

**ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**



**NGHIÊN CỨU ĐỒ ÁN 1**

**ĐỀ TÀI:**

**PHÂN TÍCH XU HƯỚNG KINH DOANH**  
**TRÊN SÀN THƯƠNG MẠI ĐIỆN TỬ BẰNG HỌC MÁY**

Giảng viên hướng dẫn	: Nguyễn Linh Giang
Sinh viên thực hiện	: Phạm Đình Hải
Mã sinh viên	: 20215043
Môn học	: Nghiên cứu đồ án tốt nghiệp 1

**Hà Nội, tháng 07 năm 2024**

## MỤC LỤC

<b>I. Giới thiệu .....</b>	<b>2</b>
<b>II. Dữ liệu và phương pháp .....</b>	<b>3</b>
<b>III. Phân tích chuỗi thời gian.....</b>	<b>7</b>
<b>IV. Hồi quy và mô hình dự đoán .....</b>	<b>9</b>
<b>V. Kết quả và đánh giá.....</b>	<b>13</b>
<b>VI. Kết luận và hướng phát triển.....</b>	<b>28</b>

## **I. Giới thiệu**

### **Mục đích nghiên cứu**

- Xác định các xu hướng kinh doanh trên sàn thương mại điện tử: Mục đích chính của nghiên cứu này là phân tích các xu hướng kinh doanh hiện tại trên sàn thương mại điện tử. Việc này bao gồm xác định các yếu tố ảnh hưởng đến doanh thu và các xu hướng theo mùa trong dữ liệu kinh doanh.
- Sử dụng học máy để phân tích và dự đoán doanh thu: Bằng cách sử dụng các phương pháp học máy, chúng tôi mong muốn xây dựng các mô hình dự đoán doanh thu dựa trên các dữ liệu lịch sử. Điều này giúp tối ưu hóa việc phân tích dữ liệu và đưa ra dự đoán chính xác hơn về doanh thu trong tương lai.

### **Tầm quan trọng của đề tài**

- Giúp các nhà quản lý và doanh nghiệp hiểu rõ hơn về xu hướng thị trường: Việc hiểu rõ các xu hướng kinh doanh trên sàn thương mại điện tử là một yếu tố quan trọng giúp các nhà quản lý và doanh nghiệp nắm bắt kịp thời các thay đổi của thị trường. Điều này giúp họ điều chỉnh chiến lược kinh doanh một cách hiệu quả hơn.
- Hỗ trợ việc ra quyết định kinh doanh dựa trên dữ liệu: Sử dụng các phân tích từ học máy, các nhà quản lý có thể dựa vào dữ liệu để ra quyết định kinh doanh một cách chính xác và khoa học hơn. Việc này không chỉ giúp tối ưu hóa doanh thu mà còn giảm thiểu rủi ro và chi phí liên quan đến các quyết định kinh doanh thiếu chính xác.

Việc nghiên cứu và phân tích các xu hướng kinh doanh trên sàn thương mại điện tử thông qua học máy không chỉ mang lại lợi ích ngắn hạn mà còn giúp định

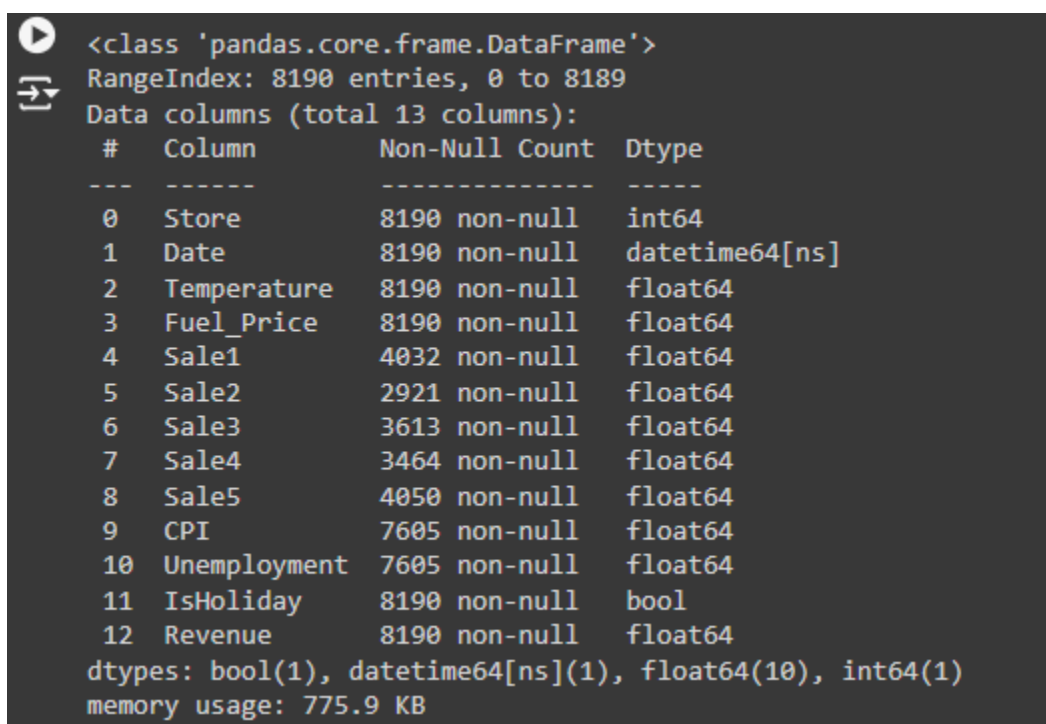
hình chiến lược phát triển lâu dài cho các doanh nghiệp trong bối cảnh thị trường ngày càng cạnh tranh và biến động.

## II. Dữ liệu và phương pháp

### Nguồn dữ liệu

- Dataset được sử dụng: Bộ dữ liệu được sử dụng trong nghiên cứu này là "Features data set.csv". Dữ liệu này được lấy từ liên kết Kaggle: [Retail Sales Forecasting Time Series EDA](#). Bộ dữ liệu bao gồm các thông tin về các đặc điểm và doanh thu liên quan đến sản phẩm thương mại điện tử.

