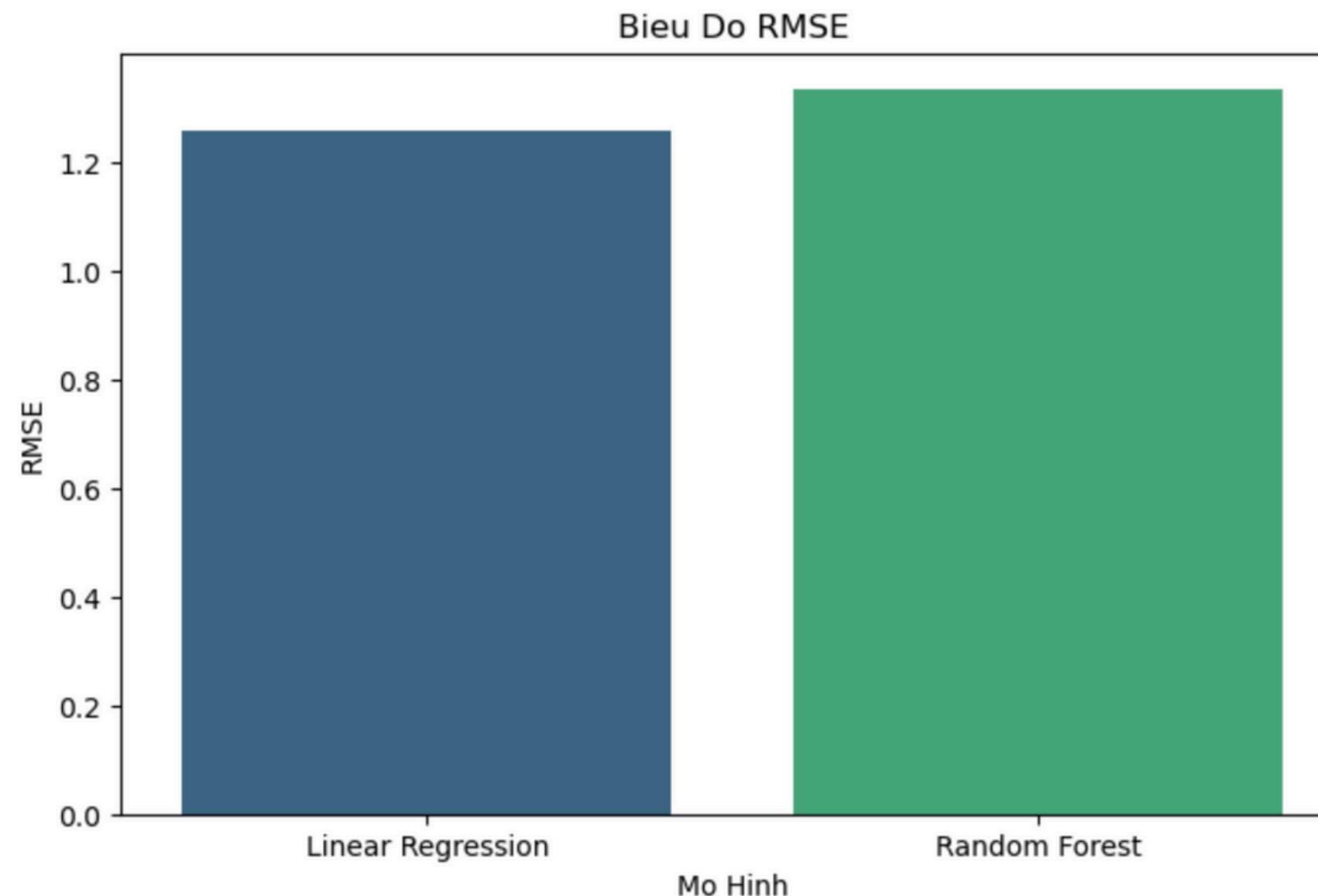
1. Khởi tạo mô hình: Sử dụng RandomForestRegressor với random_state=42.
2. Huấn luyện mô hình: Huấn luyện trên tập huấn luyện (X_{train} và y_{train})..
3. Dự đoán: Dự đoán trên tập kiểm tra (X_{test}).
4. Tính toán các chỉ số: Accuracy, Precision, Recall, F1 Score.
5. Lưu kết quả: Tạo DataFrame để lưu trữ các chỉ số đánh giá.

Kết Quả:

Random Forest RMSE: 1.3327852539094323

Random Forest R²: 0.3810972669124737

So Sánh Hồi Quy



	Mô Hình	RMSE	R^2
0	Linear Regression	1.256744	0.449705
1	Random Forest	1.332785	0.381097

Overfitting

Khái Niệm: Overfitting là một vấn đề khá là phổ biến, xảy ra khi mô hình học quá kỹ các chi tiết và nhiễu trong dữ liệu huấn luyện (để test quá cao), dẫn đến hiệu suất kém trên dữ liệu kiểm tra hoặc dữ liệu mới. Mô hình overfitting có thể có độ chính xác rất cao trên dữ liệu huấn luyện nhưng lại hoạt động kém trên dữ liệu kiểm tra.

Nguyên nhân:

1. Mô hình quá phức tạp: Mô hình có quá nhiều tham số hoặc độ phức tạp cao.
2. Dữ liệu huấn luyện không đủ: Số lượng mẫu trong dữ liệu huấn luyện quá ít so với số lượng đặc trưng.
3. Nhiễu trong dữ liệu: Dữ liệu huấn luyện chứa nhiều nhiễu hoặc giá trị ngoại lai.



Giải Pháp Overfitting

Giải Pháp Khắc Phục Overfitting

1. Regularization:

- Thêm thuật ngữ phạt vào hàm mục tiêu để giảm overfitting.
- Ví dụ: L1 (Lasso), L2 (Ridge).

