

DỰ ĐOÁN DOANH SỐ BÁN HÀNG THÔNG QUA CÁC MÔ HÌNH HỌC MÁY

Thuật toán Linear Regression, K-Nearest Neighbors, Decision Tree, Random Forest



NỘI DUNG CHÍNH

- 01** Giới thiệu tổng quan đề tài
- 02** Nghiên cứu liên quan
- 03** Mô hình dự đoán doanh thu
- 04** Kết quả thực nghiệm và đề xuất
- 05** Kết luận và hướng phát triển



1. GIỚI THIỆU TỔNG QUAN ĐỀ TÀI

Lý do chọn đề tài

- Xây dựng chiến lược phát triển
- Giảm chi phí, cải thiện doanh số
- Dự báo cảm tính dựa trên kinh nghiệm
- Phụ thuộc nhân sự dự đoán

Mục tiêu đề tài

- Cách thức chuyển đổi dữ liệu
- Phân tích khai phá dữ liệu, tìm insight
- Xác định yếu tố ảnh hưởng đến Sales
- Xác định mô hình dự đoán hiệu quả nhất

Phương pháp nghiên cứu

- Phương pháp xử lý dữ liệu
- Phương pháp học máy



2. NGHIÊN CỨU LIÊN QUAN

2.1. Sales Forecasting for Retail Chains

- Data: Rossmann Data
- Các thuật toán sử dụng:
 - Linear Regression
 - Random Forest Regression
 - XGBoost

Model	RMPSE on Test Set
Linear Regression	0,15672
Random Forest Regression	0,13198
XGBoost	0,10532

2. NGHIÊN CỨU LIÊN QUAN

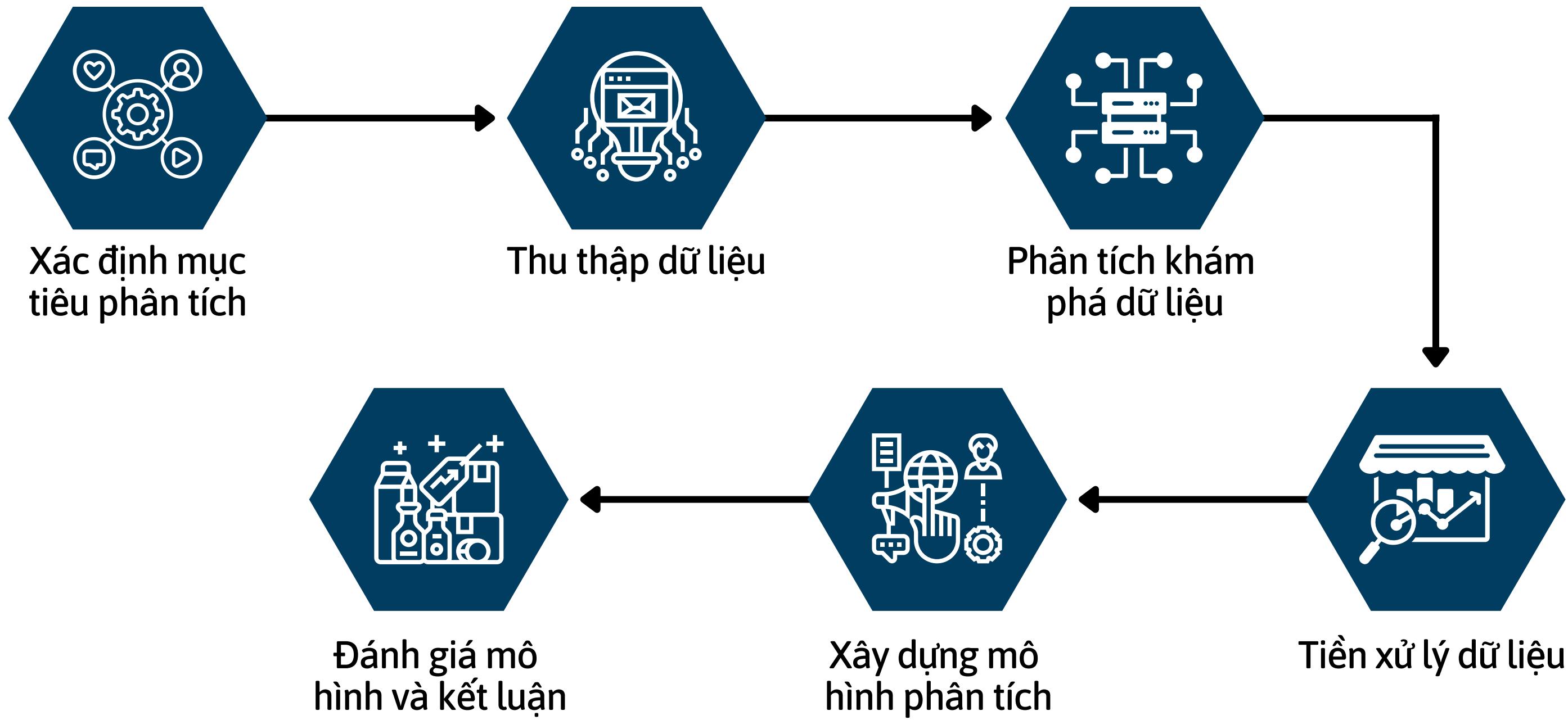
2.2. Walmart Sales Prediction Based on Decision Tree, Random Forest, and K Neighbors Regressor

- Các thuật toán sử dụng:
 - Decision Tree Regressor
 - Random Forest Regressor
 - K-Neighbors Regressor
- Data: Walmart Recruiting - Store Sales

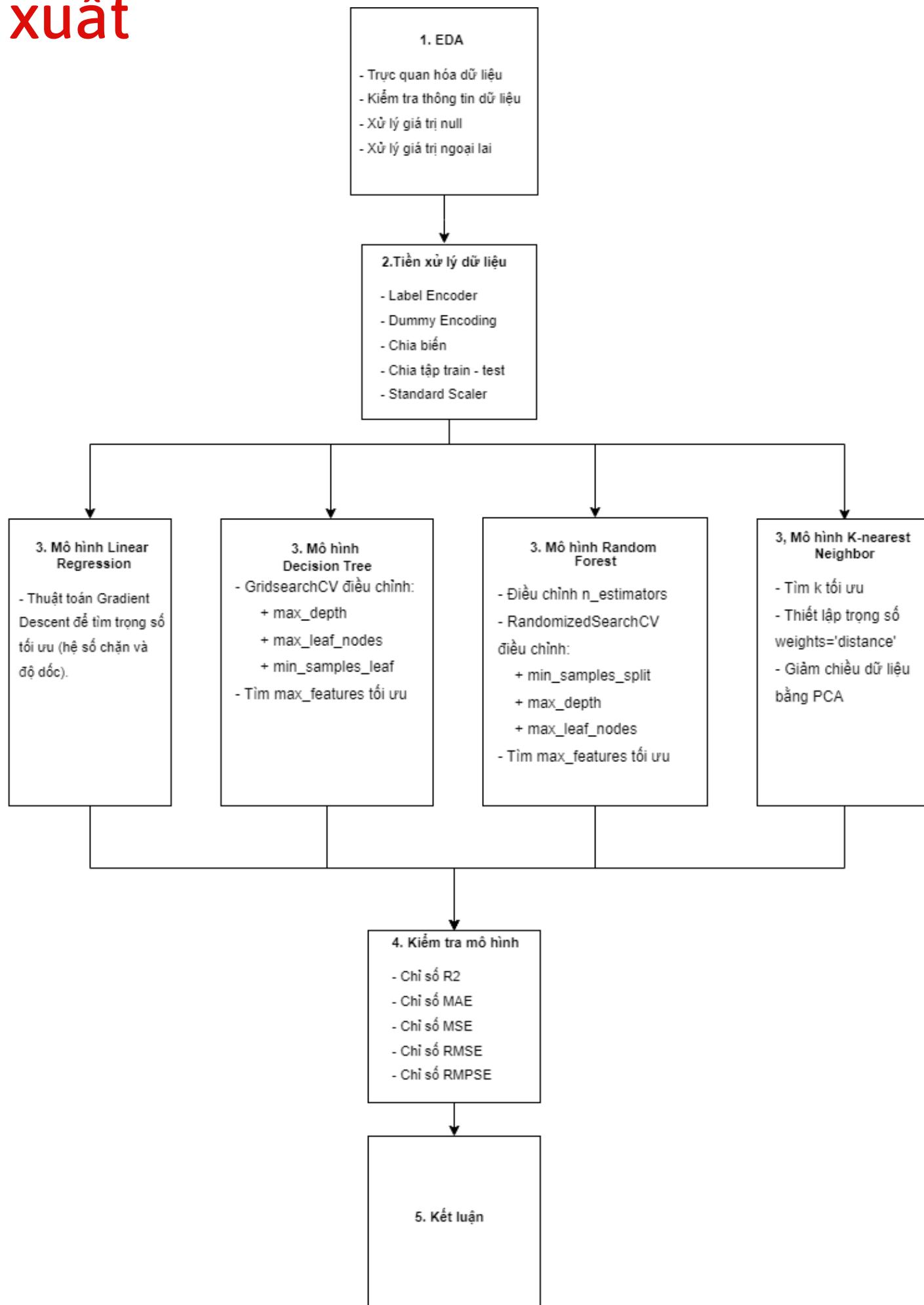
	R ²	MAE	MSE
Decision Tree Regressor	0.905	2377.970	49832386.628
Random Forest Regressor	0.937	1937.810	32993323.634
K Neighbors Regressor	0.594	8199.393	213246328.556

3. MÔ HÌNH DỰ ĐOÁN DOANH THU

3.1. Quy trình phân tích và xây dựng mô hình



3.2. Mô hình đề xuất



3.3. Phân tích khám phá dữ liệu - EDA

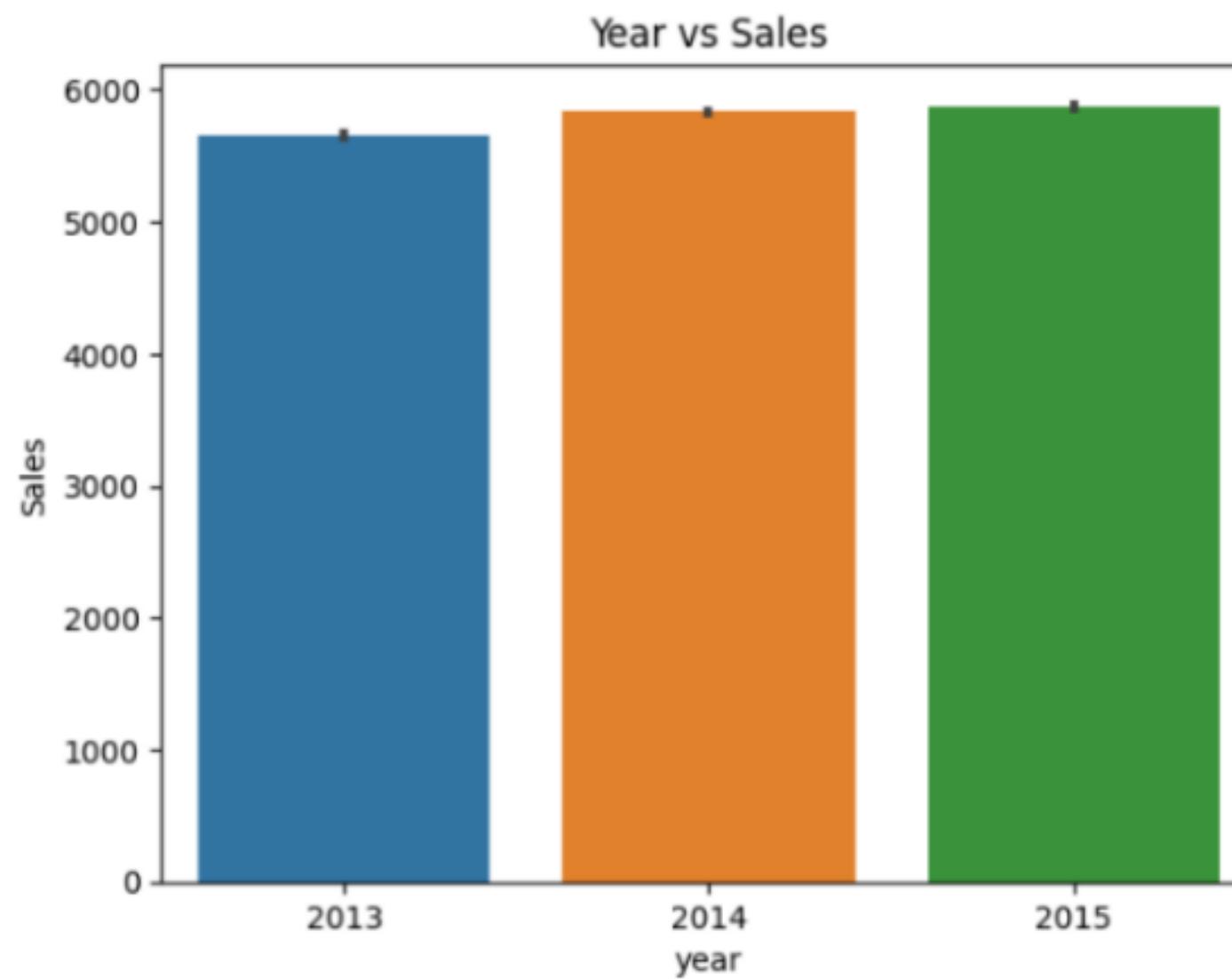
Tập dữ liệu ban đầu - Rossmann Store Sales