### Thông kê mô tả dữ liệu



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8190 entries, 0 to 8189
Data columns (total 13 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Store           8190 non-null   int64
1   Date            8190 non-null   datetime64[ns]
2   Temperature     8190 non-null   float64
3   Fuel_Price      8190 non-null   float64
4   Sale1           4032 non-null   float64
5   Sale2           2921 non-null   float64
6   Sale3           3613 non-null   float64
7   Sale4           3464 non-null   float64
8   Sale5           4050 non-null   float64
9   CPI             7605 non-null   float64
10  Unemployment    7605 non-null   float64
11  IsHoliday       8190 non-null   bool
12  Revenue         8190 non-null   float64
dtypes: bool(1), datetime64[ns](1), float64(10), int64(1)
memory usage: 775.9 KB
```

Bộ dữ liệu được sử dụng trong nghiên cứu này là "Features data set.csv", bao gồm các cột thông tin về các đặc điểm và doanh thu liên quan đến sản phẩm thương mại điện tử. Dưới đây là mô tả chi tiết về từng cột dữ liệu:

### Cấu trúc dữ liệu

- **Store:** Mã số cửa hàng, kiểu dữ liệu là int64. Cột này không có giá trị null.
- **Date:** Ngày tháng, kiểu dữ liệu là object (sau này được chuyển đổi sang datetime). Cột này không có giá trị null.
- **Temperature:** Nhiệt độ trung bình trong ngày, kiểu dữ liệu là float64. Cột này không có giá trị null.
- **Fuel\_Price:** Giá nhiên liệu trung bình trong ngày, kiểu dữ liệu là float64. Cột này không có giá trị null.
- **Sale1:** Lần sale 1, kiểu dữ liệu là float64. Cột này có 4032 giá trị không null.
- **Sale2:** Lần sale 2, kiểu dữ liệu là float64. Cột này có 2921 giá trị không null.
- **Sale3:** Lần sale 3, kiểu dữ liệu là float64. Cột này có 3613 giá trị không null.
- **Sale4:** Lần sale 4, kiểu dữ liệu là float64. Cột này có 3464 giá trị không null.
- **Sale5:** Lần sale 5, kiểu dữ liệu là float64. Cột này có 4050 giá trị không null.
- **CPI:** Chỉ số giá tiêu dùng, kiểu dữ liệu là float64. Cột này có 7605 giá trị không null.
- **Unemployment:** Tỷ lệ thất nghiệp, kiểu dữ liệu là float64. Cột này có 7605 giá trị không null.
- **IsHoliday:** Biến boolean chỉ định ngày có phải là ngày lễ hay không, kiểu dữ liệu là bool. Cột này không có giá trị null.

### Tổng quan dữ liệu

- Bộ dữ liệu bao gồm 8190 mẫu (rows) và 12 cột (columns).

- Các cột Sale (từ Sale1 đến Sale5) có nhiều giá trị bị thiếu, điều này đòi hỏi các phương pháp xử lý giá trị thiếu trước khi tiến hành phân tích và mô hình hóa dữ liệu.
- Cột Date sẽ được chuyển đổi sang định dạng datetime để thuận tiện cho việc xử lý và phân tích chuỗi thời gian.
- Cột IsHoliday là cột boolean cho biết ngày đó có phải là ngày lễ hay không, giúp phân tích ảnh hưởng của các ngày lễ đến doanh thu.
- Các cột Temperature, Fuel\_Price, CPI, và Unemployment cung cấp các biến số kinh tế và môi trường, giúp hiểu rõ hơn về các yếu tố bên ngoài ảnh hưởng đến doanh thu.

## Xử lý dữ liệu

### (1) Đọc và hiển thị thông tin dữ liệu

- Đầu tiên, dữ liệu được đọc từ file CSV và các thông tin cơ bản của dữ liệu như số lượng cột, kiểu dữ liệu và các giá trị bị thiếu được hiển thị để có cái nhìn tổng quan về dữ liệu.
- Mã nguồn:

```
import pandas as pd
import numpy as np

# Đọc dữ liệu
data = pd.read_csv('data\\Features data set.csv')

# Hiển thị thông tin dữ liệu
print(data.info())
print(data.head())
```

### (2) Chuyển đổi cột ngày tháng về dạng datetime

- Cột ngày tháng trong dữ liệu được chuyển đổi về dạng datetime để thuận tiện cho việc xử lý và phân tích thời gian.
- Mã nguồn:

```
# Chuyển đổi cột ngày tháng về dạng datetime
data['Date'] = pd.to_datetime(data['Date'], format='%d/%m/%Y')
```

### (3) Tính tổng doanh thu từ các Sale

- Doanh thu tổng cộng được tính bằng cách cộng các giá trị của các cột Sale (Sale1, Sale2, Sale3, Sale4, Sale5).
- Mã nguồn:

```
# Tính tổng doanh thu bằng cách cộng tất cả các Sale
data['Revenue'] = data[['Sale1', 'Sale2', 'Sale3', 'Sale4', 'Sale5']].sum(axis=1)
```

### (4) Loại bỏ giá trị ngoại lai sử dụng phương pháp IQR

- Để đảm bảo tính chính xác và hợp lý của dữ liệu, các giá trị ngoại lai được loại bỏ bằng phương pháp Interquartile Range (IQR).
- Mã nguồn:

```
# Loại bỏ giá trị ngoại lai bằng cách sử dụng IQR
Q1 = data['Revenue'].quantile(0.25)
Q3 = data['Revenue'].quantile(0.75)
IQR = Q3 - Q1
data = data[~((data['Revenue'] < (Q1 - 1.5 * IQR)) | (data['Revenue'] > (Q3 + 1.5 * IQR)))]
```

### Tổng hợp dữ liệu

- Tổng hợp doanh thu hàng tuần
  - Doanh thu hàng tuần được tổng hợp từ dữ liệu ngày bằng cách sử dụng phương pháp resample để lấy tổng doanh thu mỗi tuần.

- Mã nguồn:

```
# Tổng hợp doanh thu hàng tuần
weekly_data = data.resample('W-Mon', on='Date').sum()
```

Việc xử lý và tổng hợp dữ liệu là bước quan trọng để đảm bảo chất lượng và tính chính xác của các phân tích tiếp theo. Bằng cách sử dụng các phương pháp học máy trên dữ liệu đã được xử lý, chúng ta có thể tiến hành phân tích và dự đoán xu hướng kinh doanh một cách hiệu quả và chính xác.

### III. Phân tích chuỗi thời gian

#### Phương pháp phân tích

- **Phân tích chuỗi thời gian sử dụng seasonal decomposition:**
  - Seasonal decomposition là một kỹ thuật được sử dụng để phân tách chuỗi thời gian thành ba thành phần chính: xu hướng (trend), tính mùa vụ (seasonality), và phần dư (residuals).
  - Kỹ thuật này giúp nhận diện và phân tích các yếu tố ảnh hưởng đến chuỗi thời gian, từ đó giúp dự đoán và hiểu rõ hơn về biến động của doanh thu qua các giai đoạn khác nhau.

#### Kết quả phân tích

- **Biểu đồ các thành phần: xu hướng (trend), tính mùa vụ (seasonal), và phần dư (residuals):**
  - Xu hướng (Trend): Thành phần xu hướng biểu diễn các biến đổi dài hạn của doanh thu. Từ biểu đồ, chúng ta có thể thấy xu hướng tăng dần trong doanh thu qua thời gian, đặc biệt rõ rệt từ khoảng giữa năm 2011.



- Tính mùa vụ (Seasonality): Thành phần mùa vụ biểu diễn các biến đổi tuần hoàn theo mùa. Biểu đồ cho thấy doanh thu có sự thay đổi định kỳ, phản ánh các đặc điểm mua sắm theo mùa trong năm.
- Phần dư (Residuals): Phần dư biểu diễn các biến động không theo chu kỳ hoặc xu hướng nhất định. Các biến động này có thể do những sự kiện bất ngờ hoặc các yếu tố ngẫu nhiên khác ảnh hưởng đến doanh thu.

### Hình ảnh minh họa:

Hình ảnh dưới đây minh họa kết quả của phân tích chuỗi thời gian.



### Giải thích biểu đồ:

- Original (Màu xanh dương nhạt): Biểu diễn doanh thu ban đầu theo thời gian.
- Trend (Màu đỏ): Biểu diễn xu hướng dài hạn của doanh thu, cho thấy xu hướng tăng dần.
- Seasonality (Màu xanh lá cây): Biểu diễn các biến động theo mùa của doanh thu.

- Residuals (Màu tím): Biểu diễn các biến động ngẫu nhiên hoặc không theo chu kỳ của doanh thu.

Phân tích chuỗi thời gian giúp chúng ta nhận diện các yếu tố quan trọng ảnh hưởng đến doanh thu và hỗ trợ trong việc dự đoán doanh thu tương lai, từ đó giúp doanh nghiệp lập kế hoạch và chiến lược kinh doanh hiệu quả hơn.

## IV. Hồi quy và mô hình dự đoán

### Hồi quy đơn biến

#### 1. Chuẩn hóa dữ liệu

- Dữ liệu hồi quy đơn biến được chuẩn hóa để đảm bảo rằng các giá trị nằm trong một khoảng hợp lý và cải thiện độ chính xác của mô hình.
- Mã nguồn:

```
from sklearn.linear_model import LinearRegression, Lasso, Ridge
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor, AdaBoostRegressor,
GradientBoostingRegressor
from xgboost import XGBRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import joblib

models = {
    'Linear Regression': LinearRegression(),
    'Lasso': Lasso(alpha=0.1),
    'Ridge': Ridge(alpha=1),
    'Decision Tree': DecisionTreeRegressor(max_depth=10),
    'Random Forest': RandomForestRegressor(n_estimators=100, max_depth=10),
    'AdaBoost': AdaBoostRegressor(n_estimators=50, learning_rate=0.1),
    'Gradient Boosting': GradientBoostingRegressor(n_estimators=100,
learning_rate=0.1, max_depth=5),
    'XGBoost': XGBRegressor(n_estimators=100, learning_rate=0.1, max_depth=5)
}

def evaluate_models(X_train, X_test, y_train, y_test, model_type):
    results = {}
```