2. Cross-Validation:

- Sử dụng kỹ thuật cross-validation để đánh giá mô hình một cách chính xác hơn.

3. Pruning (Decision Tree):

- Giới hạn độ sâu cây (max_depth).
- Số lượng mẫu tối thiểu để tách (min_samples_split).
- Số lượng mẫu tối thiểu tại một lá (min_samples_leaf).

4. Tuning Hyperparameters:

- Tinh chỉnh các siêu tham số của mô hình để tìm ra cấu hình tốt nhất.

Giải Pháp Khắc Phục Overfitting Cho Các Phương Pháp Ở Câu 1

1. K-Nearest Neighbors (KNN):

Tăng giá trị k: Sử dụng giá trị k lớn hơn để giảm độ nhạy của mô hình đối với nhiễu.

Chuẩn hóa dữ liệu: Sử dụng chuẩn hóa dữ liệu để đảm bảo rằng tất cả các đặc trưng đều có cùng thang đo.

2. Logistic Regression:

Sử dụng regularization (L1, L2): Thêm thuật ngữ phạt vào hàm mục tiêu để giảm overfitting.

3. Random Forest Regression:

Sử dụng nhiều cây hơn (n_estimators): Tăng số lượng cây trong rừng để giảm overfitting.

Giới hạn độ sâu cây (max_depth): Giới hạn độ sâu tối đa của mỗi cây quyết định.

4. Decision Tree:

Giới hạn độ sâu cây (max_depth).

Số lượng mẫu tối thiểu để tách (min_samples_split).

Số lượng mẫu tối thiểu tại một lá (min_samples_leaf).

5. Linear Regression

Sử dụng Ridge Regression.

Áp Dụng Giải Pháp cho Decision Tree

1. Recursive Feature Elimination (RFE) là một kỹ thuật chọn lọc đặc trưng(feature selection) giúp giảm số lượng đặc trưng đầu vào bằng cách loại bỏ dần các đặc trưng ít quan trọng nhất.

Chỉ giữ lại các đặc trưng quan trọng nhất ---> giảm thiểu overfitting và cải thiện hiệu suất Model

2. Giới hạn độ sâu cây (max_depth), Bằng cách giới hạn độ sâu ---> ngăn cây quyết định học quá kỹ các chi tiết và nhiễu trong dữ liệu huấn luyện ---> giảm thiểu overfitting

3. Số lượng mẫu tối thiểu để tách (min_samples_split), xác định số lượng mẫu tối thiểu cần thiết để tách một nút.

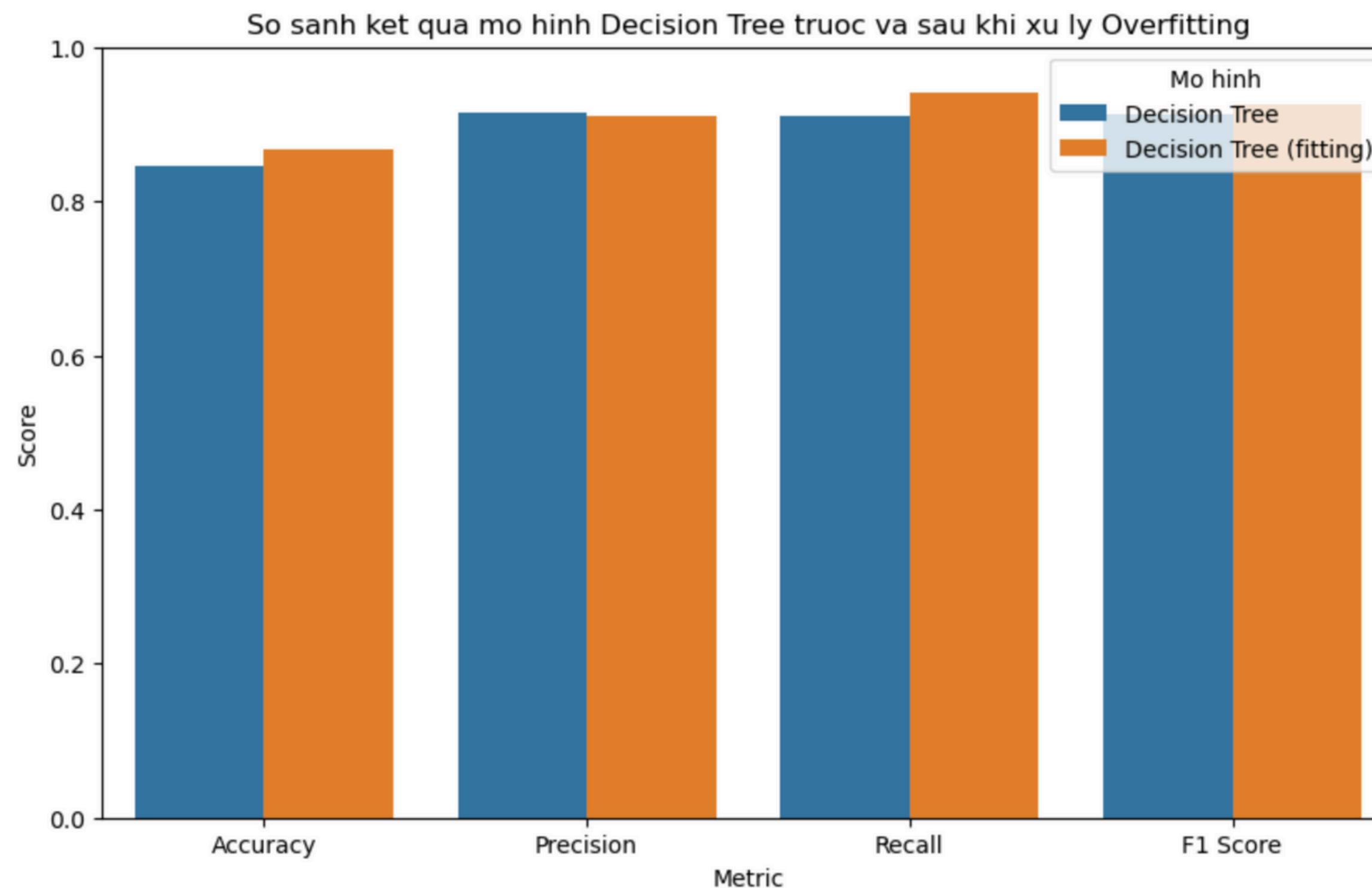
Bằng cách tăng giá trị này ---> ngăn cây quyết định tách quá nhiều lần ---> giảm thiểu overfitting