- 1.115 cửa hàng thuộc Rossmann
- Từ tháng 1 năm 2013 đến tháng 7 năm 2015
- 18 cột và 1.017.209 dòng

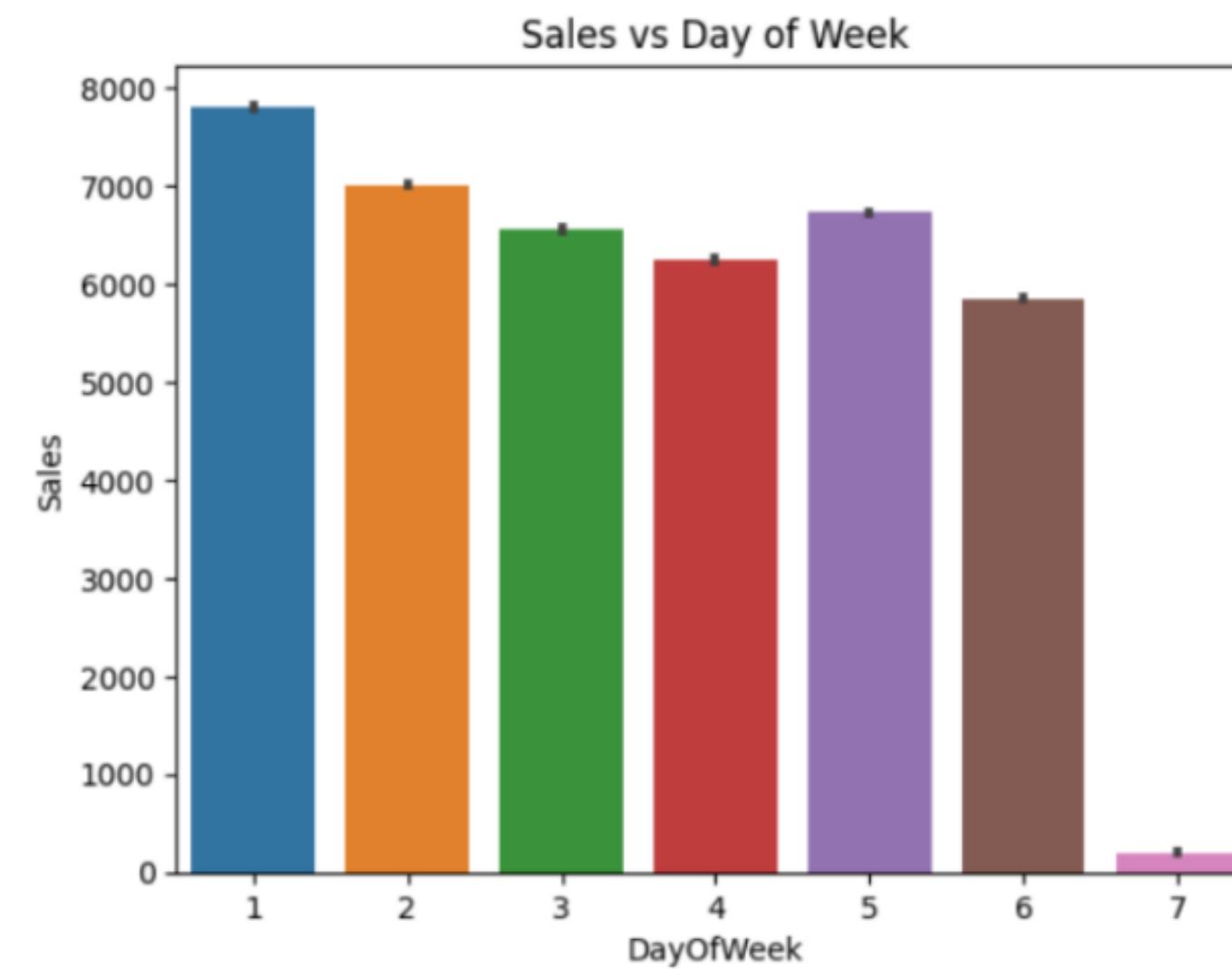
STT	Thuộc tính	Mô tả
1	Store	ID của mỗi cửa hàng
2	DayOfWeek	Ngày thứ bao nhiêu trong tuần (thứ 2 đến chủ nhật), thông thường các cửa hàng đóng cửa vào thứ bảy và chủ nhật
3	Date	Thời gian diễn ra hoạt động của cửa hàng
4	Sales	Doanh thu cho bất kỳ ngày nào (Giá trị chúng ta cần dự đoán)
5	Customers	Số lượng khách hàng vào một ngày
6	Open	Chỉ báo xem cửa hàng có mở hay không (0: đóng cửa, 1: mở cửa)
7	Promo	Cho biết cửa hàng có khuyến mãi vào ngày đó không (0: không có, 1: có)
8	StateHoliday	Cho biết ngày lễ của tiểu bang. Trừ một vài ngoại lệ, thông thường các cửa hàng đều đóng cửa vào các ngày lễ của tiểu bang (a: nghỉ lễ, b: lễ Phục sinh, c: Giáng sinh, 0: không)
9	SchoolHoliday	Cho biết liệu cửa hàng có bị ảnh hưởng bởi việc đóng cửa, nghỉ lễ của trường công lập không (0: không, 1: có)
10	StoreType	Chỉ ra 4 mô hình cửa hàng khác nhau gồm a, b, c, d. Các loại cửa hàng khác nhau thì bán khác sản phẩm
11	Assortment	Mô tả cấp độ phân loại (a: cơ bản, b: bổ sung, c: mở rộng). Cho biết sự đa dạng trong các mặt hàng của cửa hàng
12	CompetitionDistance	Khoảng cách đến đối thủ cạnh tranh gần nhất (tính bằng mét)
13	CompetitionOpenSinceMonth	Tháng mà đối thủ cạnh tranh gần nhất được mở
14	CompetitionOpenSinceYear	Năm mà đối thủ cạnh tranh gần nhất được mở
15	Promo2	Chương trình khuyến mãi liên tục, liên tiếp cho một số cửa hàng (1: cửa hàng đang tiến hành khuyến mãi, 0: cửa hàng không tiến hành khuyến mãi)
16	Promo2SinceWeek	Tuần khi cửa hàng bắt đầu tham gia Promo2
17	Promo2SinceYear	Năm khi cửa hàng bắt đầu tham gia Promo2
18	Promointerval	Cho biết Promo2 chạy trong những tháng nhất định nào trong năm (khoảng thời gian liên tiếp promo2 được bắt đầu lại)

3.3. Phân tích khám phá dữ liệu - EDA

- Trực quan hóa dữ liệu



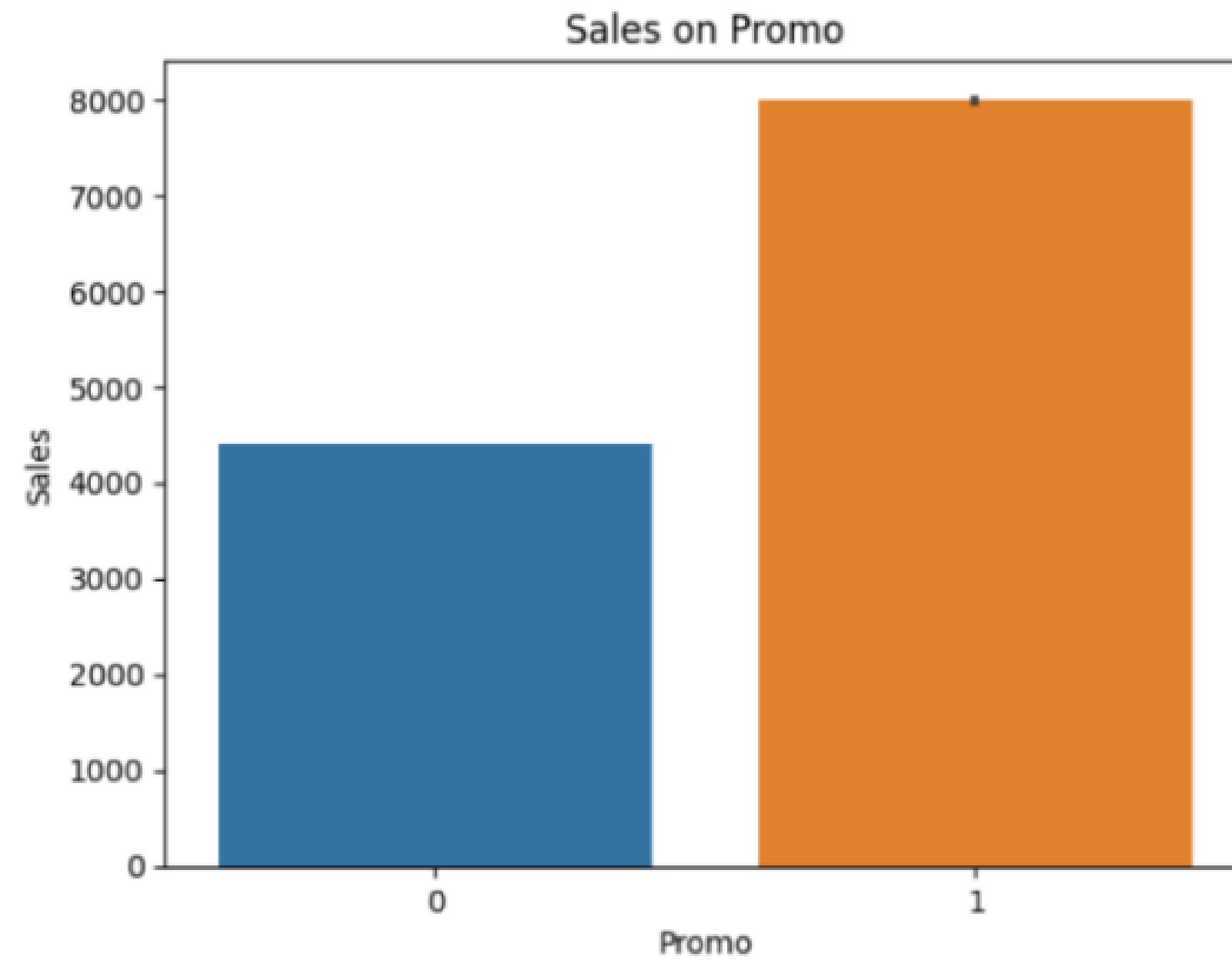
Biểu đồ thể hiện doanh thu theo năm



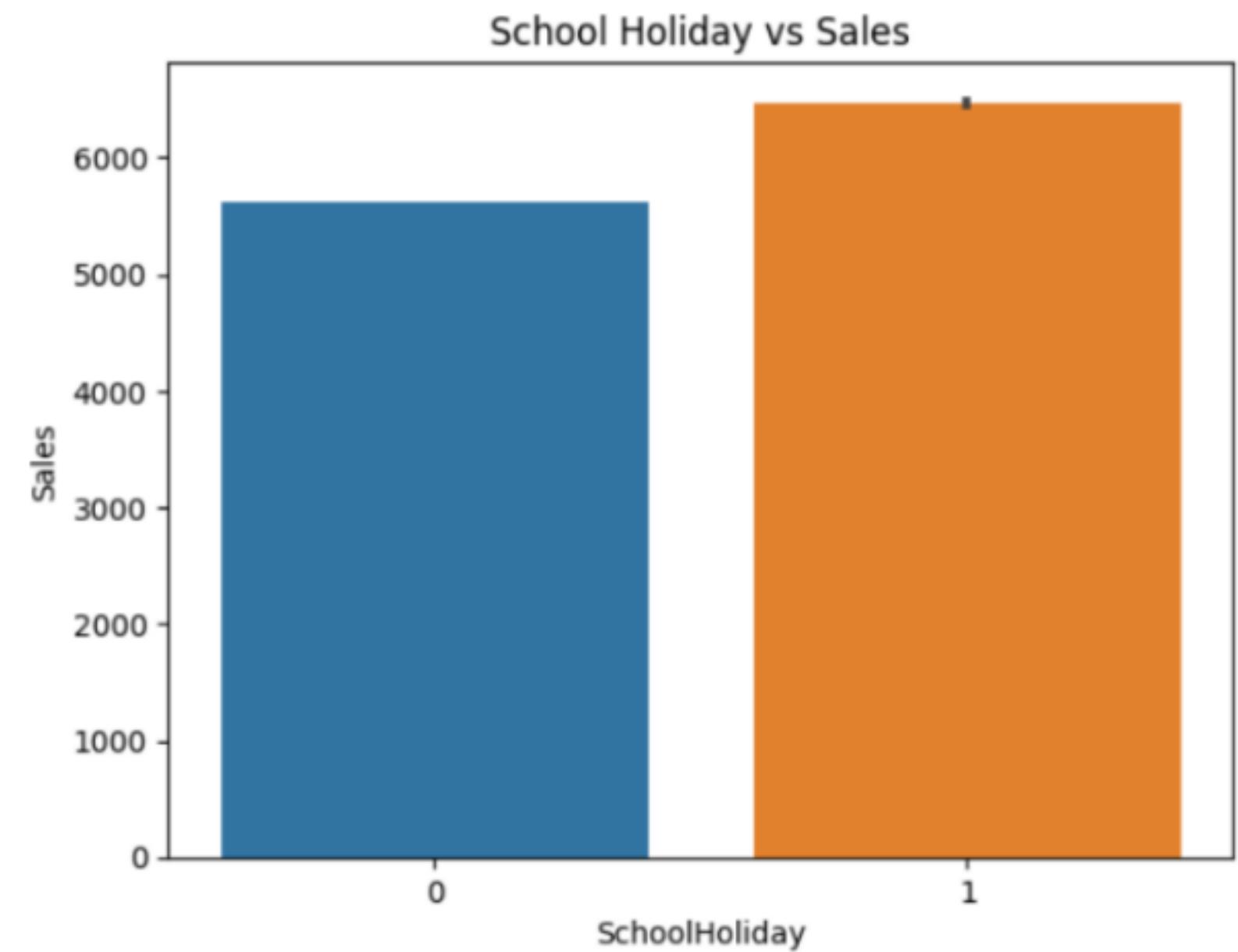
Biểu đồ thể hiện doanh thu theo ngày trong tuần