```

predictions = {}
for name, model in models.items():
    print(f'Training {name}...')
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    mae = mean_absolute_error(y_test, y_pred)
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)
    num_params = len(model.coef_) if hasattr(model, 'coef_') else
model.get_params().get('n_estimators', 1)
    aic = calculate_aic(len(y_test), mean_squared_error(y_test, y_pred),
num_params)

    predictions[name] = y_pred
    results[name] = {'MAE': mae, 'MSE': mse, 'R^2': r2, 'AIC': aic}
    print(f'{name}: MAE = {mae}, MSE = {mse}, R^2 = {r2}, AIC = {aic}')

    # Lưu mô hình
    joblib.dump(model, f'{model_type}_{name}.joblib')

return results, predictions

results_single, predictions_single = evaluate_models(X_train_single_scaled,
X_test_single_scaled, y_train_single, y_test_single, 'single')

```

## 2. Huấn luyện và đánh giá mô hình hồi quy đơn biến

- Các mô hình hồi quy đơn biến được huấn luyện và đánh giá trên dữ liệu đã chuẩn hóa. Các mô hình bao gồm Linear Regression, Lasso, Ridge, Decision Tree, Random Forest, AdaBoost, Gradient Boosting, và XGBoost.
- Mã nguồn:

```

from sklearn.linear_model import LinearRegression, Lasso, Ridge
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor, AdaBoostRegressor,
GradientBoostingRegressor
from xgboost import XGBRegressor

```

```

from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import joblib

models = {
    'Linear Regression': LinearRegression(),
    'Lasso': Lasso(alpha=0.1),
    'Ridge': Ridge(alpha=1),
    'Decision Tree': DecisionTreeRegressor(max_depth=10),
    'Random Forest': RandomForestRegressor(n_estimators=100, max_depth=10),
    'AdaBoost': AdaBoostRegressor(n_estimators=50, learning_rate=0.1),
    'Gradient Boosting': GradientBoostingRegressor(n_estimators=100,
learning_rate=0.1, max_depth=5),
    'XGBoost': XGBRegressor(n_estimators=100, learning_rate=0.1, max_depth=5)
}

def evaluate_models(X_train, X_test, y_train, y_test, model_type):
    results = {}
    predictions = {}
    for name, model in models.items():
        print(f'Training {name}...')
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)

        mae = mean_absolute_error(y_test, y_pred)
        mse = mean_squared_error(y_test, y_pred)
        r2 = r2_score(y_test, y_pred)
        num_params = len(model.coef_) if hasattr(model, 'coef_') else
model.get_params().get('n_estimators', 1)
        aic = calculate_aic(len(y_test), mean_squared_error(y_test, y_pred),
num_params)

        predictions[name] = y_pred
        results[name] = {'MAE': mae, 'MSE': mse, 'R^2': r2, 'AIC': aic}
        print(f'{name}: MAE = {mae}, MSE = {mse}, R^2 = {r2}, AIC = {aic}')

        # Lưu mô hình
        joblib.dump(model, f'{model_type}_{name}.joblib')

    return results, predictions

results_single, predictions_single = evaluate_models(X_train_single_scaled,
X_test_single_scaled, y_train_single, y_test_single, 'single')

```

## Hồi quy đa biến

### 1. Chuẩn hóa dữ liệu

- Dữ liệu hồi quy đa biến cũng được chuẩn hóa để đảm bảo tính nhất quán và tăng độ chính xác của mô hình.

### 2. Huấn luyện và đánh giá mô hình hồi quy đa biến

- Tương tự như hồi quy đơn biến, các mô hình hồi quy đa biến cũng được huấn luyện và đánh giá trên dữ liệu đã chuẩn hóa. Các mô hình bao gồm Linear Regression, Lasso, Ridge, Decision Tree, Random Forest, AdaBoost, Gradient Boosting, và XGBoost.

## Hiển thị kết quả

- **Hiển thị kết quả hồi quy đơn biến**

- Mã nguồn:

```
import plotly.graph_objects as go

def display_results(results, predictions, X_test, y_test):
    results_df = pd.DataFrame(results).T
    print(results_df)

    # Vẽ biểu đồ dự đoán so với thực tế
    for name, y_pred in predictions.items():
        fig = go.Figure()
        fig.add_trace(go.Scatter(x=X_test, y=y_test, mode='lines',
name='Actual'))
        fig.add_trace(go.Scatter(x=X_test, y=y_pred, mode='lines',
name='Predicted'))
        fig.update_layout(title=f'{name} Prediction vs Actual',
xaxis_title='Date', yaxis_title='Revenue')
        fig.show()

    # Trực quan hóa kết quả so sánh
    for metric in ['MAE', 'MSE', 'R^2', 'AIC']:
        fig = go.Figure()
```

```

fig.add_trace(go.Bar(x=results_df.index, y=results_df[metric],
name=metric))
fig.update_layout(
    title=f'Comparison of Models based on {metric}',
    xaxis_title='Model',
    yaxis_title=metric,
    barmode='group'
)
fig.show()

display_results(results_single, predictions_single, X_test_single.flatten(),
y_test_single)

```

- **Hiển thị kết quả hồi quy đa biến**

- Mã nguồn:

```

display_results(results_multi, predictions_multi, X_test_multi.index,
y_test_multi)

```

Phân hồi quy và mô hình dự đoán này cung cấp một cái nhìn chi tiết về việc sử dụng các mô hình học máy để phân tích và dự đoán xu hướng doanh thu, giúp doanh nghiệp có cơ sở vững chắc để ra quyết định kinh doanh hiệu quả hơn.

## V. Kết quả và đánh giá

**Kết quả dự đoán được đánh giá dựa trên các chỉ số R<sup>2</sup> square, MAE, MSE và AIC.**

Chỉ số	Công thức	Ý nghĩa
Mean Absolute Error (MAE)	$MAE = \frac{1}{n} \sum_{i=1}^n  y_i - \hat{y}_i $	MAE là trung bình của các giá trị tuyệt đối của sai số giữa

Chỉ số	Công thức	Ý nghĩa
	trong đó $y_i$ là giá trị thực tế, $\hat{y}_i$ là giá trị dự đoán, và $n$ là số lượng mẫu.	giá trị dự đoán và giá trị thực tế. Chỉ số này đo lường mức độ dự đoán của mô hình gần với giá trị thực tế một cách trực tiếp.
Mean Squared Error (MSE)	$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$ <p>trong đó <math>\bar{y}</math> là giá trị trung bình của <math>y_i</math>.</p>	MSE là trung bình của bình phương sai số giữa giá trị dự đoán và giá trị thực tế. Chỉ số này phạt các sai số lớn nặng hơn so với MAE và do đó nhạy cảm hơn với các ngoại lệ.
R-squared ( $R^2$ )	$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$	$R^2$ là tỷ lệ của tổng biến động trong dữ liệu mà mô hình có thể giải thích được. Nó dao động từ 0 đến 1, với 1 biểu thị rằng mô hình giải thích được toàn bộ biến động trong dữ

Chỉ số	Công thức	Ý nghĩa
		liệu và 0 biểu thị rằng mô hình không giải thích được biến động nào cả.