4. Số lượng mẫu tối thiểu tại một lá (min_samples_leaf), xác định số lượng mẫu tối thiểu cần thiết tại một lá.

Bằng cách tăng giá trị này ---> ngăn cây quyết định tạo ra các lá quá nhỏ ---> giảm thiểu overfitting.

```
criterion='entropy',
max_depth=10,          # Gioi han do sau cay
min_samples_split=10,   # So luong mau toi thieu de tach
min_samples_leaf=5,     # So luong mau toi thieu tai mot la
random_state=0
```

Kết Quả



Model	Accuracy	Precision	Recall	F1 Score
Decision Tree	0.845201	0.914573	0.910000	0.912281
Decision Tree (fitting)	0.867403	0.911729	0.941323	0.926290



Áp Dụng Giải Pháp cho Linear Regression

Ridge Regression là một kỹ thuật regularization bằng cách thêm một thuật ngữ phạt vào hàm mục tiêu của Linear Regression. Điều này giúp giảm overfitting bằng cách hạn chế độ lớn của các hệ số hồi quy.

Lý do chọn Ridge?

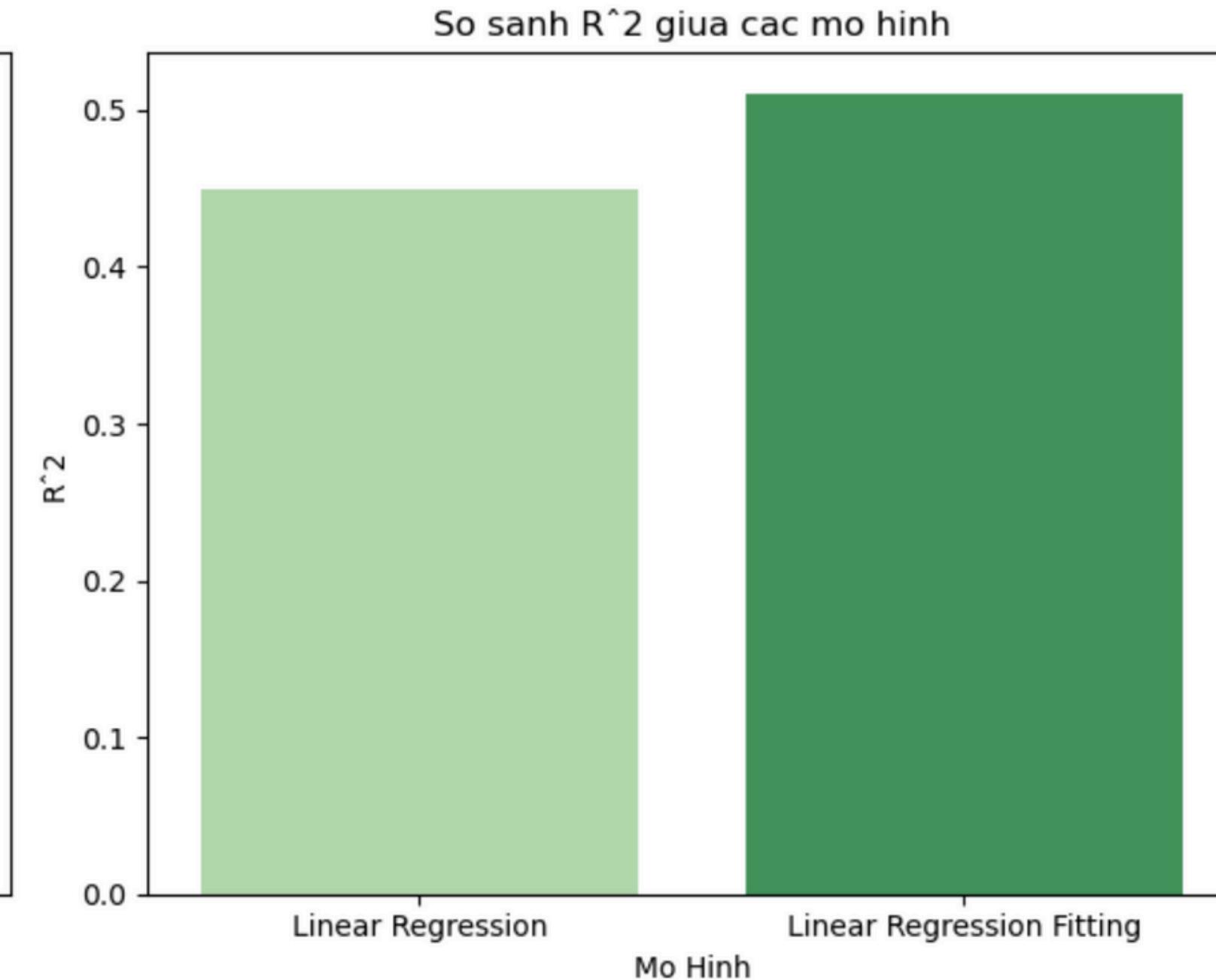
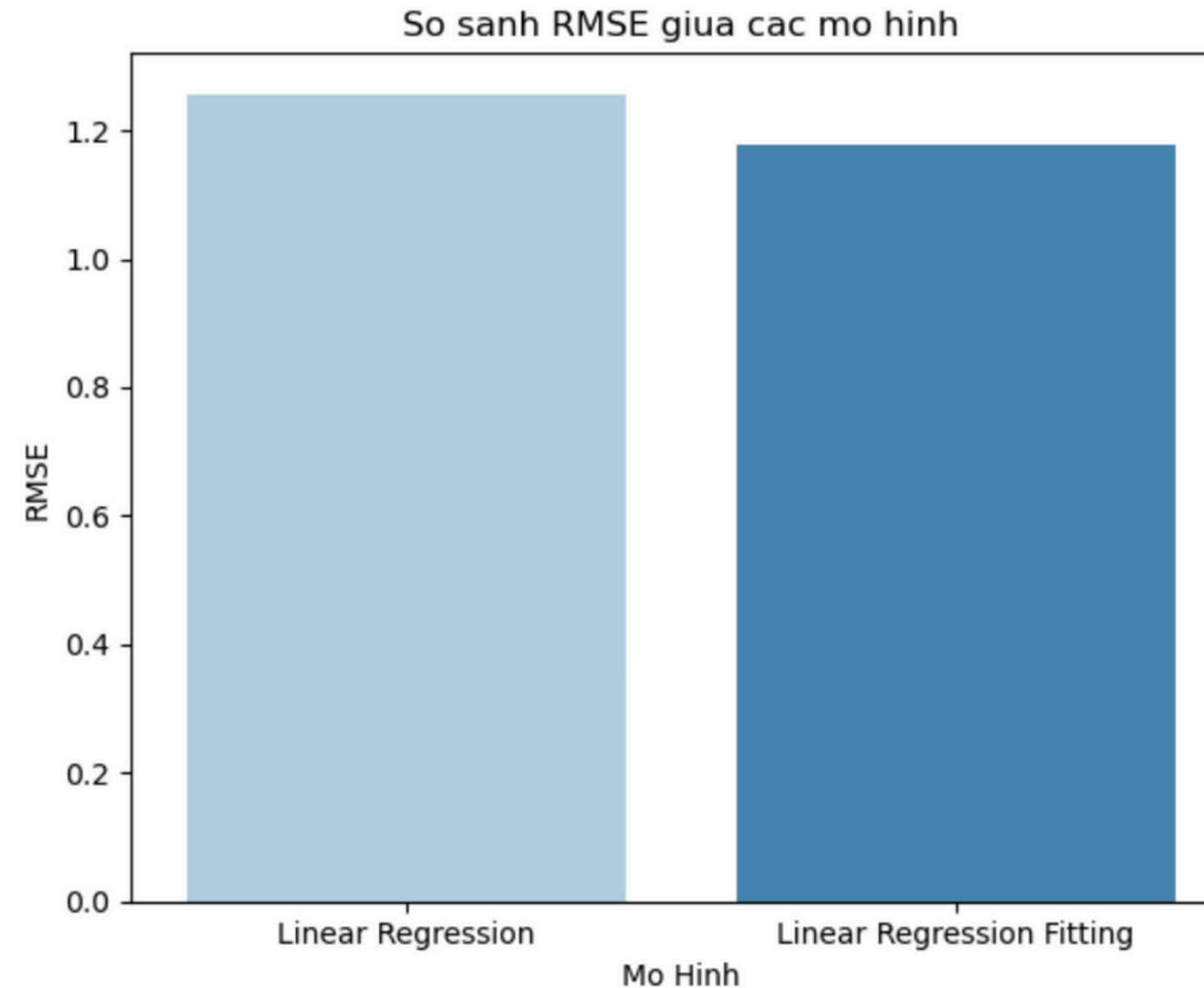
Ridge Regression giúp giảm overfitting bằng cách thêm một thuật ngữ phạt vào hàm mục tiêu, làm giảm độ lớn của các hệ số hồi quy.
--> Tăng hiệu suất.