3.3. Phân tích khám phá dữ liệu - EDA

- Trực quan hóa dữ liệu



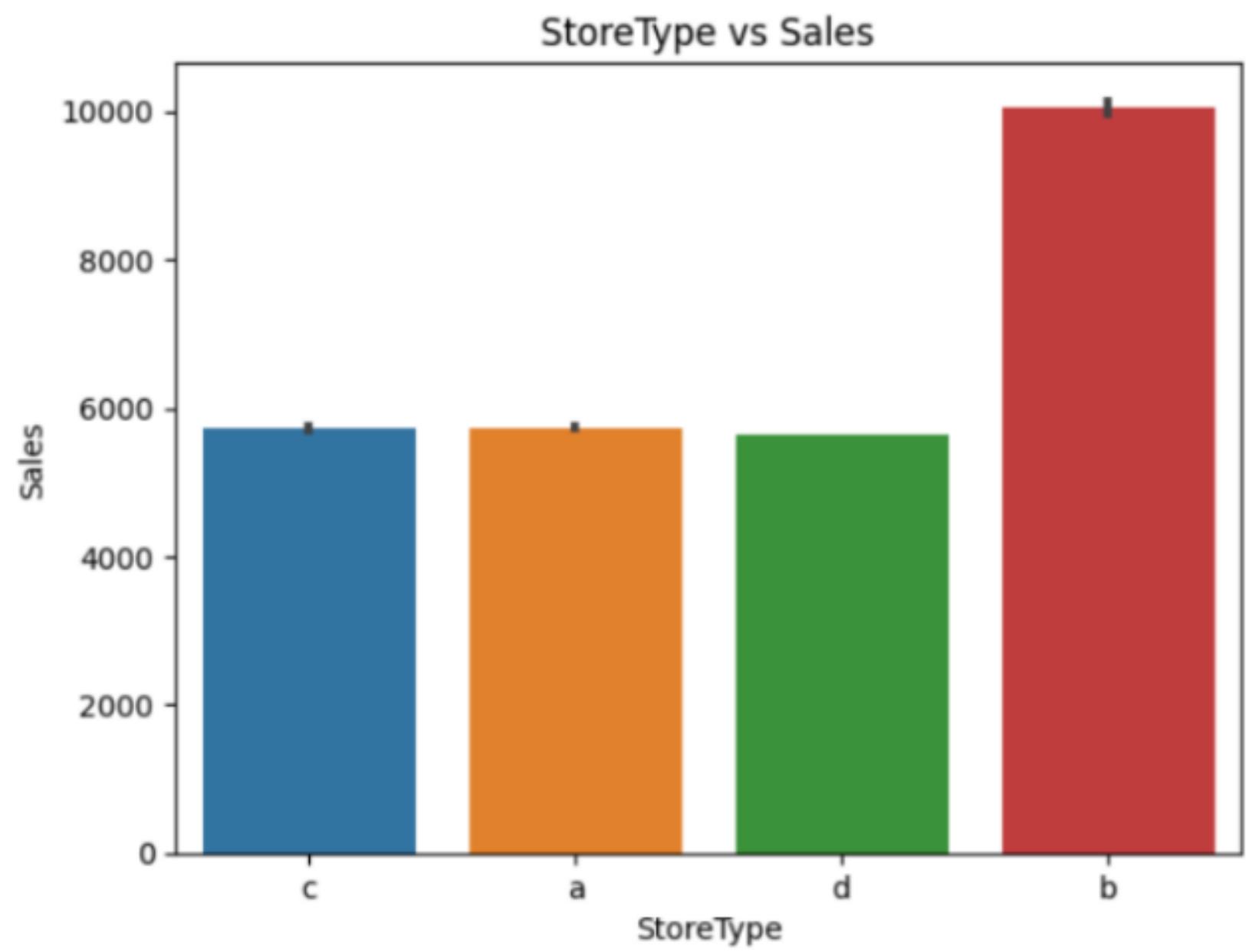
Biểu đồ thể hiện doanh thu theo khuyến mãi



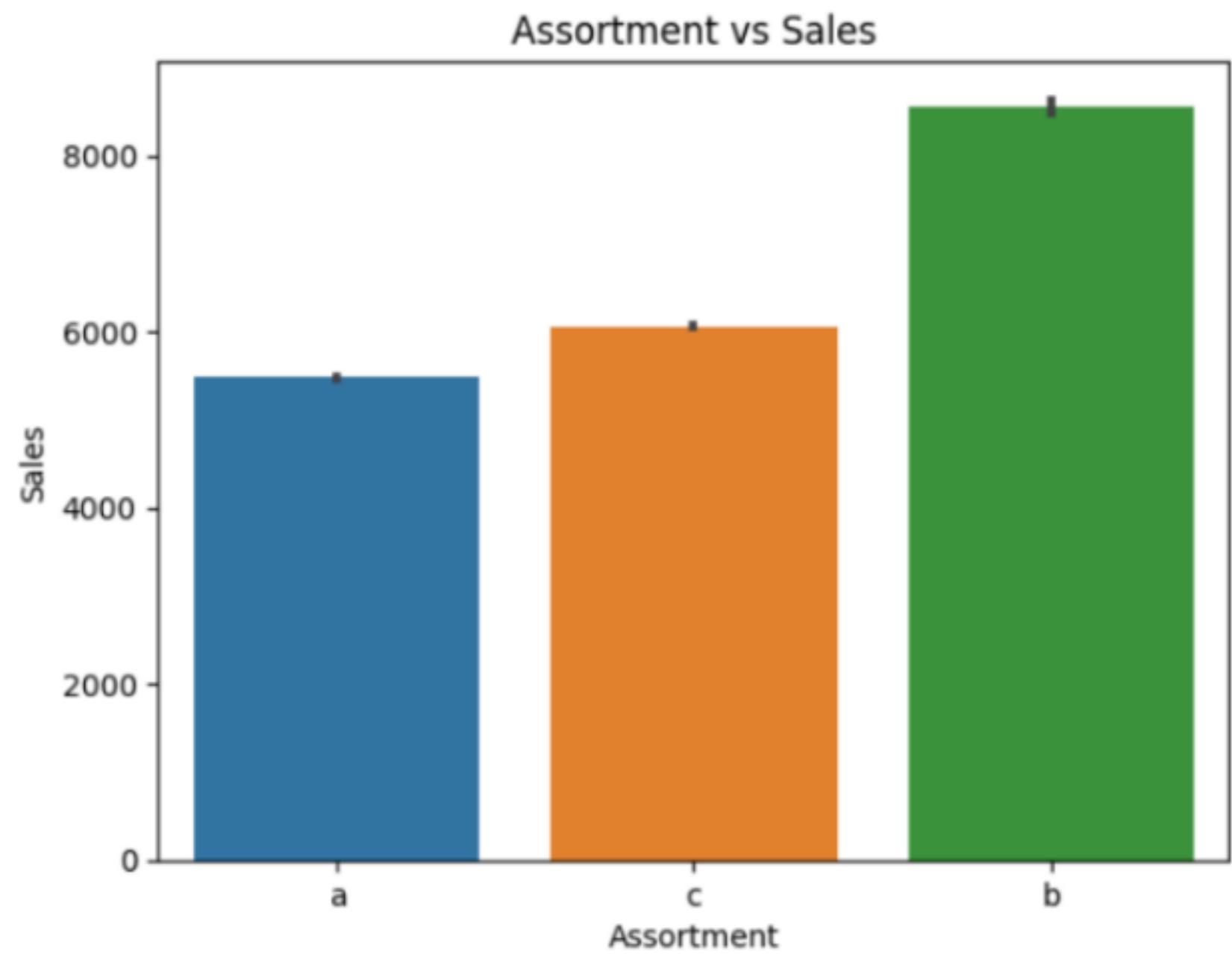
Biểu đồ thể hiện doanh thu theo ngày nghỉ ở trường

3.3. Phân tích khám phá dữ liệu - EDA

- Trực quan hóa dữ liệu



Biểu đồ thể hiện doanh thu theo loại cửa hàng



Biểu đồ thể hiện doanh thu theo Assortment

3.3. Phân tích khám phá dữ liệu - EDA

- Xử lý giá trị null

Kiểm tra giá trị null ở file store và file train.

```
store_data.isnull().sum()
```

Store	0
StoreType	0
Assortment	0
CompetitionDistance	3
CompetitionOpenSinceMonth	354
CompetitionOpenSinceYear	354
Promo2	0
Promo2SinceWeek	544
Promo2SinceYear	544
PromoInterval	544
dtype: int64	

```
train_data.isnull().sum()
```

Store	0
DayOfWeek	0
Date	0
Sales	0
Customers	0
Open	0
Promo	0
StateHoliday	0
SchoolHoliday	0
dtype: int64	

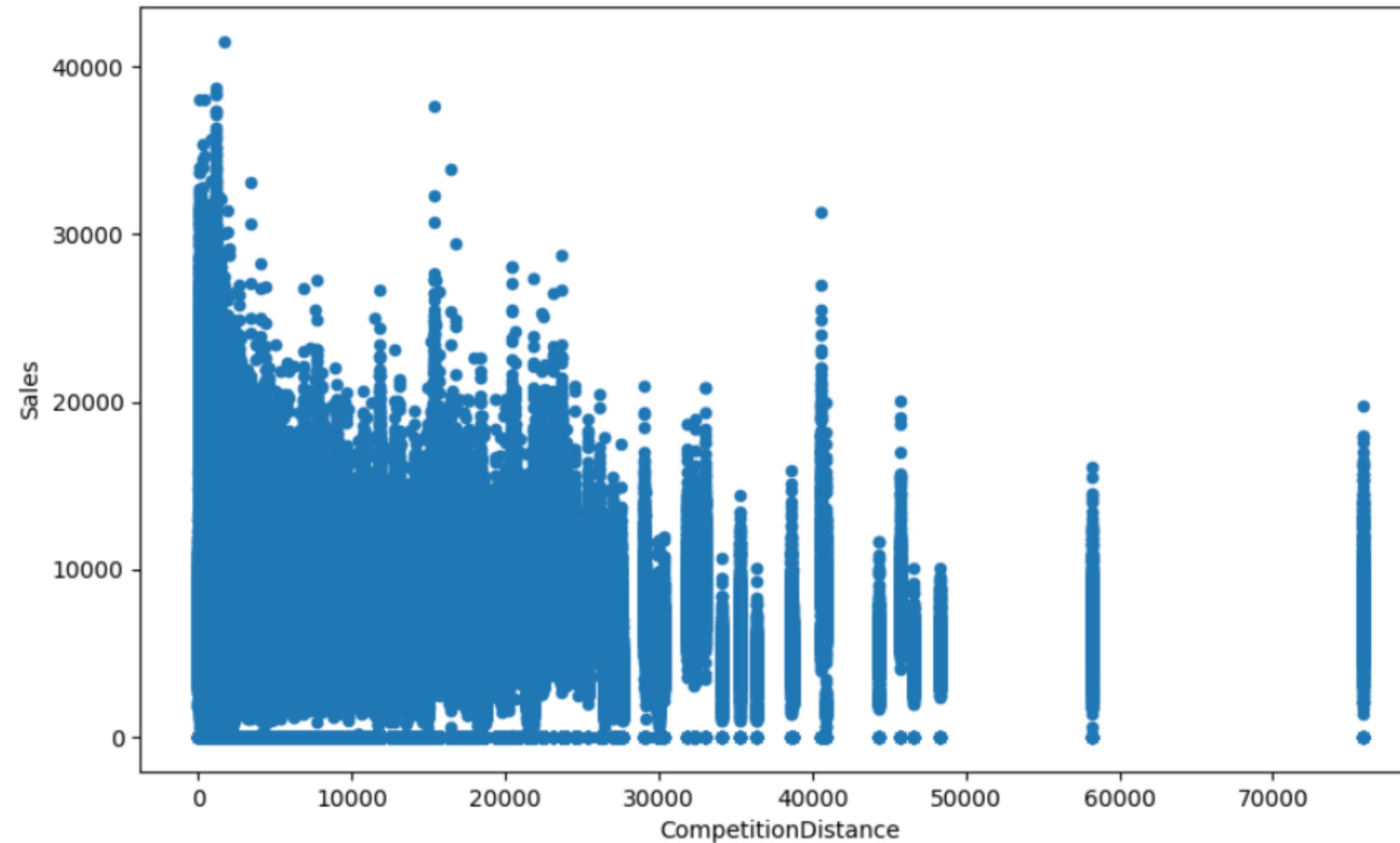
3.3. Phân tích khám phá dữ liệu - EDA

- Xử lý giá trị null

Tiến hành fill null

- Giá trị null ở những cột Promo2SinceWeek, Promo2SinceYear, Promointerval được fill null bằng 0.
- Giá trị null ở cột CompetitionDistance được fill null bằng giá trị trung bình.
- Giá trị null ở cột CompetitionOpenSinceMonth, CompetitionOpenSinceYear được điền bằng giá trị tháng, năm hiện tại