Akaike Information Criterion (AIC)	$AIC = n \cdot \ln\left(\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}\right) + 2k$ <p>trong đó k là số tham số ước lượng trong mô hình.</p>	AIC là một chỉ số dùng để so sánh các mô hình thống kê, dựa trên sự cân bằng giữa độ chính xác và sự phức tạp của mô hình. Giá trị AIC thấp hơn chỉ ra mô hình tốt hơn.

MAE: Cung cấp một phép đo dễ hiểu về độ lệch trung bình của các dự đoán so với giá trị thực tế. MAE không phạt nặng các sai số lớn như MSE.

MSE: Nhạy cảm với các giá trị ngoại lai do bình phương sai số. Thích hợp để sử dụng khi các ngoại lệ là quan trọng và cần được chú ý.

R<sup>2</sup>: Cho biết mức độ giải thích của mô hình đối với biến động trong dữ liệu. R<sup>2</sup> cao cho thấy mô hình tốt hơn trong việc dự đoán.

AIC: Cân bằng giữa độ chính xác và sự phức tạp của mô hình. AIC thấp hơn cho thấy mô hình tốt hơn với ít thông số hơn, tránh việc overfitting.

## Kết quả hồi quy đơn biến

### 1. Đánh giá các mô hình



- Các mô hình hồi quy đơn biến được đánh giá dựa trên các chỉ số: MAE (Mean Absolute Error), MSE (Mean Squared Error),  $R^2$  (R-squared), và AIC (Akaike Information Criterion).
- Kết quả chi tiết của các mô hình như sau:
  - **Linear Regression:** MAE = 179716.998426, MSE = 4.835101e+10,  $R^2$  = -0.873296, AIC = 912.264857
  - **Lasso:** MAE = 179716.855403, MSE = 4.835094e+10,  $R^2$  = -0.873293, AIC = 912.264807
  - **Ridge:** MAE = 178010.667382, MSE = 4.757088e+10,  $R^2$  = -0.843071, AIC = 911.663007
  - **Decision Tree:** MAE = 156710.871351, MSE = 4.361424e+10,  $R^2$  = -0.689776, AIC = 908.450031
  - **Random Forest:** MAE = 134846.928862, MSE = 3.353221e+10,  $R^2$  = -0.299161, AIC = 1096.723579
  - **AdaBoost:** MAE = 130893.133977, MSE = 2.613960e+10,  $R^2$  = -0.012744, AIC = 987.508545