Best Ridge Alpha là giá trị của tham số alpha trong Ridge Regression mà Grid Search đã tìm thấy là tốt nhất cho mô hình. Tham số alpha trong Ridge Regression điều chỉnh mức độ regularization được áp dụng cho mô hình.

Áp Dụng Giải Pháp cho Linear Regression

1. Thiết lập các tham số cho Ridge Regression: : Sử dụng hàm np.logspace để tạo ra 90 giá trị alpha từ (10^{-4}) đến (10^4) .
- 2.Ridge Regression với tìm kiếm Grid Search:--> tìm kiếm giá trị alpha tốt nhất.
- 3 Sử dụng ridge_grid_search.best_estimator_ để lấy mô hình có giá trị alpha tốt nhất
- 4.Sử dụng mô hình Ridge Regression tốt nhất để dự đoán trên tập kiểm tra.

Kết Quả



Linear Regression	1.256744	0.449705
Linear Regression Fitting	1.178445	0.510961



Feature selection using correlation analysis

Khái niệm:

Feature Selection là quá trình chọn ra những đặc trưng (features) có ảnh hưởng lớn nhất đến biến mục tiêu trong một tập dữ liệu. Việc này giúp cải thiện độ chính xác của mô hình, giảm thiểu độ phức tạp và thời gian huấn luyện.

Correlation Analysis là phương pháp giúp xác định mối quan hệ giữa các đặc trưng và biến mục tiêu. Tương quan thường được đo bằng hệ số tương quan Pearson (cho các biến liên tục) hoặc hệ số tương quan Spearman (cho các biến thứ tự).