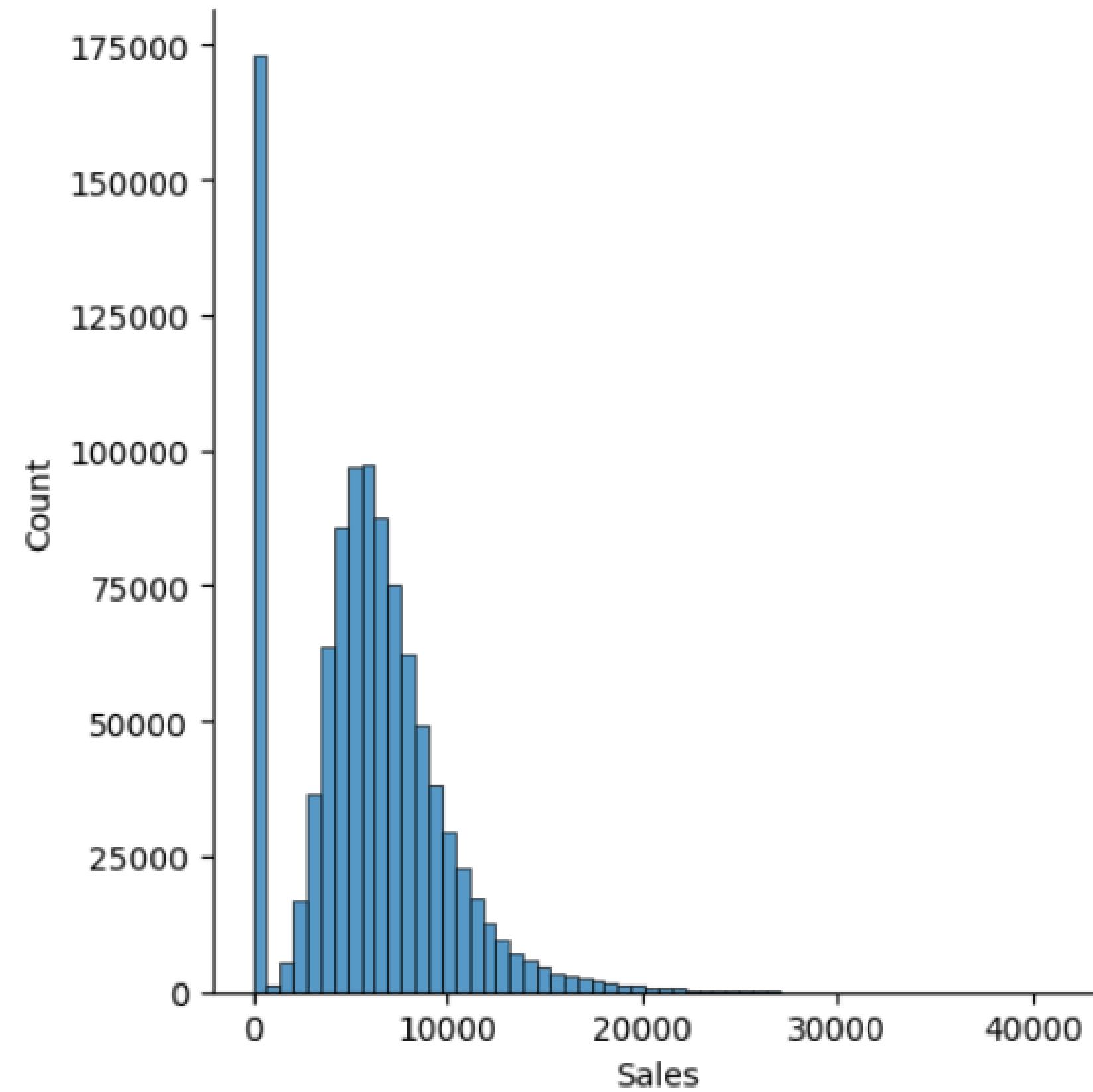
3.3. Phân tích khám phá dữ liệu - EDA



Biểu đồ thể hiện doanh thu theo CompetitionDistance

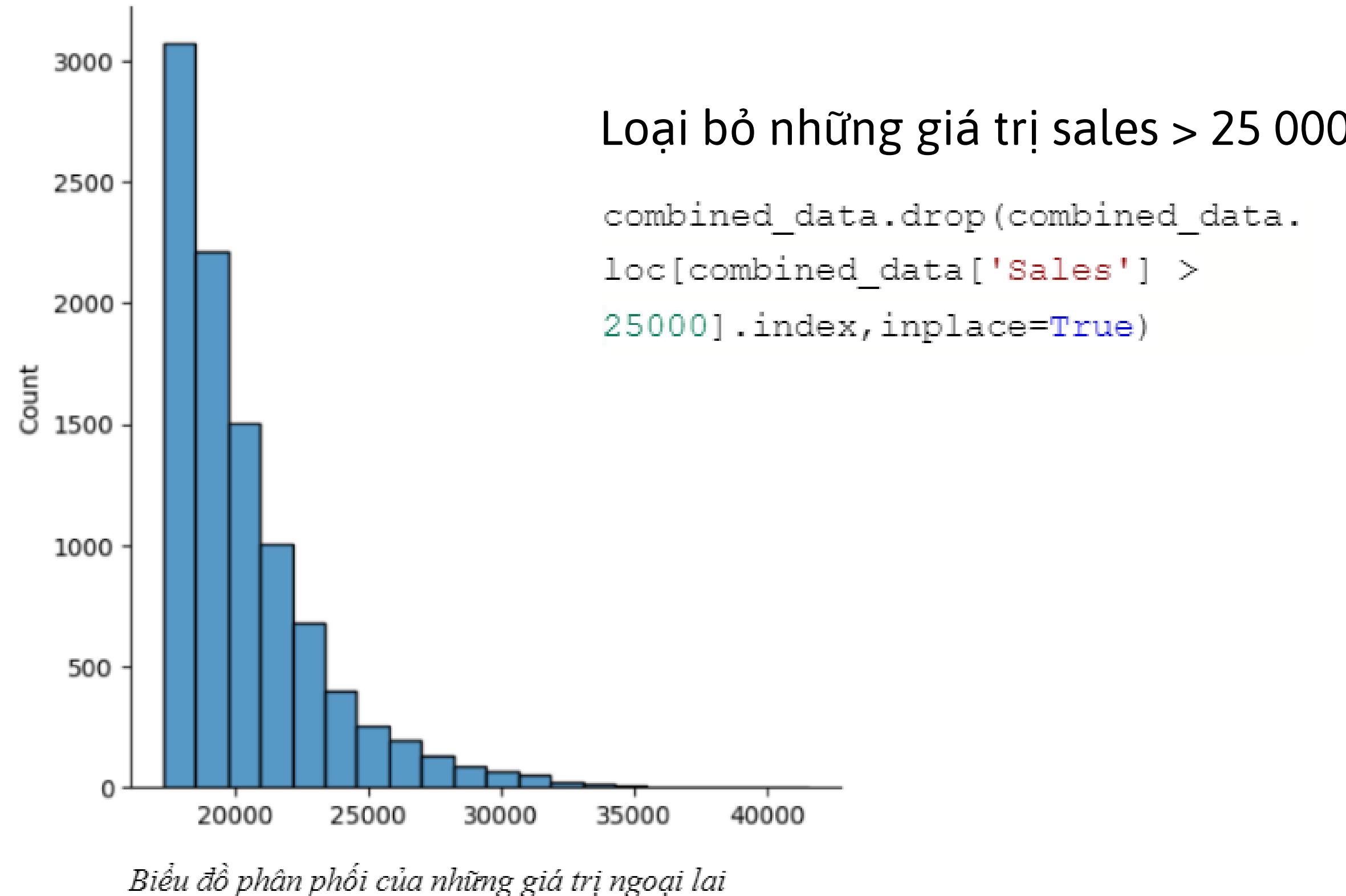
3.3. Phân tích khám phá dữ liệu - EDA

- Xử lý giá trị ngoại lai



3.3. Phân tích khám phá dữ liệu - EDA

- Xử lý giá trị ngoại lai



3.4. Tiền xử lý dữ liệu

- Tiến hành tách cột ‘Date’ thành 2 cột ‘Year’ và ‘Month’, đồng thời xóa cột ‘Date’.

Store	int64	Sales	int64
StoreType	object	Customers	int64
Assortment	object	Open	int64
CompetitionDistance	float64	Promo	int64
CompetitionOpenSinceMonth	float64	StateHoliday	object
CompetitionOpenSinceYear	float64	SchoolHoliday	int64
Promo2	int64	Year	int64
Promo2SinceWeek	float64	Month	int64
Promo2SinceYear	float64	dtype: object	int64
PromoInterval	object		
DayOfWeek	int64		

3.4. Tiền xử lý dữ liệu

- Chuyển đổi kiểu dữ liệu phân loại sang số nhị phân

Có 4 cột có kiểu dữ liệu object là: StoreType, Assortment, Promolnterval và StateHoliday.

Label encoder: Promolnterval, StateHoliday

Dummy encoding: StoreType, Assortment

- StoreType : 'StoreType_a', 'StoreType_b', 'StoreType_c', 'StoreType_d'.
- Assortment: 'Assortment_a', 'Assortment_b', 'Assortment_c'.

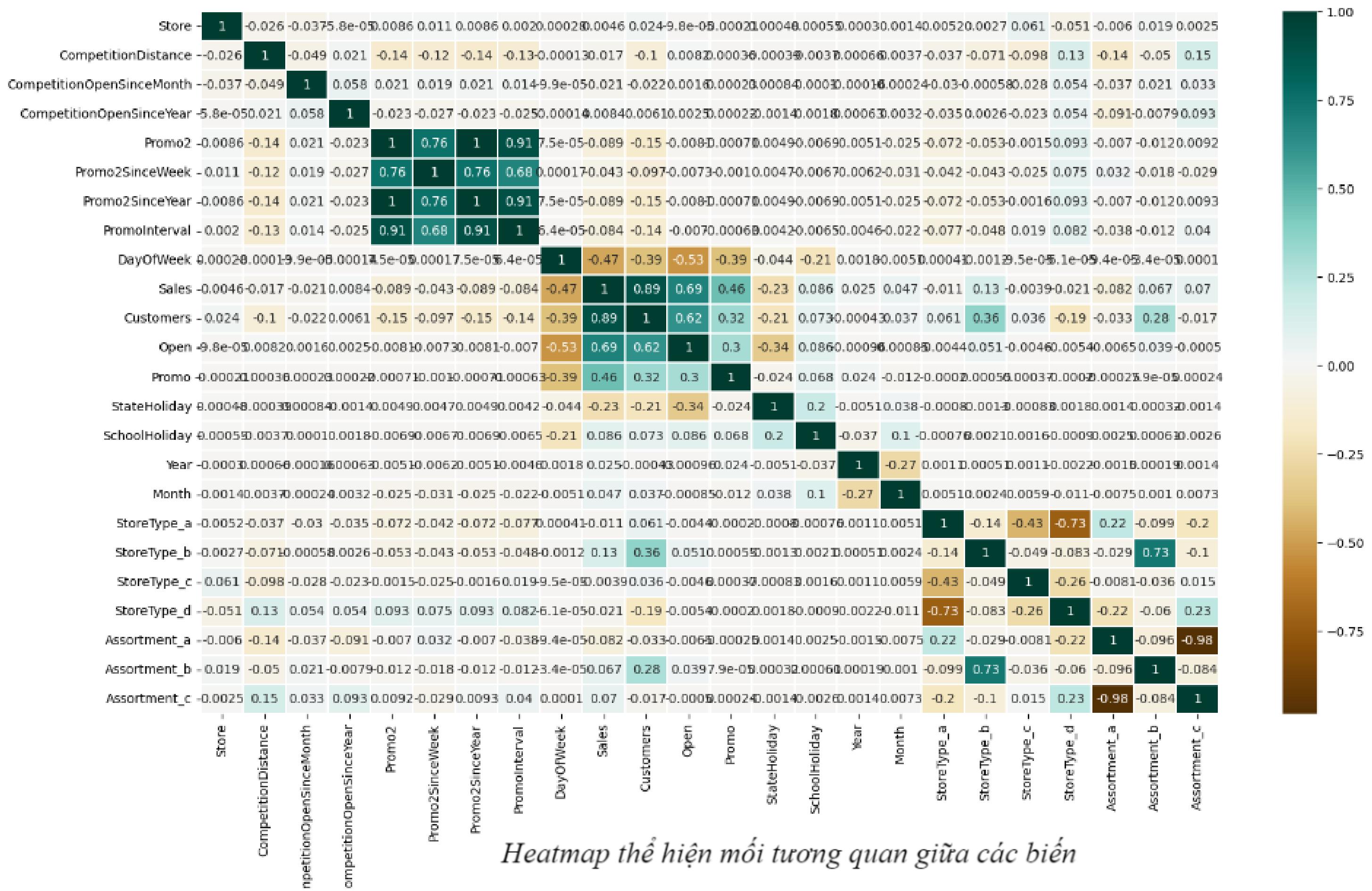
3.4. Tiền xử lý dữ liệu

- Chuyển đổi kiểu dữ liệu phân loại sang số nhị phân

SchoolHoliday	Year	Month	StoreType_a	StoreType_b	StoreType_c	StoreType_d	Assortment_a	Assortment_b	Assortment_c
1	2015	7	0	0	1	0	1	0	0
1	2015	7	0	0	1	0	1	0	0
1	2015	7	0	0	1	0	1	0	0
1	2015	7	0	0	1	0	1	0	0
1	2015	7	0	0	1	0	1	0	0
...
1	2013	1	0	0	0	1	0	0	1
1	2013	1	0	0	0	1	0	0	1
1	2013	1	0	0	0	1	0	0	1
1	2013	1	0	0	0	1	0	0	1
1	2013	1	0	0	0	1	0	0	1