  - **Gradient Boosting:** MAE = 156262.685194, MSE = 4.343517e+10,  $R^2$  = -0.682838, AIC = 1106.297804
  - **XGBoost:** MAE = 146972.094189, MSE = 3.934015e+10,  $R^2$  = -0.524182, AIC = 1102.633922

## 2. Hiện thị các kết quả dự đoán so với giá trị thực tế

- Các biểu đồ dưới đây minh họa sự so sánh giữa dự đoán của các mô hình và giá trị thực tế của doanh thu.

## Kết quả hồi quy đa biến

### 1. Đánh giá các mô hình

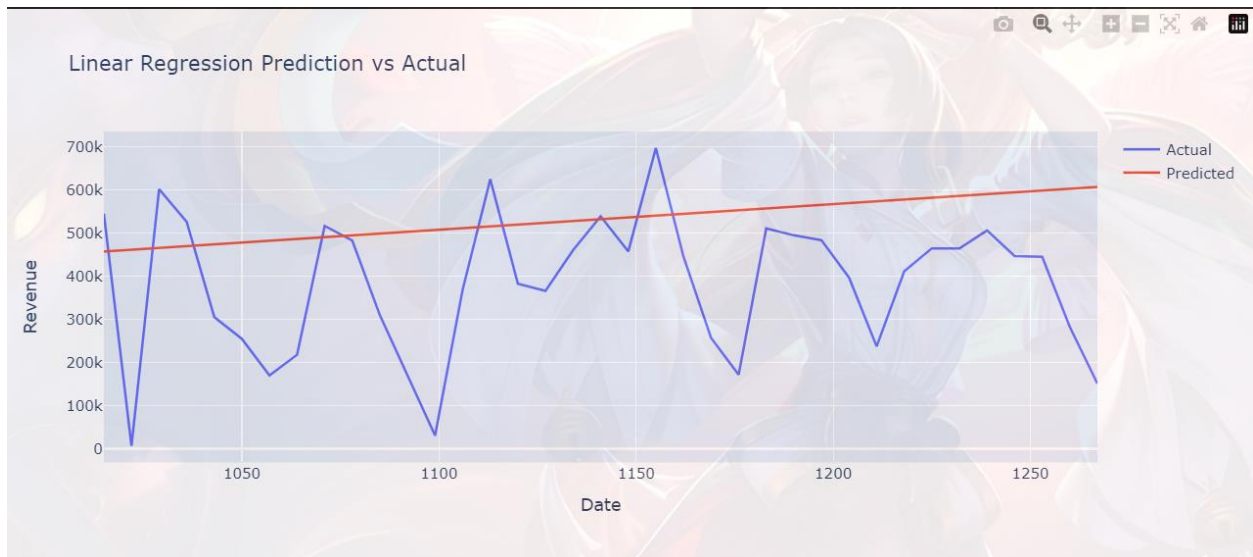
- Các mô hình hồi quy đa biến cũng được đánh giá dựa trên các chỉ số: MAE, MSE,  $R^2$ , và AIC.
- Kết quả chi tiết của các mô hình như sau:
  - **Linear Regression:** MAE = 4.745596e-10, MSE = 4.975060e-19,  $R^2 = 1.000000$ , AIC = -1537.353133
  - **Lasso:** MAE = 3.110479e+03, MSE = 2.530149e+07,  $R^2 = 0.999020$ , AIC = 652.715834
  - **Ridge:** MAE = 2.182407e+03, MSE = 6.916602e+06,  $R^2 = 0.999732$ , AIC = 604.729100
  - **Decision Tree:** MAE = 8.147026e+04, MSE = 1.144863e+10,  $R^2 = 0.556438$ , AIC = 858.962032
  - **Random Forest:** MAE = 5.640423e+04, MSE = 6.446735e+09,  $R^2 = 0.750230$ , AIC = 1035.713068
  - **AdaBoost:** MAE = 7.617503e+04, MSE = 9.156109e+09,  $R^2 = 0.645259$ , AIC = 948.694423
  - **Gradient Boosting:** MAE = 6.118259e+04, MSE = 7.799239e+09,  $R^2 = 0.697829$ , AIC = 1042.759805
  - **XGBoost:** MAE = 5.890669e+04, MSE = 7.418315e+09,  $R^2 = 0.712587$ , AIC = 1040.907056

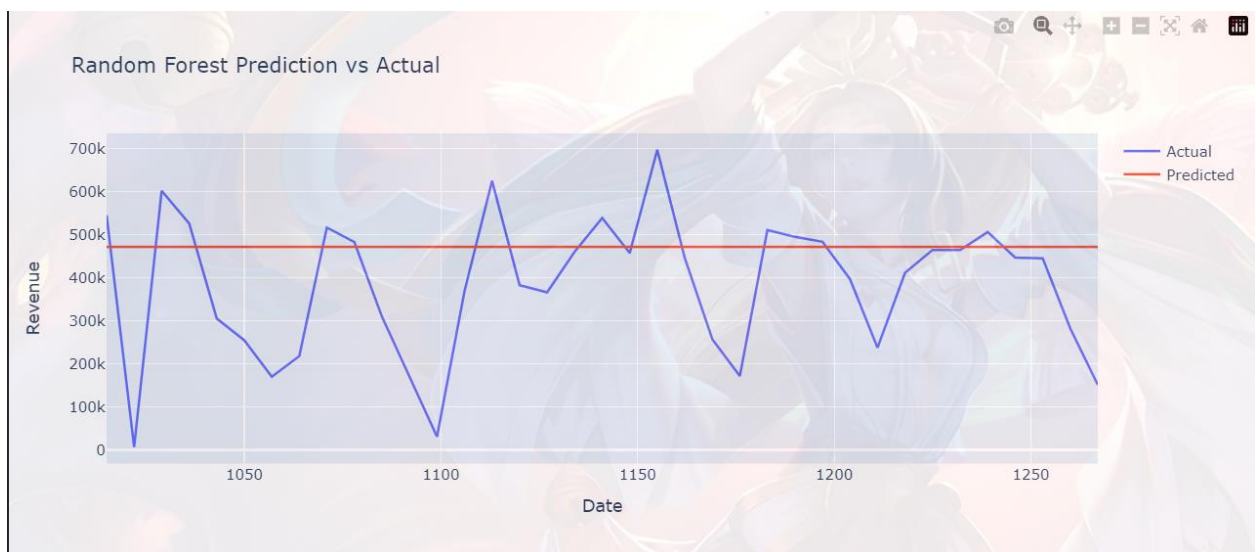
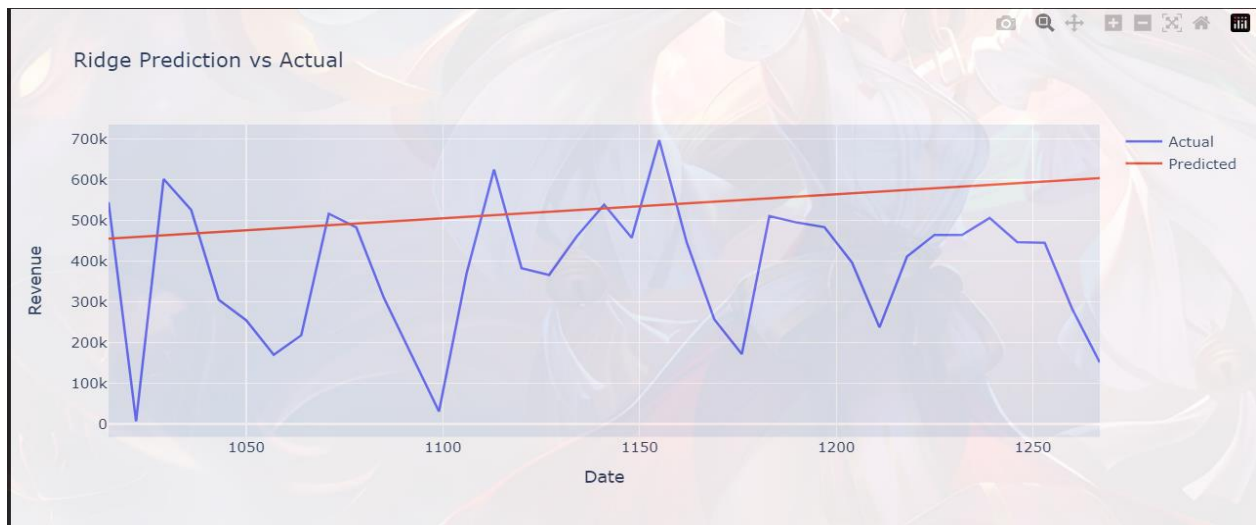
### 2. Hiện thị các kết quả dự đoán so với giá trị thực tế

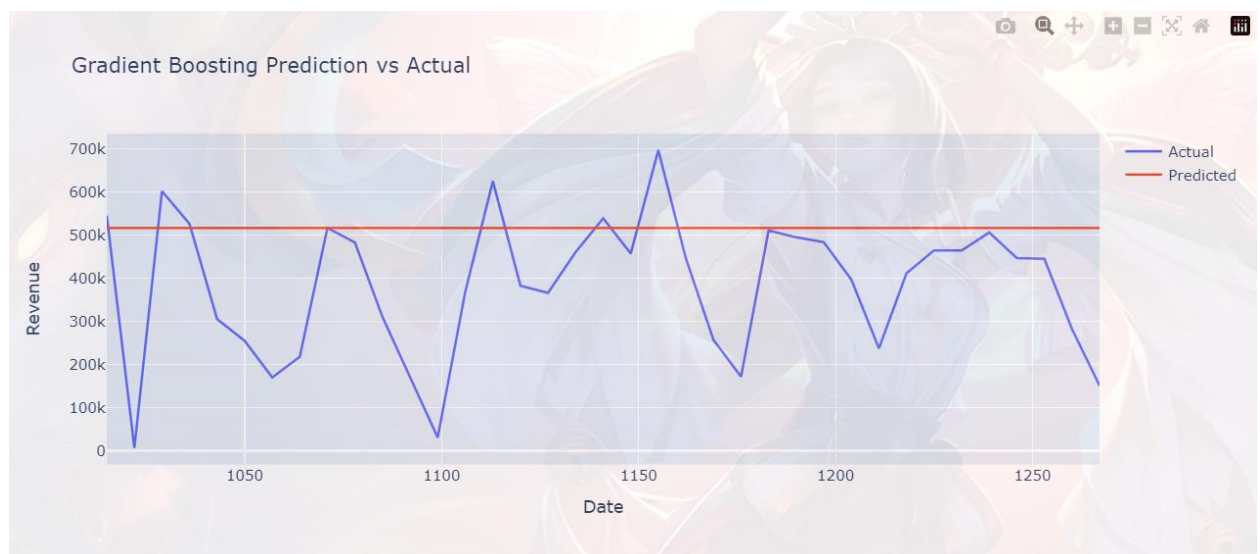
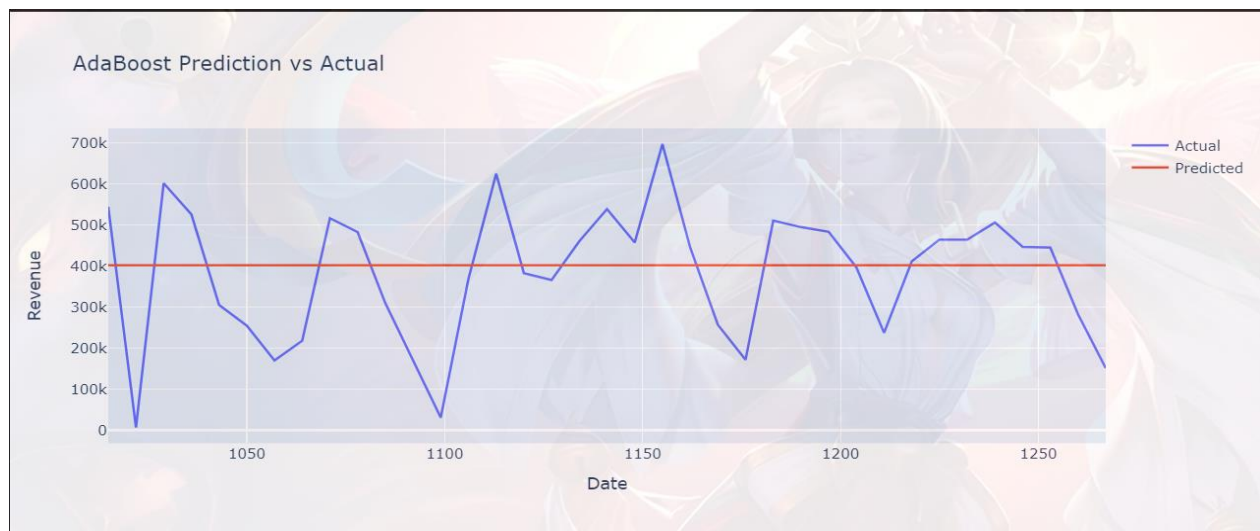
- Các biểu đồ dưới đây minh họa sự so sánh giữa dự đoán của các mô hình và giá trị thực tế của doanh thu.