Tại sao Feature Selection Quan Trọng?

1.1 Giảm thiểu overfitting: Mô hình phức tạp với quá nhiều đặc trưng dễ bị overfitting, tức là mô hình quá khớp với dữ liệu huấn luyện mà không tổng quát được cho dữ liệu mới.

1.2 Tăng cường hiệu suất: Việc chọn lựa các đặc trưng quan trọng giúp mô hình hoạt động hiệu quả hơn.

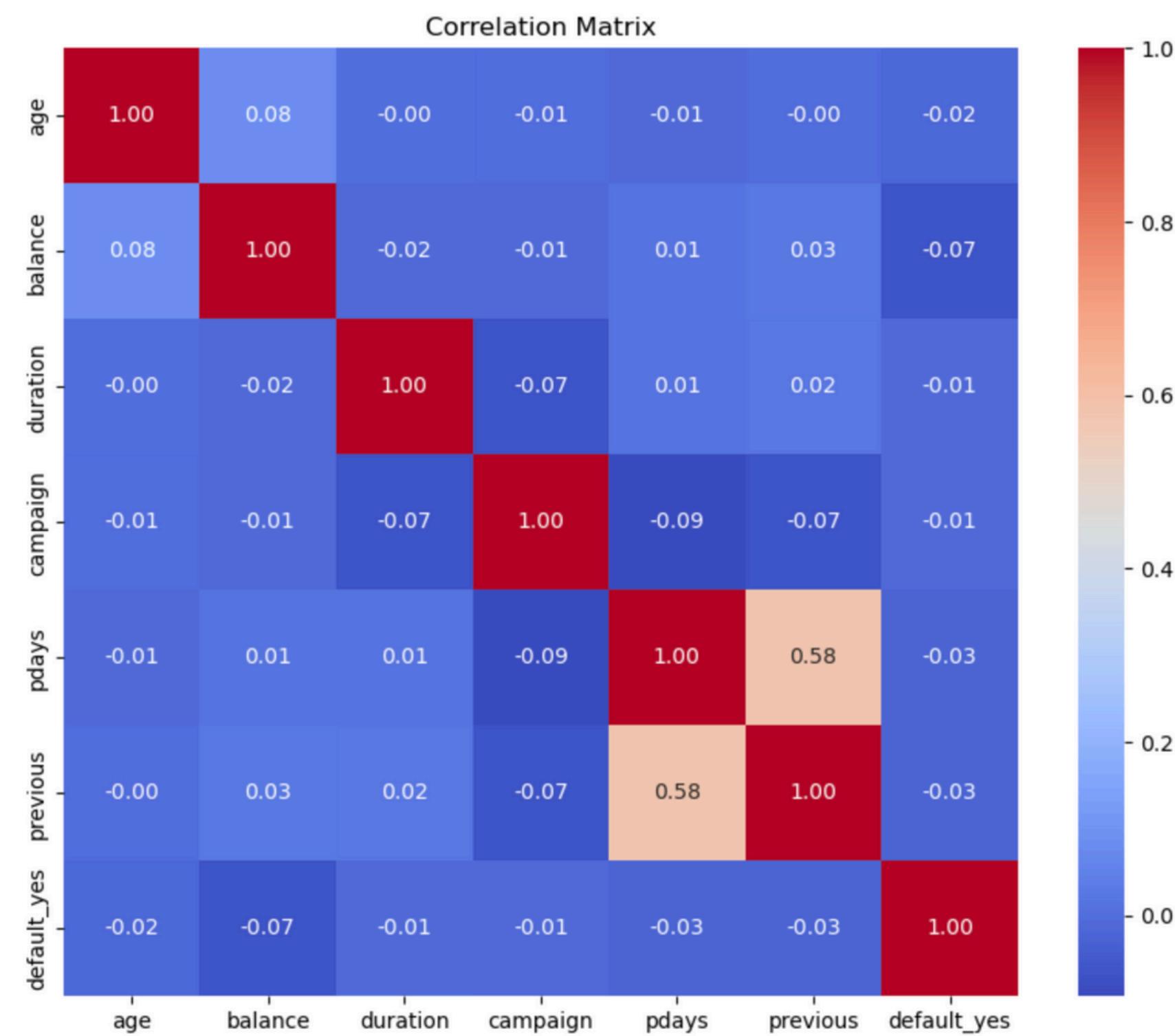
1.3 Tiết kiệm thời gian và tài nguyên: Một mô hình đơn giản hơn với ít đặc trưng sẽ giảm thiểu thời gian huấn luyện và tài nguyên tính toán.

Feature selection using correlation analysis

Các bước thực hiện:

1. Tính toán ma trận tương quan: `correlation_matrix = X.corr()` (phương pháp Pearson).
2. Hiển thị ma trận tương quan bằng heatmap: Sử dụng thư viện Seaborn để vẽ heatmap hiển thị mối tương quan giữa các đặc trưng.
3. Đặt ngưỡng tương quan: `correlation_threshold` (Chỉ những đặc trưng có hệ số tương quan tuyệt đối lớn hơn `correlation_threshold` với biến mục tiêu mới được chọn)
3. Tính toán hệ số tương quan giữa từng đặc trưng và biến mục tiêu: Sử dụng hàm ‘apply’ để áp dụng hàm corr cho từng cột trong X và tính toán hệ số tương quan tuyệt đối với biến mục tiêu y.
5. Chọn các đặc trưng có mối tương quan cao với biến mục tiêu:Lọc các đặc trưng có hệ số tương quan tuyệt đối lớn hơn ngưỡng đã đặt và lưu danh sách các đặc trưng này.
6. Loại bỏ các đặc trưng có mối tương quan cao với nhau: Loại bỏ các đặc trưng có mối tương quan cao với nhau để tránh đa cộng tuyến (multicollinearity).