Bảng dữ liệu sau khi dùng label encoder và dummy encoding

3.4. Tiền xử lý dữ liệu



3.4. Tiền xử lý dữ liệu

- Xây dựng model Regression

Chia biến, chia tập train - test và loại bỏ biến tương quan mạnh: Customer, Open

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import
r2_score,mean_squared_error,mean_absolute_error
import math

X_train, X_test, y_train, y_test_open =
train_test_split(combined_data_open.drop(['Sales','Customers','Open'],a
xis=1),
combined_data_open['Sales'], test_size=0.2, random_state=23)
```

3.4. Tiền xử lý dữ liệu

- Xây dựng model Regression

Dùng StandardScaler chia tỷ lệ cho tập huấn luyện và kiểm tra các biến độc lập để giảm kích thước xuống các giá trị nhỏ hơn phù hợp với việc chạy mô hình.

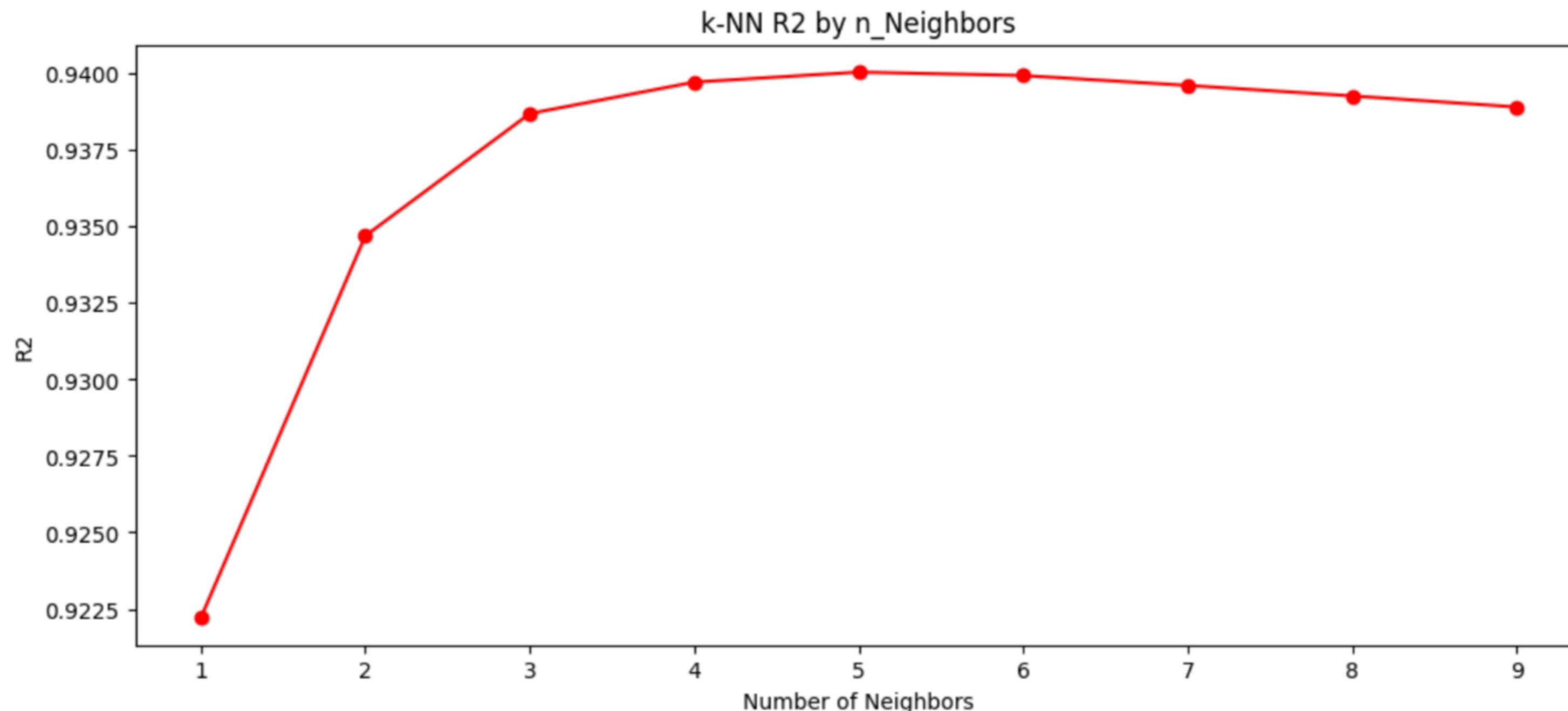
```
array([[-1.02784156, -1.46288917, -0.1750752 , ..., -1.05424542,
       -0.09920987,  1.07508071],
      [-1.25787501,  1.43870354, -0.24599842, ..., -1.05424542,
       -0.09920987,  1.07508071],
      [ 0.21869109, -0.88257063, -0.24937571, ...,  0.94854574,
       -0.09920987, -0.93016273],
      ...,
      [-0.57399176, -0.88257063,  0.24033218, ...,  0.94854574,
       -0.09920987, -0.93016273],
      [-1.62468512, -1.46288917,  1.31768953, ..., -1.05424542,
       -0.09920987,  1.07508071],
      [-0.35639255,  0.858385 ,  2.17889996, ...,  0.94854574,
       -0.09920987, -0.93016273]])
```

Dữ liệu X_train sau khi Standard Scaler

4. KẾT QUẢ THỰC NGHIỆM VÀ ĐỀ XUẤT

4.1.1 Mô hình K-Nearest Neighbor

- Tìm hệ số K tối ưu



4. KẾT QUẢ THỰC NGHIỆM VÀ ĐỀ XUẤT

4.1.1 Mô hình K-Nearest Neighbor

- Tiến hành đánh trọng số

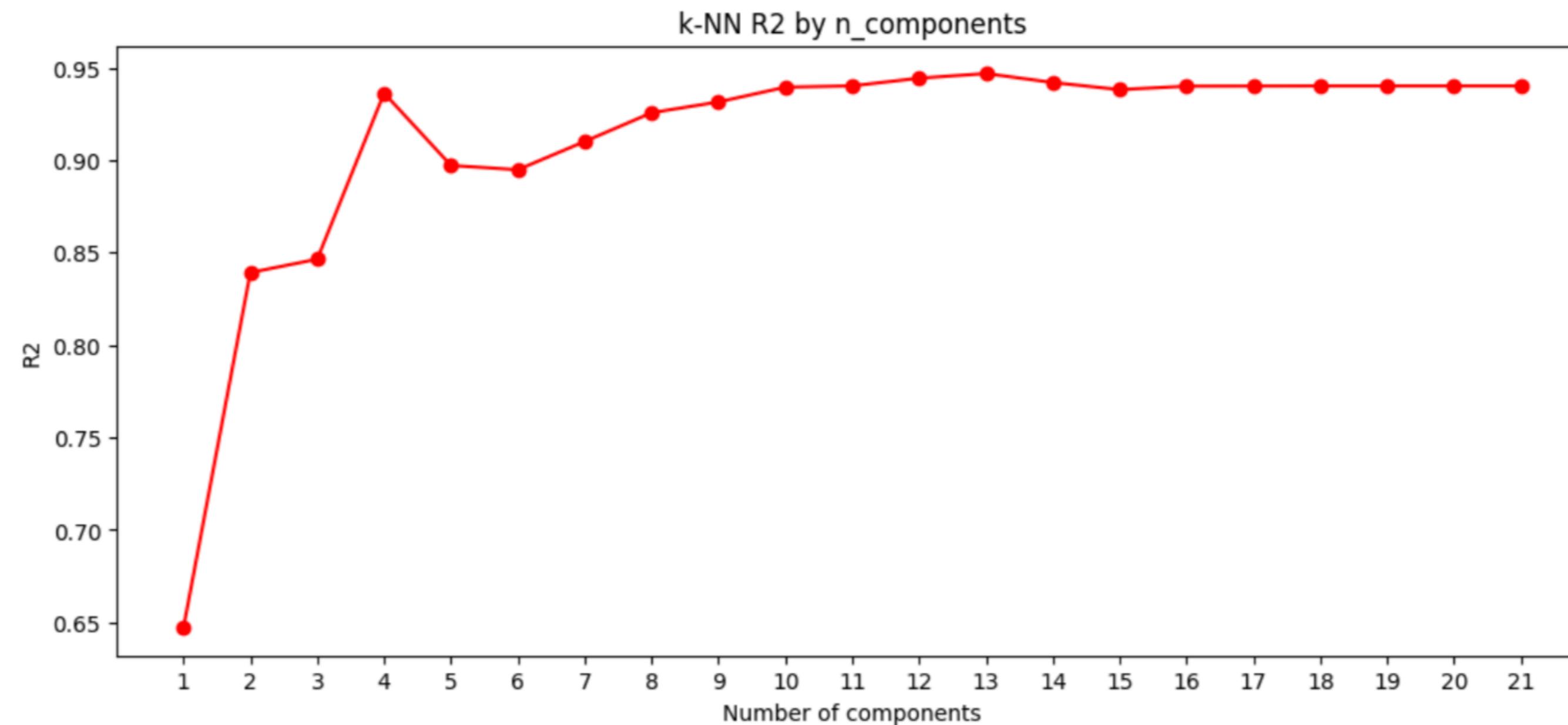
```
KNN_original = KNeighborsRegressor(n_neighbors=5,  
algorithm='auto', leaf_size=30, p=2, metric='minkowski',  
weights='distance')
```

	Mô hình chưa điều chỉnh	Mô hình đánh trọng số
R2	0.931	0.940
MAE	487.2	460.9
MSE	1149970.0	994853.8
RMSE	1072.4	997.4
RMSPE	0.23176	0.23253

4. KẾT QUẢ THỰC NGHIỆM VÀ ĐỀ XUẤT

4.1.1 Mô hình K-Nearest Neighbor

- Thực hiện thay đổi chiều dữ liệu, R2 đạt cao nhất tại n_components =13



4. KẾT QUẢ THỰC NGHIỆM VÀ ĐỀ XUẤT

4.1.1 Mô hình K-Nearest Neighbor

- k = 5
- algorithm='auto'
- leaf_size=30
- p=2, metric='minkowski'
- weights='distance'
- n_components =13

	Mô hình chưa điều chỉnh	Mô hình đã điều chỉnh
R2	0.931	0.947
MAE	487.2	441.7
MSE	1149970.0	883274.0
RMSE	1072.4	939.8
RMSPE	0.23176	0.22318

4. KẾT QUẢ THỰC NGHIỆM VÀ ĐỀ XUẤT

4.1.2 Mô hình Linear Regression

- Sử dụng thuật toán Gradient Descent cho mô hình Linear Regression

Tính toán các trọng số của mô hình với learning rate = 0.556 và epochs = 900000 (số vòng lặp tối đa)

```
Cost is: 8593986.492160352
Cost is: 3616048.7993090674
Cost is: 3615981.8394388487
Cost is: 3615918.91830362
Cost is: 3615859.7923040404
Cost is: 3615804.2325336444
Cost is: 3615752.0238926336
Cost is: 3615702.9642551155
Cost is: 3615656.863686576
Cost is: 3615613.5437085396
```

Hình: Kết quả giảm dần của hàm chi phí khi chạy thuật toán Gradient Descent

4.1.2 Mô hình Linear Regression

- Sử dụng thuật toán Gradient Descent cho mô hình Linear Regression

Hệ số hồi quy:

```
array([ 1.47002157e+01, -1.74732672e+02, -1.07032819e+02, -5.36671200e+00,
       1.14284572e+04,  3.09294501e+02, -1.18801866e+04, -1.60310286e+02,
      -2.53568680e+02,  1.06667811e+03, -5.44722032e+00,  2.92811234e+01,
       1.52342285e+02,  2.47851618e+02, -3.07150989e+01,  6.19889673e+02,
      -4.62800017e+01, -1.12572277e+02, -1.91057595e+02, -2.81769640e+02,
       2.46691340e+02])
```

Hệ số chặn:

6936.608734240404

4. KẾT QUẢ THỰC NGHIỆM VÀ ĐỀ XUẤT

4.1.2 Mô hình Linear Regression

- Kết quả của mô hình Linear Regression sau khi chạy Gradient Descent

R2	0.784
MAE	984.3
MSE	3580237.9
RMSE	1892.1
RMSPE	0.34174

4.1.3 Mô hình Decision Tree

- Cắt tỉa Decision Tree bằng cách đặt ra các giới hạn cho cây
- GridSearchCV