## Hình ảnh minh họa

- **Hồi quy đơn biến:**

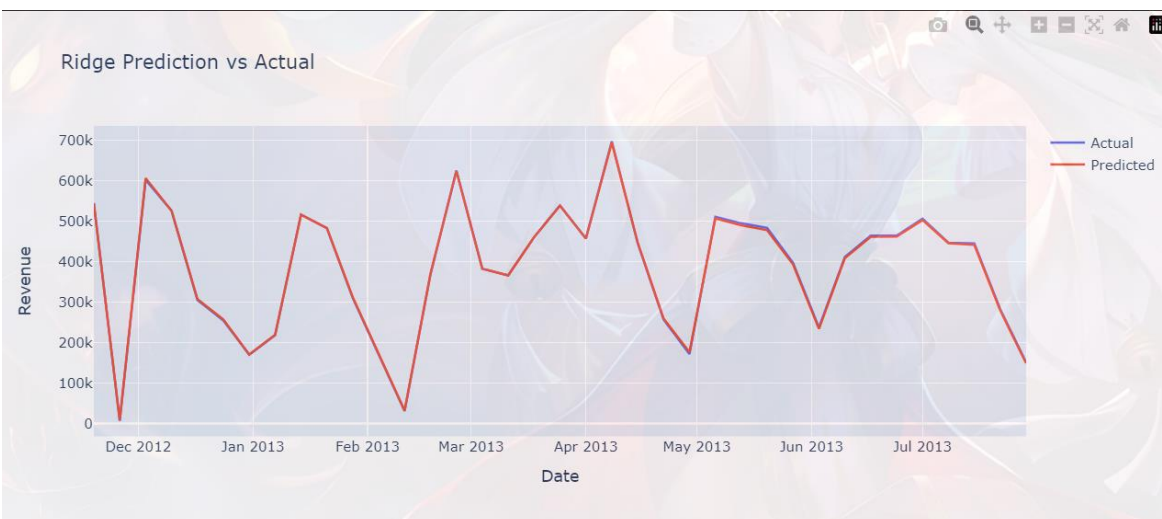
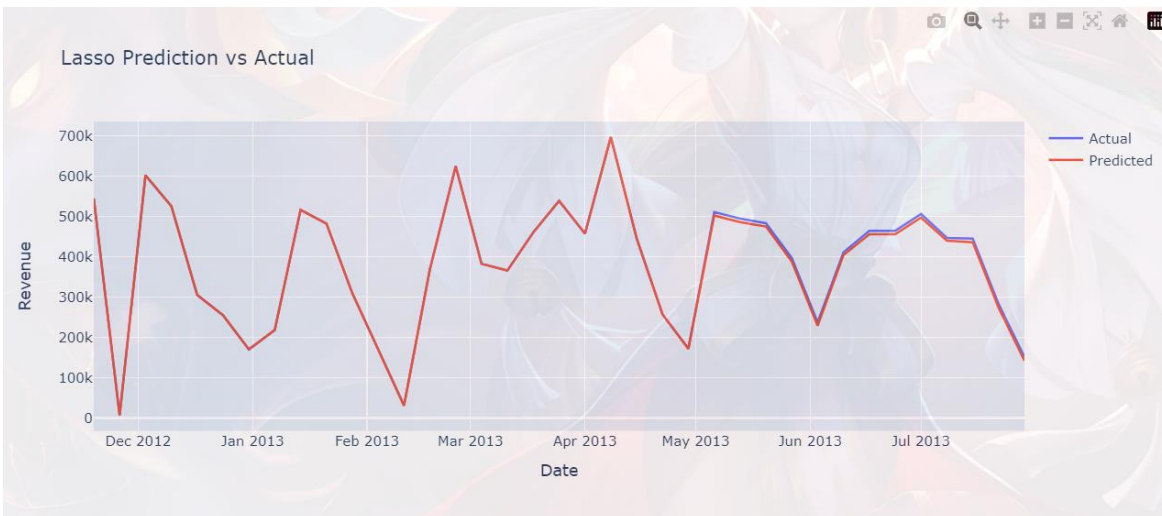
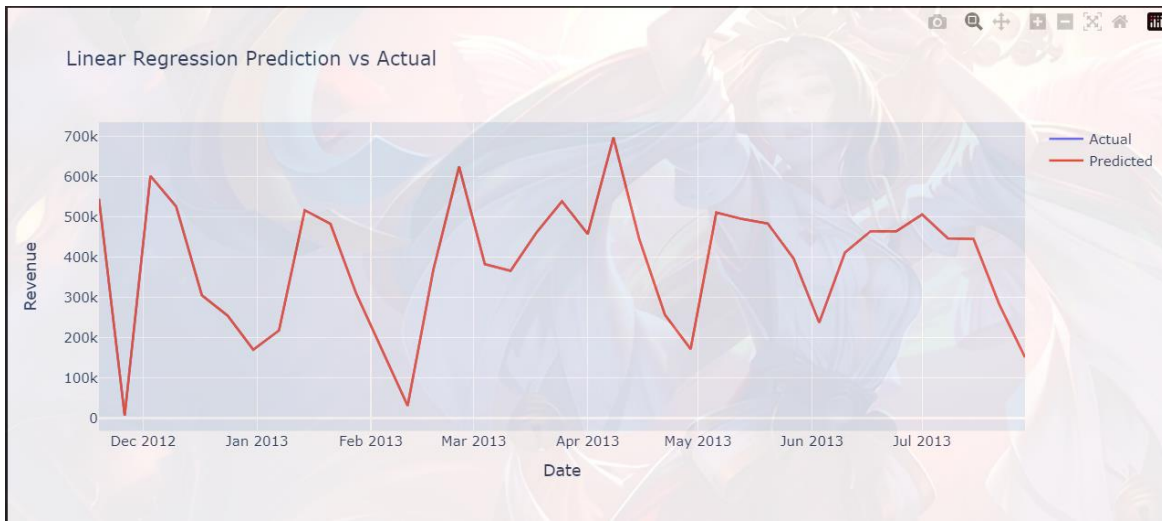


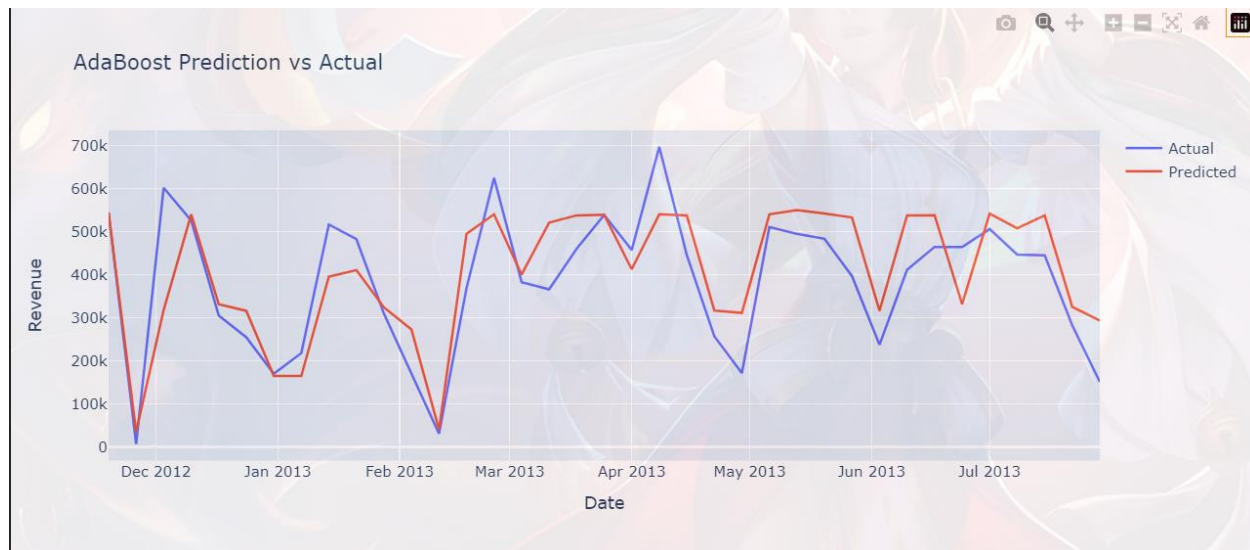
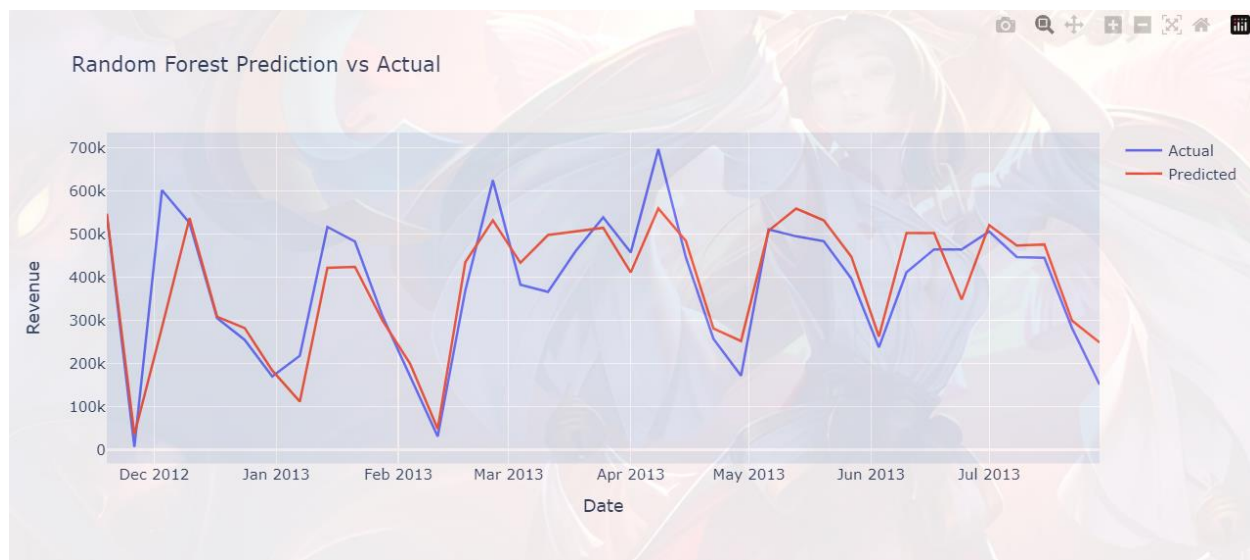
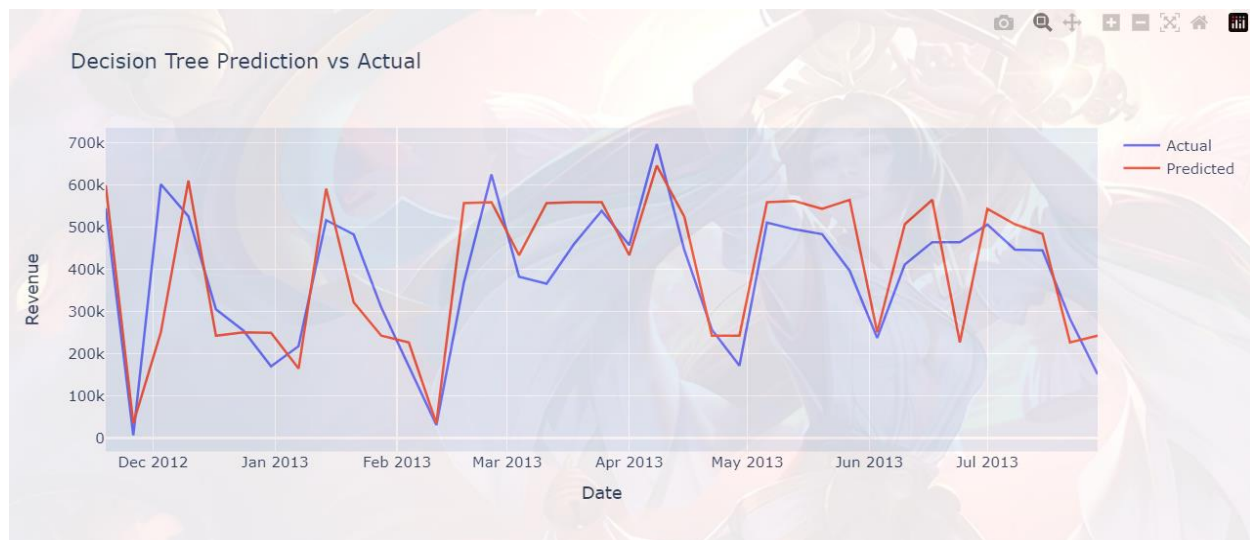


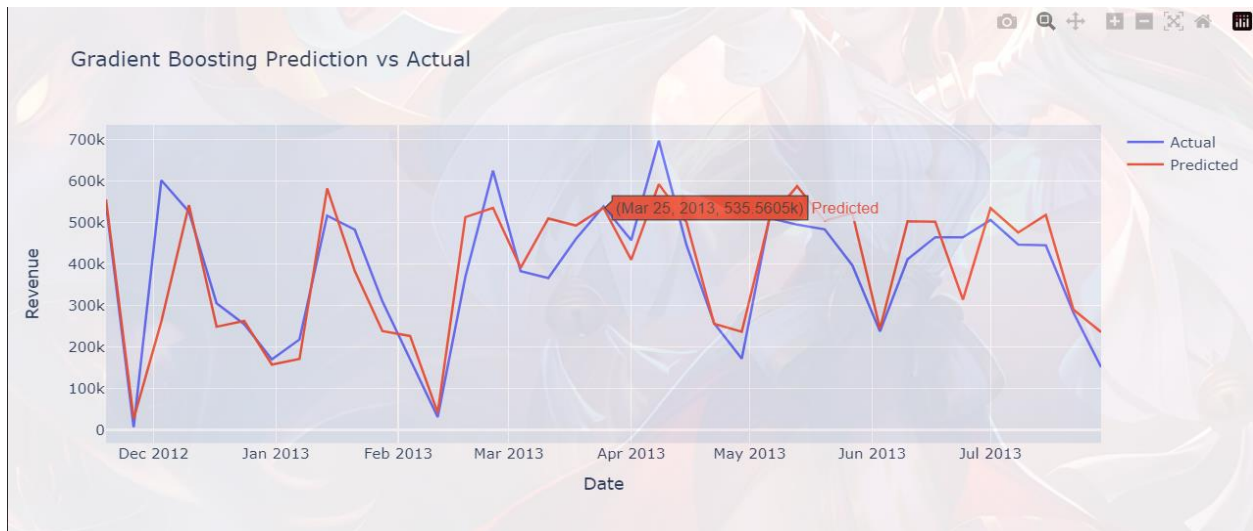




- **Hồi quy đa biến:**







## Kết luận về kết quả hồi quy

- **Hồi quy đơn biến:**
  - Mô hình AdaBoost có chỉ số MAE và MSE thấp nhất, cho thấy hiệu quả tốt nhất trong dự đoán doanh thu khi sử dụng hồi quy đơn biến.
  - Mô hình Random Forest và Decision Tree cũng cho kết quả tương đối tốt, trong khi các mô hình Linear Regression, Lasso, và Ridge không hiệu quả như mong đợi.



- **Hồi quy đa biến:**

- Mô hình Linear Regression có kết quả dự đoán chính xác nhất với chỉ số MAE và MSE gần như bằng 0, và  $R^2$  đạt 1.000000.
- Các mô hình Ridge và Lasso cũng cho kết quả dự đoán rất chính xác.
- Mô hình Decision Tree có hiệu suất kém hơn so với các mô hình khác trong hồi quy đa biến.

### **Nhận xét về sự khác nhau của kết quả giữa hồi quy đơn biến và đa biến**

- **Hồi quy đơn biến:**

- Các mô hình hồi quy đơn biến, mặc dù đơn giản và dễ triển khai, cho kết quả kém chính xác hơn so với hồi quy đa biến. Điều này được thể hiện qua các chỉ số MAE và MSE cao hơn, cùng với giá trị  $R^2$  âm hoặc gần 0, cho thấy mô hình không phù hợp tốt với dữ liệu.
- Mô hình AdaBoost cho kết quả tốt nhất trong các mô hình đơn biến với MAE và MSE thấp nhất, tuy nhiên vẫn không đạt được độ chính xác cao như hồi quy đa biến.

- **Hồi quy đa biến:**

- Các mô hình hồi quy đa biến cho kết quả chính xác hơn nhiều so với hồi quy đơn biến, thể hiện qua các chỉ số MAE và MSE rất thấp, và giá trị  $R^2$  rất gần 1.