Decision Tree



Ma trận tương quan

Decision Tree

*Đặt ngưỡng tương quan để chọn lọc đặc trưng, **correlation_threshold** = 0.1*

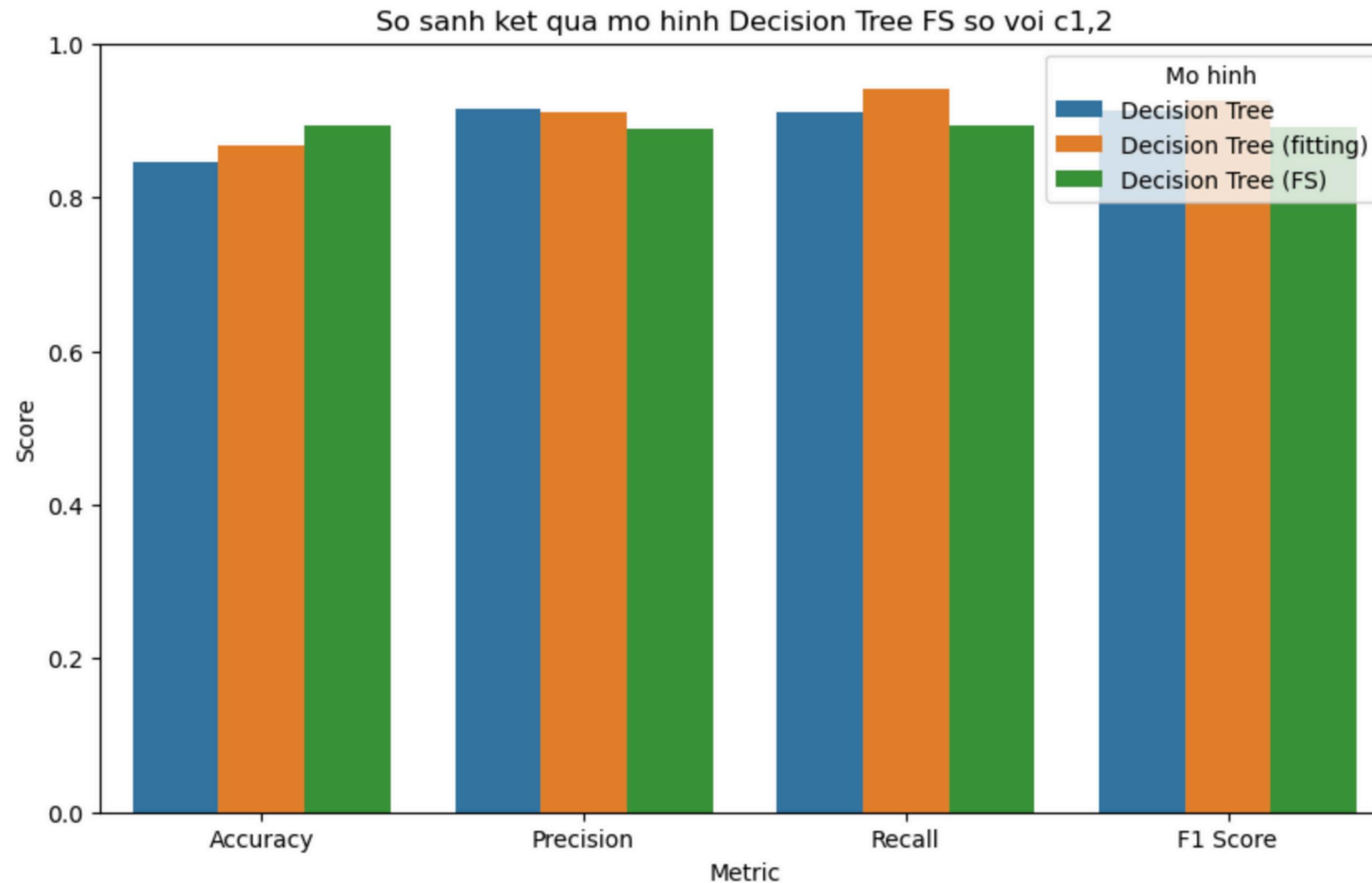
Các đặc trưng được lựa chọn:

Chon cac dt: ['duration', 'pdays', 'previous']

Kết quả đạt được khi áp dụng:

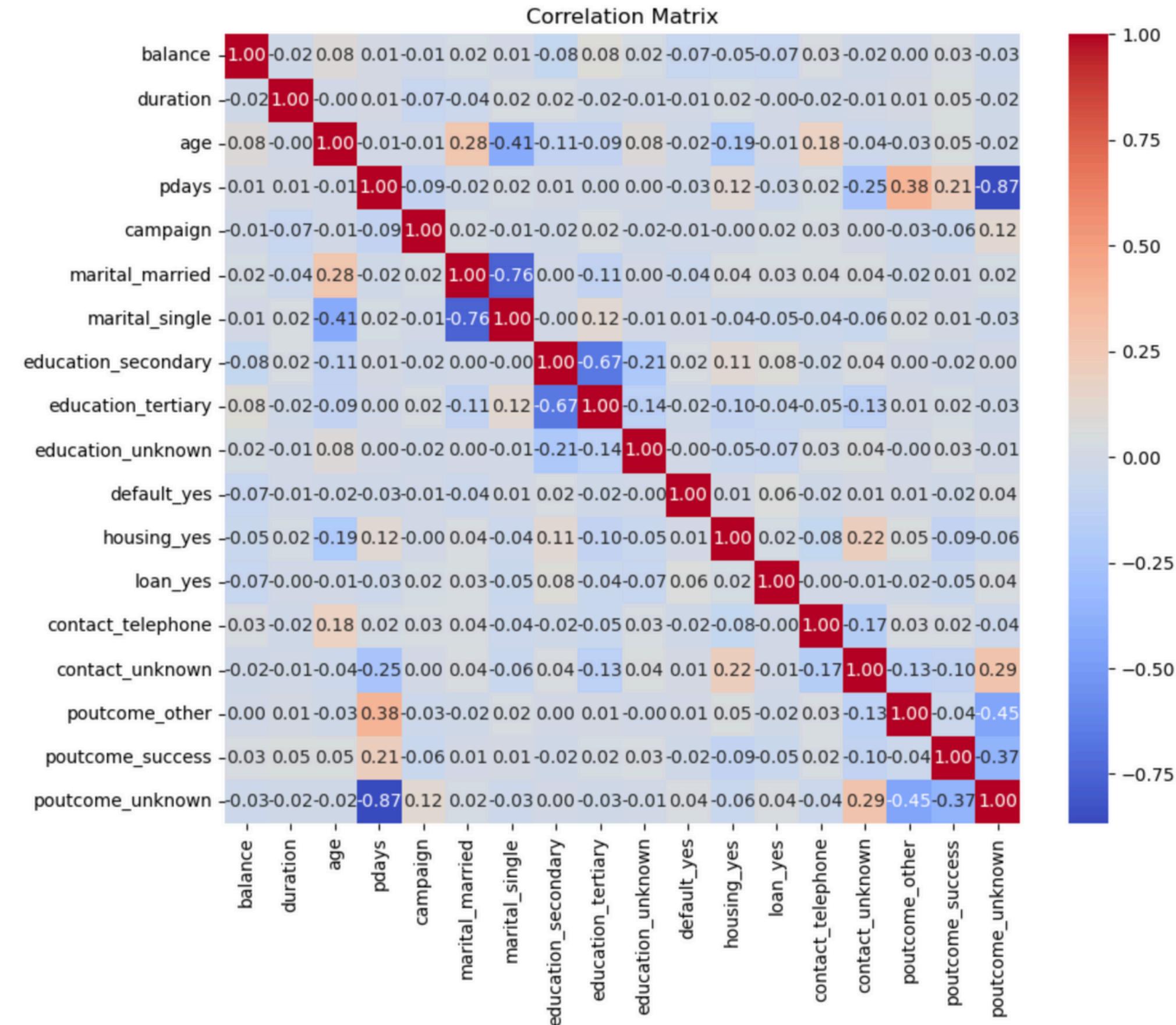
Model	Accuracy	Precision	Recall	F1 Score
Decision Tree (FS)	0.893923	0.888321	0.893923	0.890855

Decision Tree



Decision Tree	0.845201	0.914573	0.910000	0.912281
Decision Tree (fitting)	0.867403	0.911729	0.941323	0.926290
Decision Tree (FS)	0.893923	0.888321	0.893923	0.890855

Linear Regression



Ma trận tương quan



Linear Regression

*Đặt ngưỡng tương quan để chọn lọc đặc trưng, **correlation_threshold** = 0.3*

Thông qua chọn lọc và loại bỏ, thì các đặc trưng được lựa chọn:

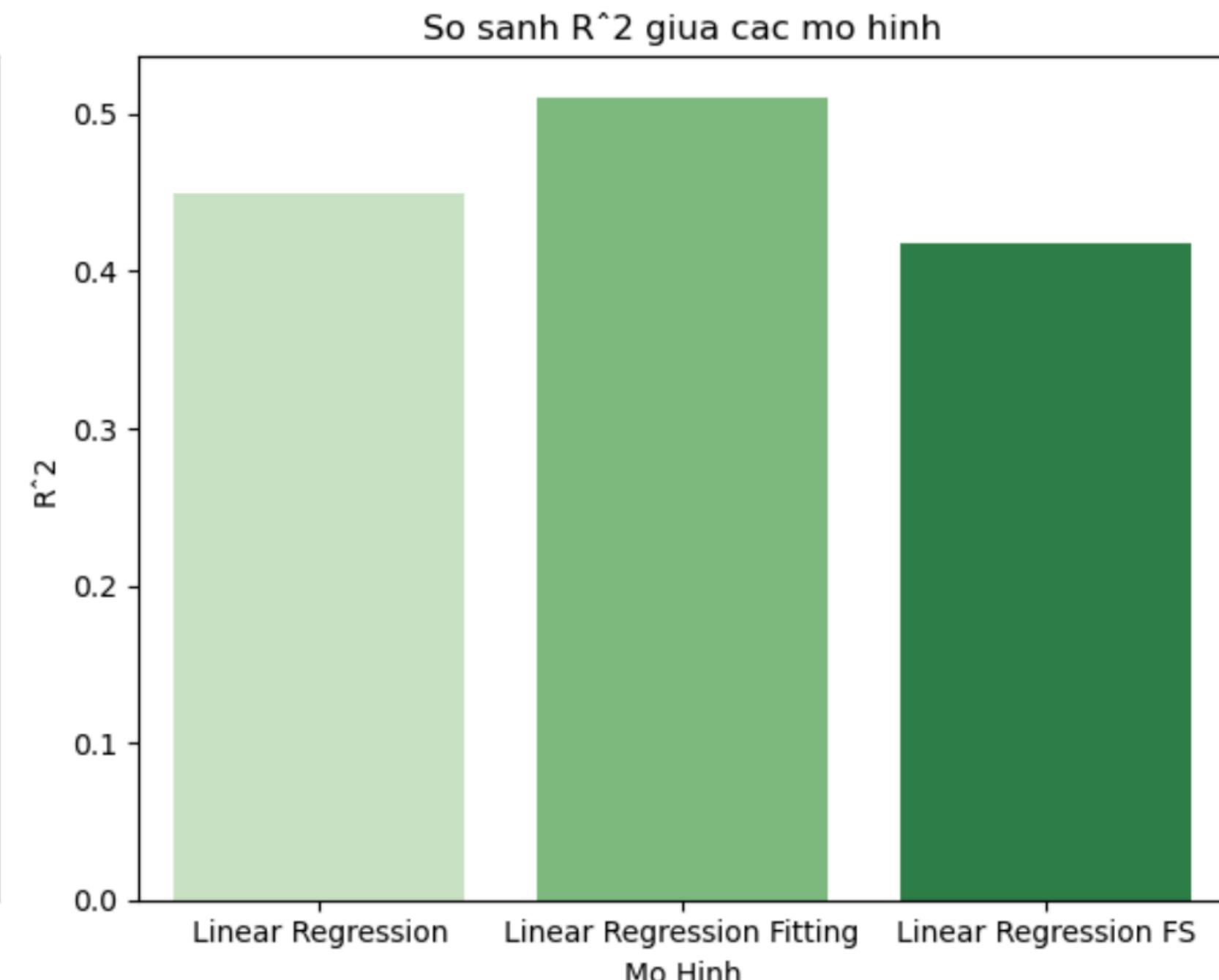
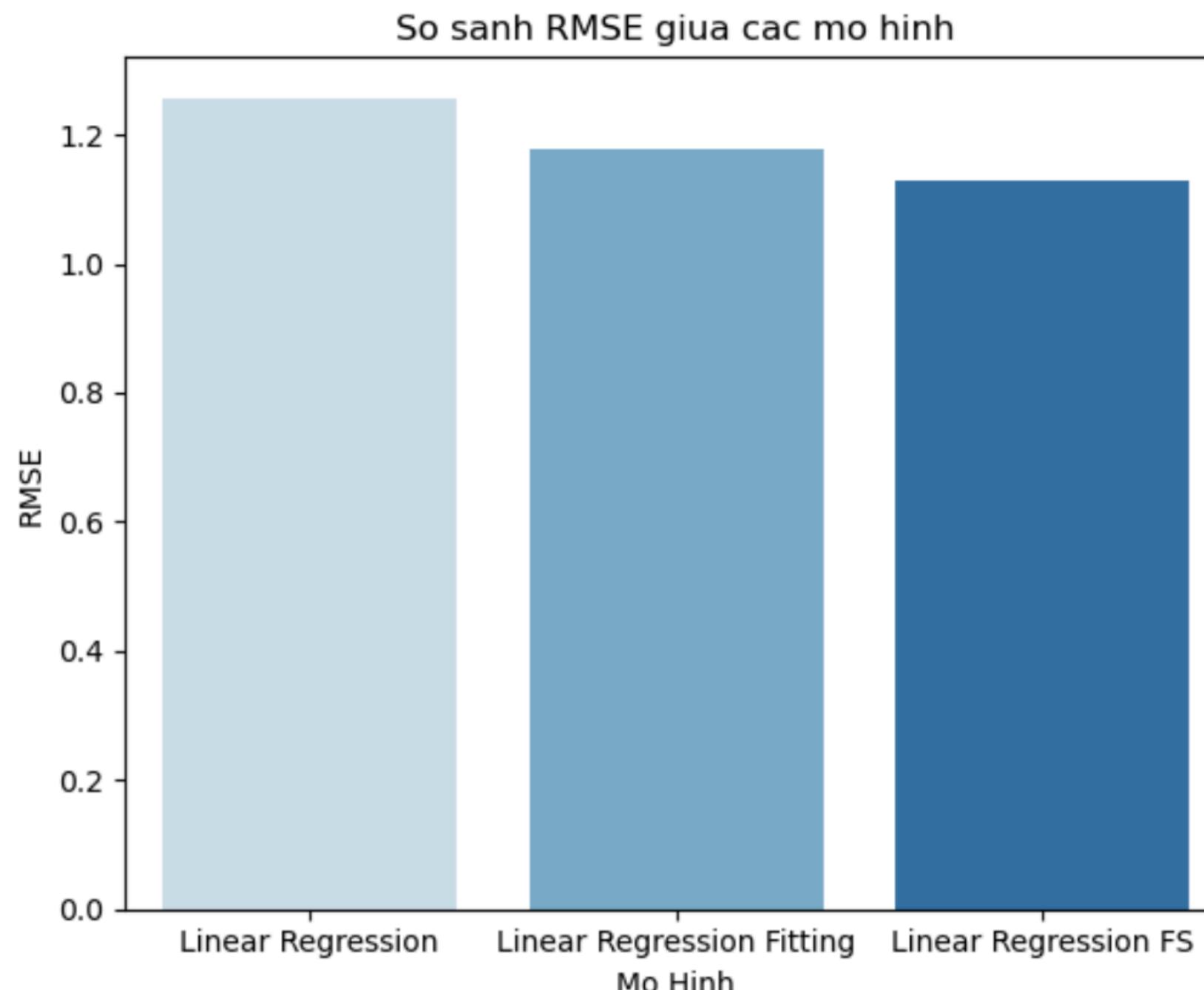
Chon cac dt: ['pdays', 'poutcome_other']

Kết quả đạt được khi áp dụng:

Linear Regression RMSE: 1.1282281302636747

Linear Regression R²: 0.41720812959733755

Linear Regression



Mo Hin	RMSE	R^2
Linear Regression	1.256744	0.449705
Linear Regression Fitting	1.178445	0.510961
Linear Regression FS	1.128228	0.417208



Học Máy

Thanks for your listening