```
'max_depth': [3000, 4000, 5000, 6000, 7000],  
'min_samples_leaf': [1, 2, 3, 4, 5, 6, 7, 10, 15, 20],  
'max_leaf_nodes' : [10000, 11000, 12000, 13000, 14000, 15000]
```

- Kết quả

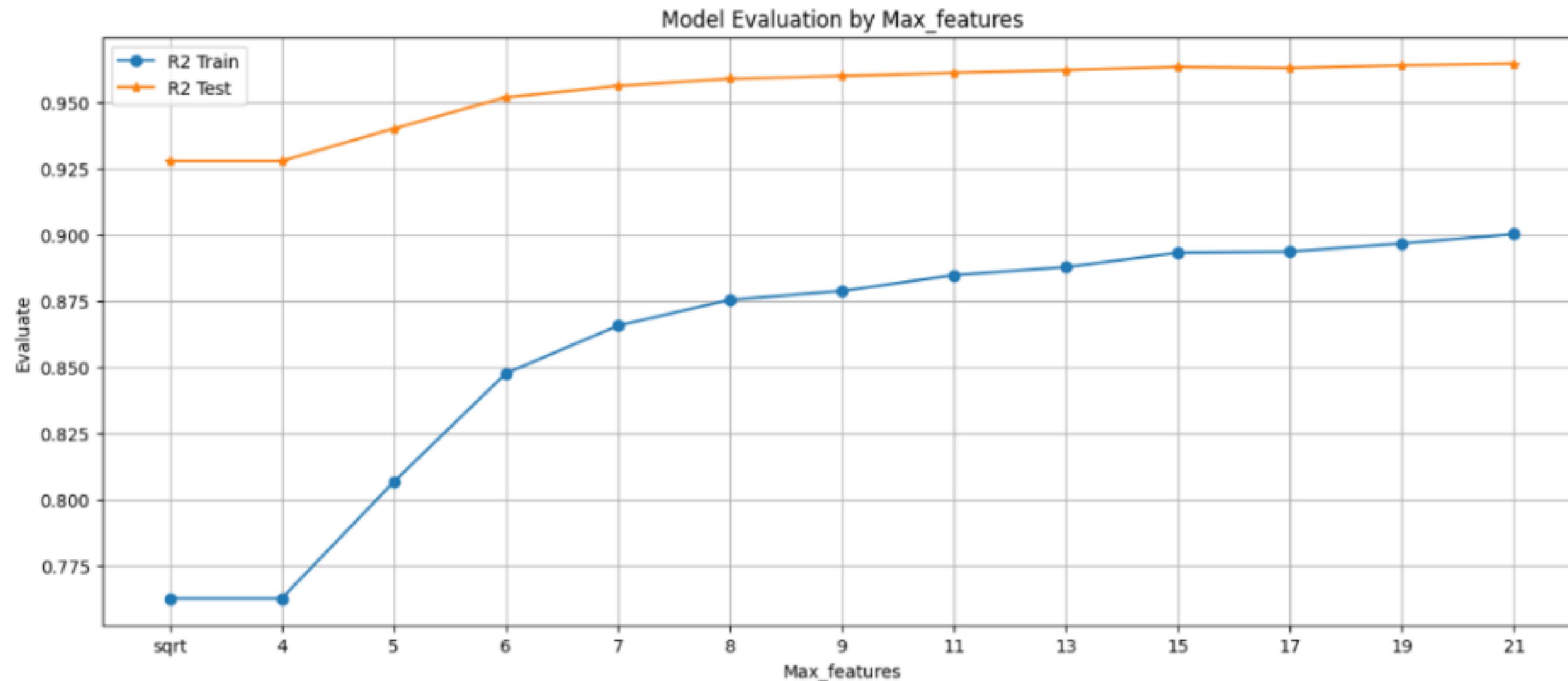
max_depth = 3000

min_samples_leaf = 5

max_leaf_nodes = 15000

4.1.3 Mô hình Decision Tree

- Tìm max_features



Hình: Kiểm tra max_features và hiệu quả của mô hình Decision Tree

4.1.3 Mô hình Decision Tree

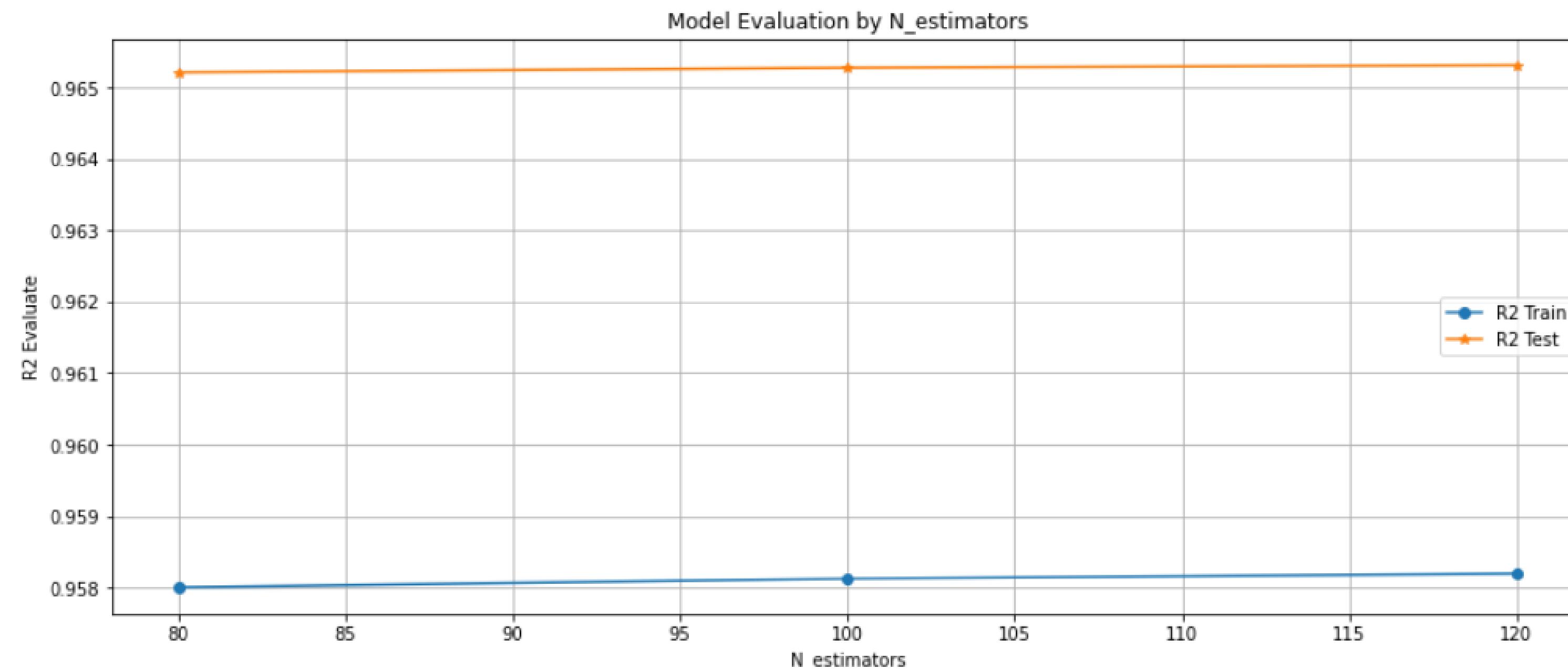
- Mô hình đã điều chỉnh

- max_depth = 3000
- min_samples_leaf = 5
- max_leaf_nodes = 15000
- max_feature = 15

	Mô hình chưa điều chỉnh	Mô hình đã điều chỉnh
R2 in train	0.964	0.893
R2 in test	0.9515	0.9632
MAE	419.9	364.0
MSE	803718.4	611153.6
RMSE	896.5	781.8
RMSPE	0.18057	0.11078

4.1.4 Mô hình Random Forest

- Điều chỉnh `n_estimators` với các thông số ngẫu nhiên tương ứng là 80, 100 và 120



4.1.4 Mô hình Random Forest

- Dùng RandomizedSearchCV điều chỉnh: max_depth, min_samples_split, max_leaf_nodes

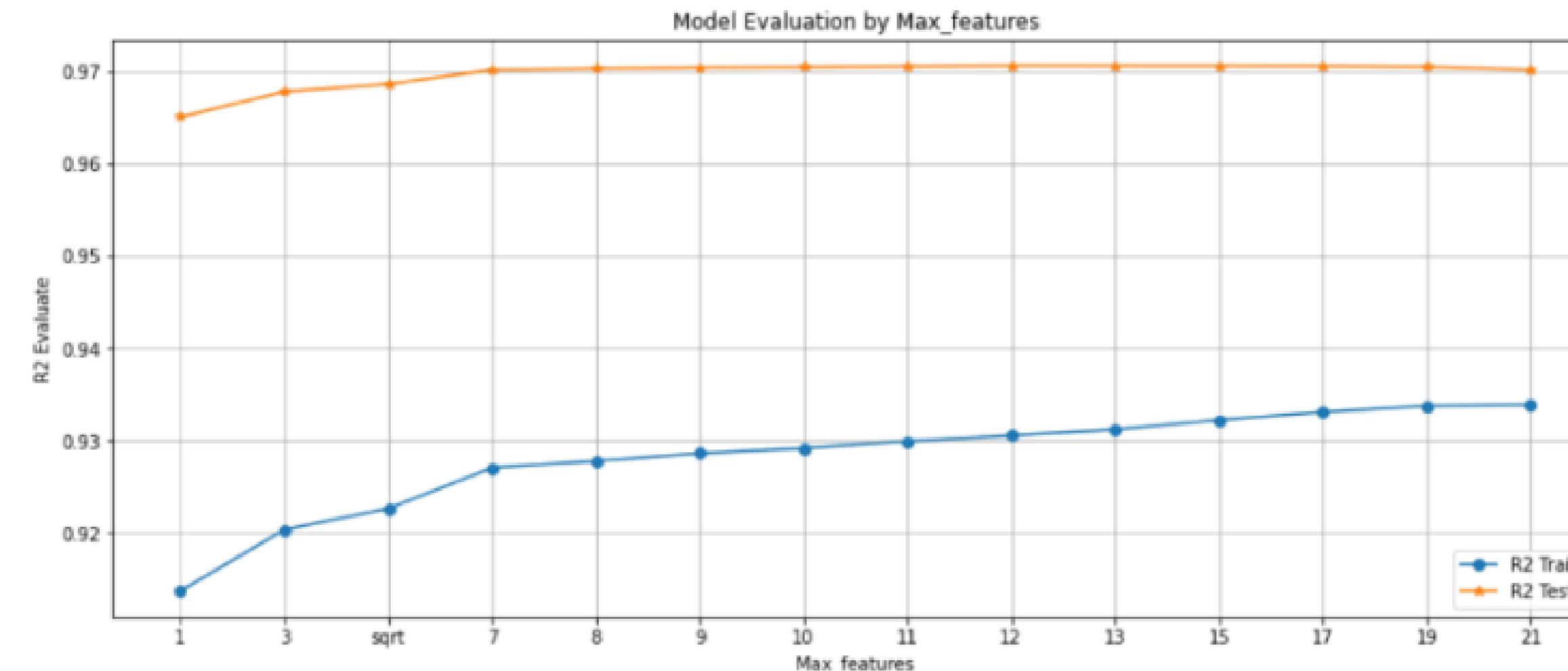
Kết quả chạy thu được là:

- Best estimator: RandomForestRegressor(max_depth=4000, min_samples_split=10, n_estimators=120, random_state=42)
- Best parameter: {'min_samples_split': 10, 'max_leaf_nodes': None, 'max_depth': 4000}
- R2 score: 0.9701706801461882

4.1.4 Mô hình Random Forest

- Điều chỉnh max_features để tìm ra giá trị max_features phù hợp của mô hình

```
features = [1, 3, 'sqrt', 7, 8, 9, 10, 11, 12, 13, 15, 17, 19, 21]
for i in features:
    regressor = RandomForestRegressor(max_depth=4000,
random_state=42, max_leaf_nodes=None, min_samples_split=10,
n_estimators=120, max_features=i)
```



Hình: Kiểm tra max_features trên mô hình Random Forest

4.1.4 Mô hình Random Forest

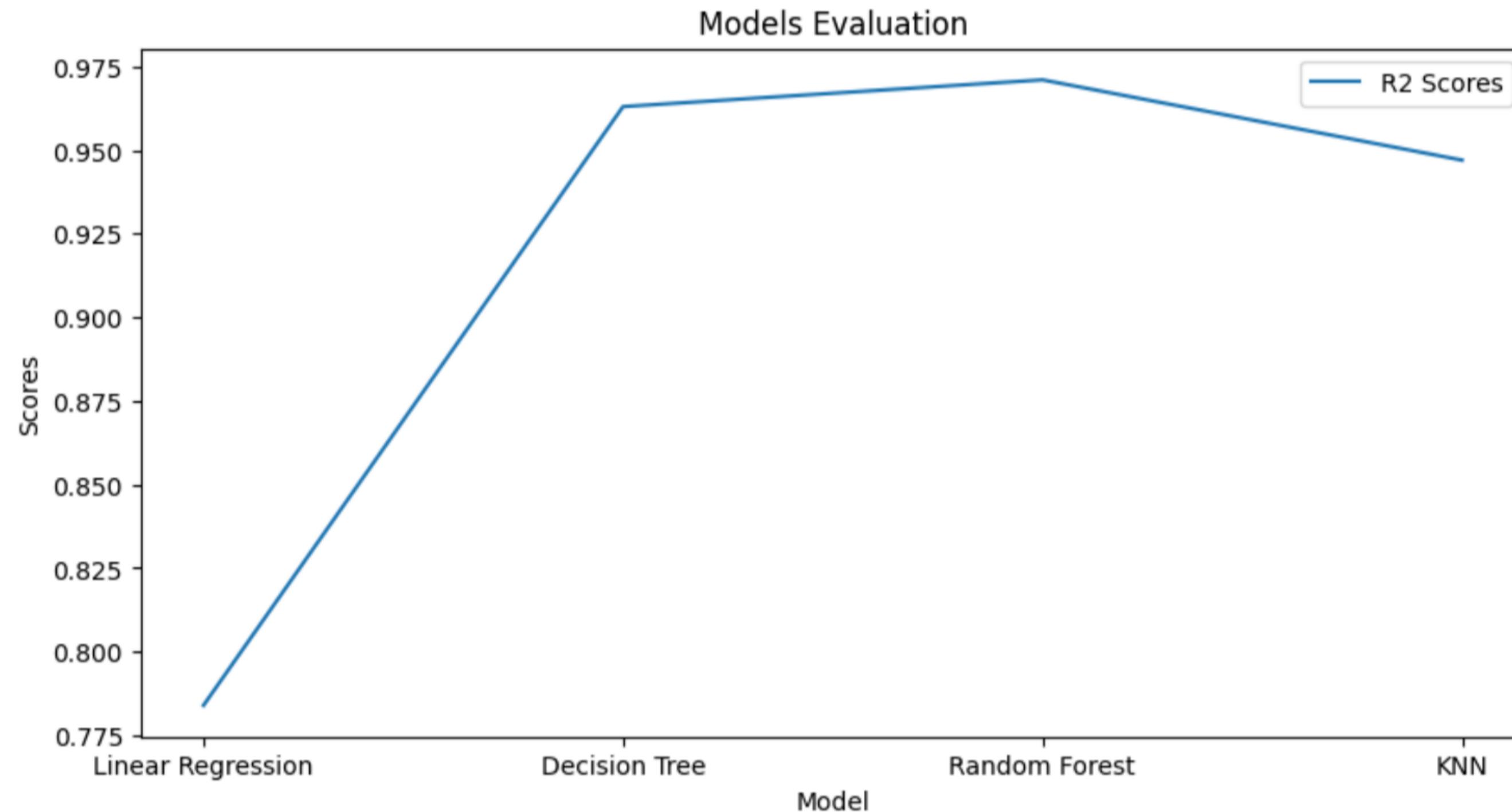
- Mô hình đã điều chỉnh

- n_estimators=120
- max_depth=4000
- min_samples_split=120
- max_leaf_nodes = None
- max_features: 12

	Mô hình chưa điều chỉnh	Mô hình đã điều chỉnh
R2 in train	0.9581	0.9305
R2 in test	0.9653	0.9706
MAE	356.7	325.1
MSE	576260.7	487562.1
RMSE	759.1	698.3
RMSPE	0.12554	0.10782

4. KẾT QUẢ THỰC NGHIỆM VÀ ĐỀ XUẤT

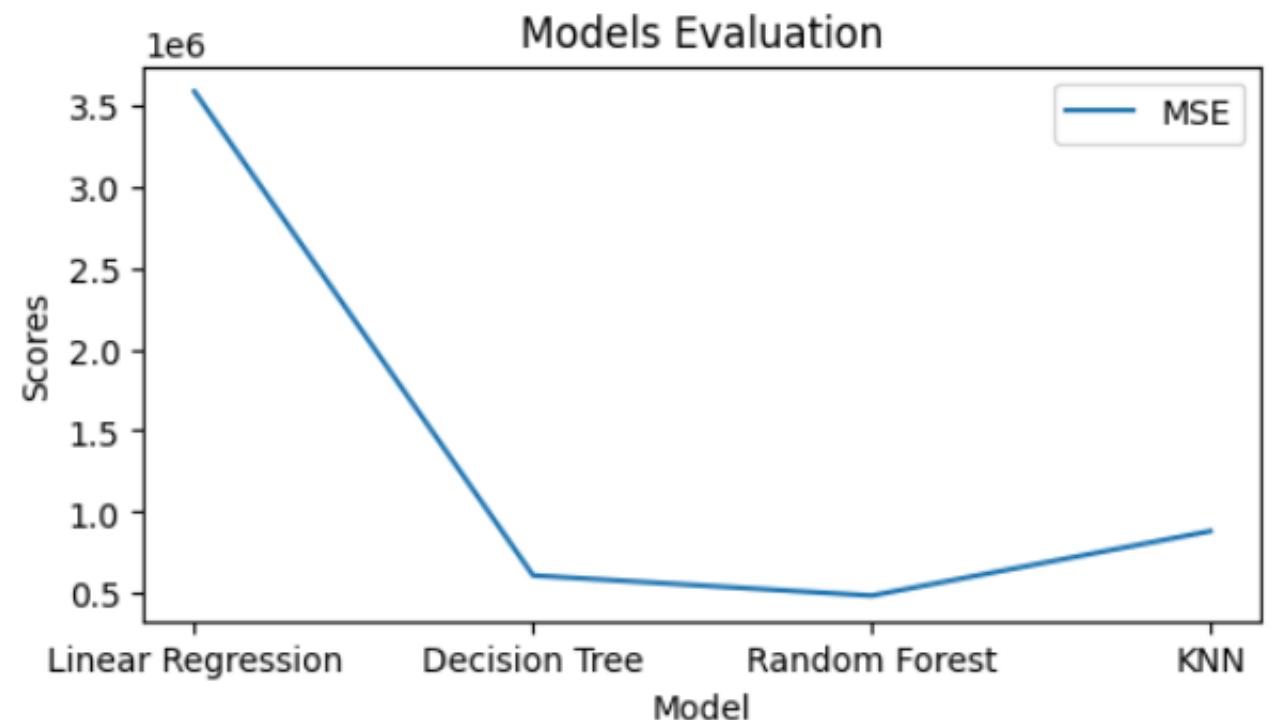
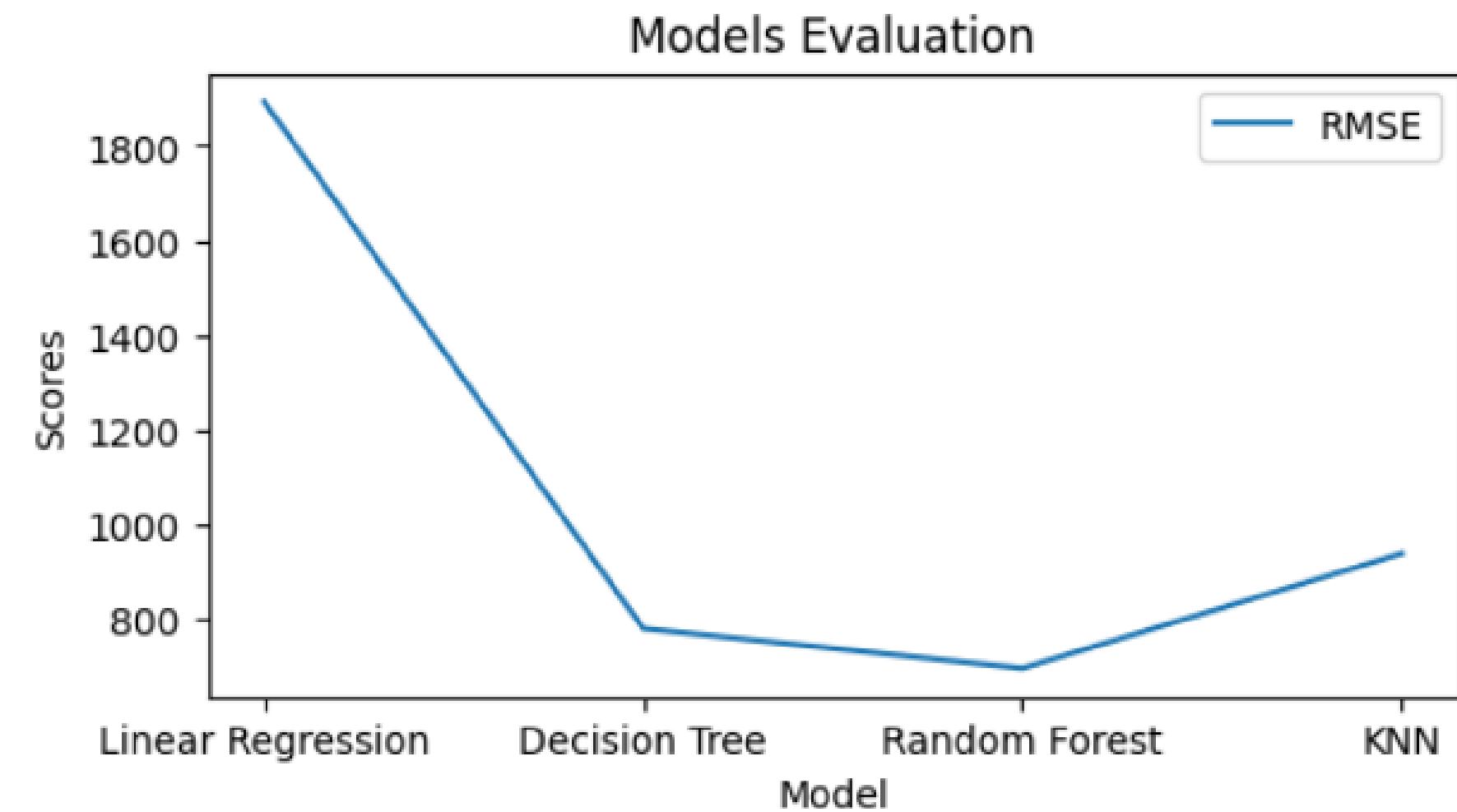
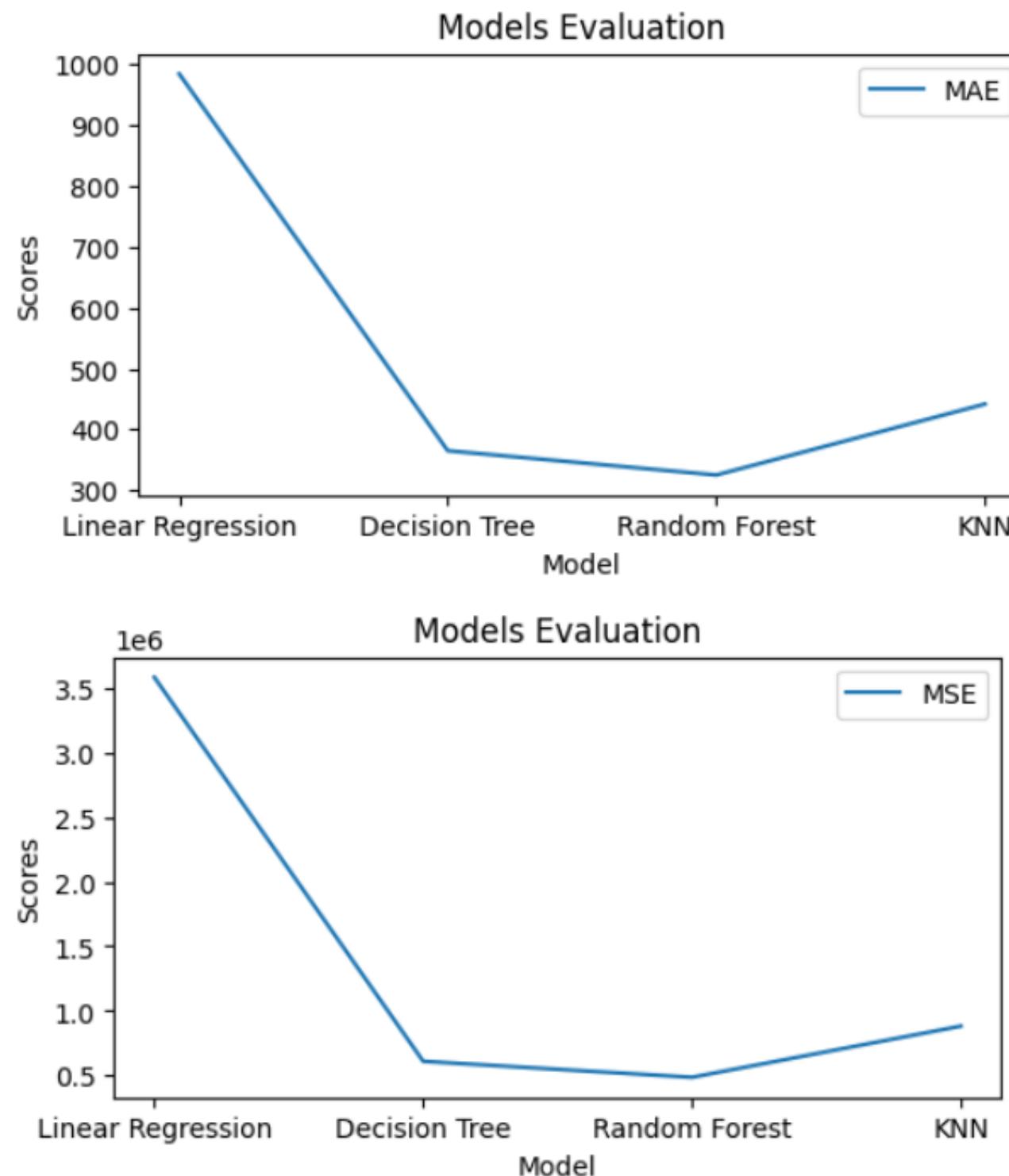
4.2 Báo cáo kết quả



Biểu đồ thể hiện chỉ số R2

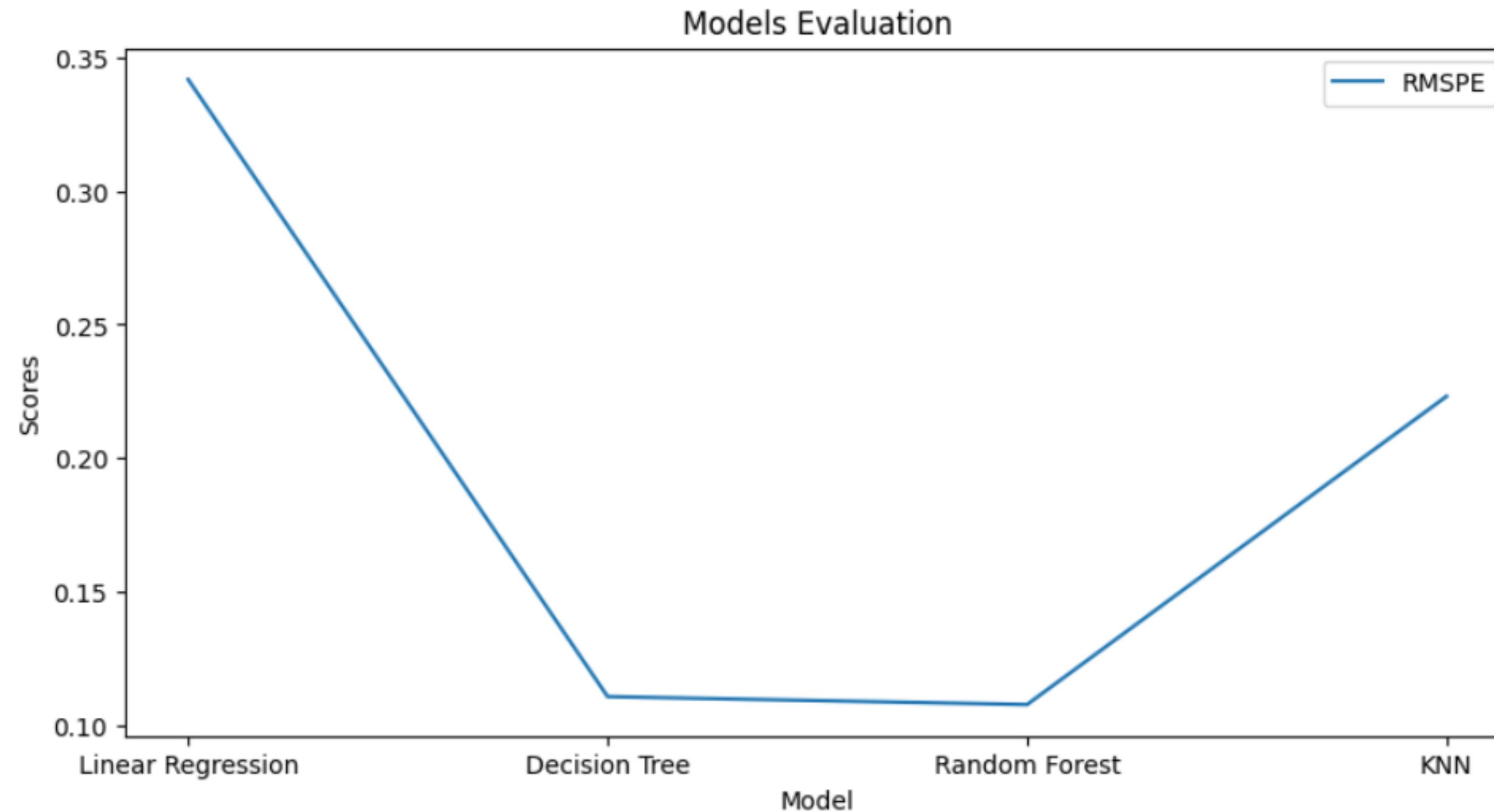
4. KẾT QUẢ THỰC NGHIỆM VÀ ĐỀ XUẤT

4.2 Báo cáo kết quả



4. KẾT QUẢ THỰC NGHIỆM VÀ ĐỀ XUẤT

4.2 Báo cáo kết quả



4. KẾT QUẢ THỰC NGHIỆM VÀ ĐỀ XUẤT

4.2 Báo cáo kết quả

- Kết luận chung: Tất cả 5 chỉ số R2, MAE, MSE, RMSE, RMPSE đều cho ra một kết quả mô hình giống nhau: Random Forest > Decision Tree > K-Nearest Neighbors > Linear Regression

4. KẾT QUẢ THỰC NGHIỆM VÀ ĐỀ XUẤT

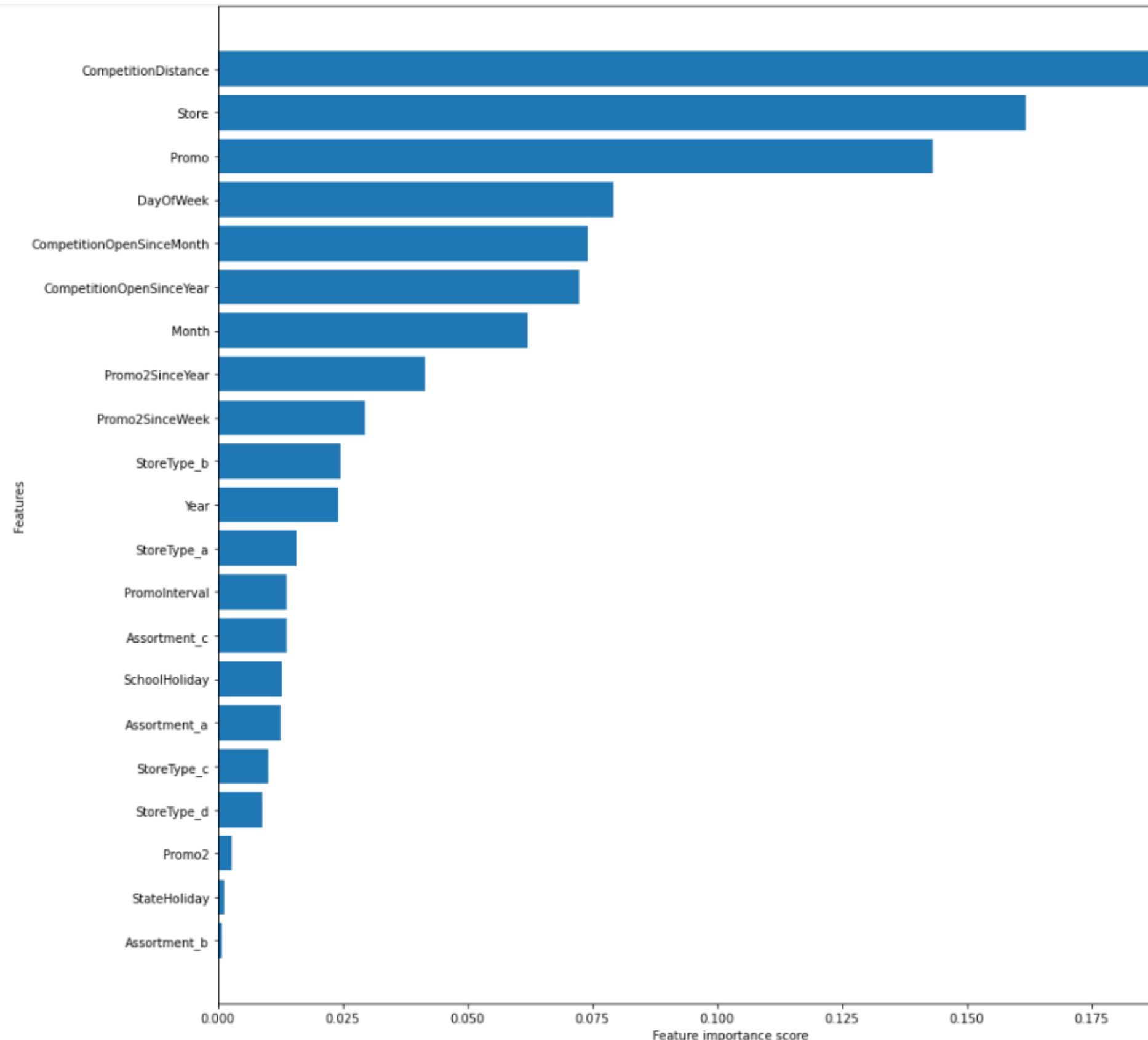
4.2 Báo cáo kết quả

- Features quan trọng ảnh hưởng tới mô hình Decision Tree

	Feature	Importance			
1	CompetitionDistance	0.198299	18	Assortment_a	0.022100
0	Store	0.164238	12	Year	0.022041
9	Promo	0.144933	14	StoreType_a	0.018030
8	DayOfWeek	0.077947	7	PromoInterval	0.013004
3	CompetitionOpenSinceYear	0.074483	11	SchoolHoliday	0.012598
2	CompetitionOpenSinceMonth	0.071056	17	StoreType_d	0.008711
13	Month	0.058147	16	StoreType_c	0.008279
6	Promo2SinceYear	0.040562	20	Assortment_c	0.005264
5	Promo2SinceWeek	0.030060	4	Promo2	0.002446
15	StoreType_b	0.026183	10	StateHoliday	0.001208
			19	Assortment_b	0.000411

4.2 Báo cáo kết quả

- Features quan trọng ảnh hưởng tới mô hình Random Forest



Kết luận chung: 10 features quan trọng nhất tác động đến hiệu quả mô hình là:

- CompetitionDistance
- Store
- Promo
- DayOfWeek
- CompetitionOpenSinceMonth
- CompetitionOpenSinceYear
- Month
- Promo2SinceYear
- Promo2SinceWeek
- StoreType_b

4.3 Đề xuất mô hình áp dụng

- Sau quá trình đào tạo và đánh giá từng mô hình, nhóm nhận thấy mô hình **Random Forest** đạt hiệu quả tốt nhất.
- Vì vậy nhà thuốc Rossmann nên áp dụng mô hình Random Forest để dự đoán doanh thu các cửa hàng nhằm có những điều chỉnh thích hợp, kịp thời trên kế hoạch kinh doanh của mình.

R2 in train	0.9305
R2 in test	0.9706
MAE	325.1
MSE	487562.1
RMSE	698.3
RMSPE	0.10782

5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

- Năm thuộc tính của bộ dữ liệu có ảnh hưởng mạnh mẽ nhất đến doanh thu xếp từ cao đến thấp: CompetitionDistance, Store, Promo, DayOfWeek, CompetitionOpenSinceMonth.
- Random Forest > Decision Tree > K-Nearest Neighbors > Linear Regression
- Mô hình Random Forest cho ra chỉ số RMPSE nhỏ hơn bài nghiên cứu trước đó

GO



5.2. Hạn chế

- Hạn chế về thời gian và tài nguyên máy
- Tập dữ liệu lớn chưa được cập nhật
- Nhiều siêu tham số được tìm hiểu nhưng chưa áp dụng vào bài
- Tập trung vào mô hình học máy, chưa mở rộng so sánh mô hình khác

GO



5.2. Hướng phát triển

- Mở rộng so sánh hiệu quả giữa các loại mô hình và đề xuất so sánh hiệu quả giữa phân tích thời gian và học máy
- Ứng dụng thêm các giải pháp tối ưu hiệu suất mô hình hiệu quả hơn
- Mở rộng thử nghiệm nhiều siêu tham số khác nhau

GO



THANK YOU

Vì đã lắng nghe bài thuyết trình.

```
    /ime + 1

    .SaveDialog()
    os.path.split(filePath)

    th + "\\"
    confirmation
    text = "Obj Sequence will be saved as:\n\n"
    path + objName + "###.obj\n\n"
    name = str(fromTime) + " to " + str(toTime) + " for " + str	animLength,
    l - c4d.gui.QuestionDialog(questionDialogText)

eedBool = True:

loop through animation and export frames
for x in range(0,animLength):

    change frame, redraw view
    time = c4d.BaseTime(fromTime,docFps) + c4d.BaseTime(x,docFps)
    tTime(moveTime)
    etAdd(c4d.EVENT_FORCEREDRAW)
    eView(c4d.DRAWFLAGS_FORCEFULLREDRAW)

    ar
    Text("Exporting " + str(x) + " of " + str(animLength))
    "(doc.GetTime().GetFrame(doc--))
```