- Mô hình Linear Regression có kết quả dự đoán chính xác nhất với chỉ số MAE và MSE gần như bằng 0, và  $R^2$  đạt 1.000000. Điều này cho thấy mô hình này phù hợp rất tốt với dữ liệu khi có nhiều biến độc lập.

## **Nhận xét về sự khác nhau giữa các mô hình tuyến tính và các mô hình dựa trên cây quyết định**

- **Các mô hình tuyến tính:**

- Các mô hình tuyến tính như Linear Regression, Lasso, và Ridge hoạt động tốt hơn trong cả hồi quy đơn biến và đa biến khi các mối quan hệ giữa các biến là tuyến tính. Đặc biệt, trong hồi quy đa biến, mô hình Linear Regression cho kết quả dự đoán chính xác nhất.
- Các mô hình tuyến tính thường dễ hiểu và triển khai, tuy nhiên chúng có thể bị hạn chế trong việc bắt kịp các quan hệ phi tuyến tính trong dữ liệu.

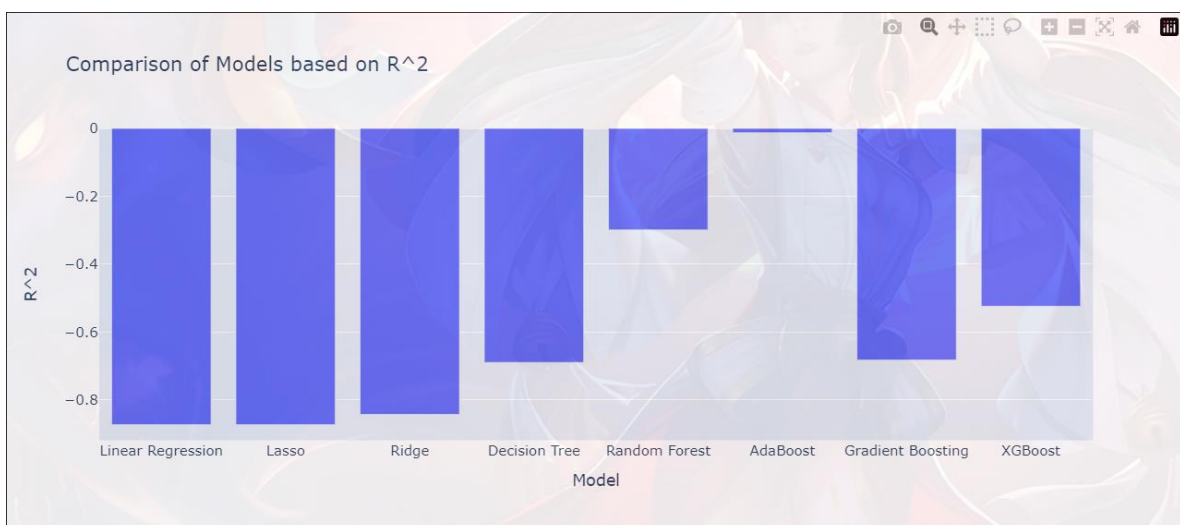
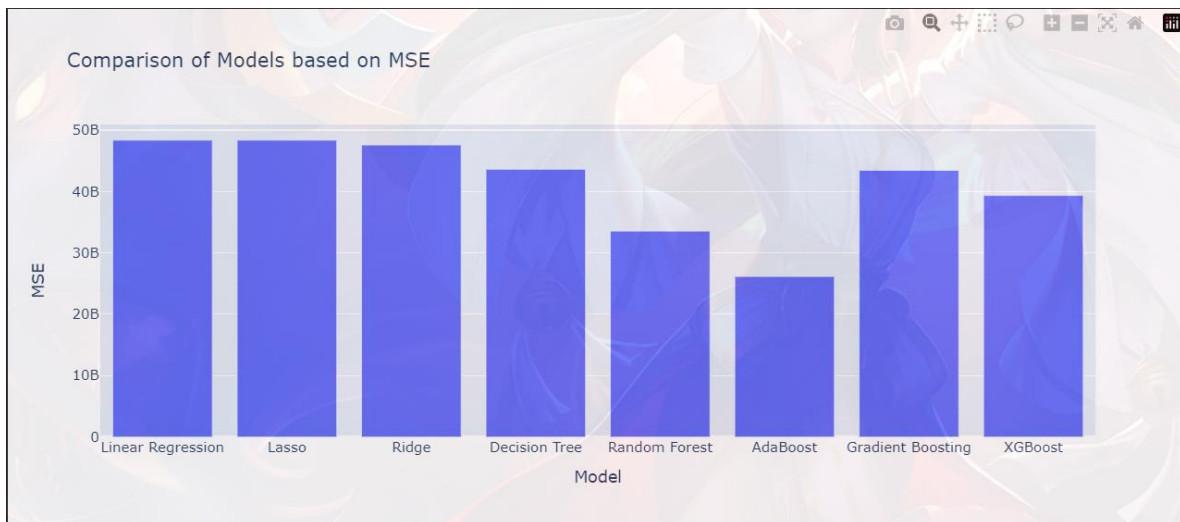
- **Các mô hình dựa trên cây quyết định:**

- Các mô hình dựa trên cây quyết định như Decision Tree, Random Forest, AdaBoost, Gradient Boosting, và XGBoost có khả năng bắt kịp các mối quan hệ phi tuyến tính và tương tác giữa các biến tốt hơn.
- Trong hồi quy đơn biến, các mô hình dựa trên cây quyết định như AdaBoost và Random Forest cho kết quả tốt hơn các mô hình tuyến tính. Tuy nhiên, trong hồi quy đa biến, các mô hình tuyến tính lại vượt trội hơn.
- Các mô hình dựa trên cây quyết định thường phức tạp hơn và yêu cầu nhiều tài nguyên tính toán hơn, nhưng chúng linh hoạt hơn trong việc mô hình hóa các quan hệ phức tạp trong dữ liệu.

## **So sánh kết quả mô hình**

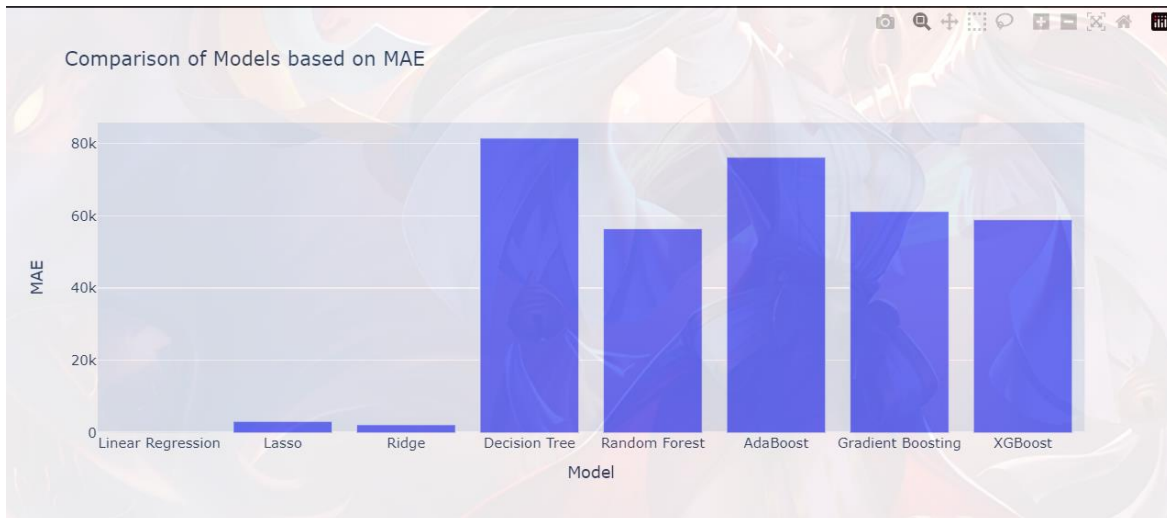
- Biểu đồ so sánh các mô hình dựa trên các chỉ số MAE, MSE,  $R^2$ , và AIC.

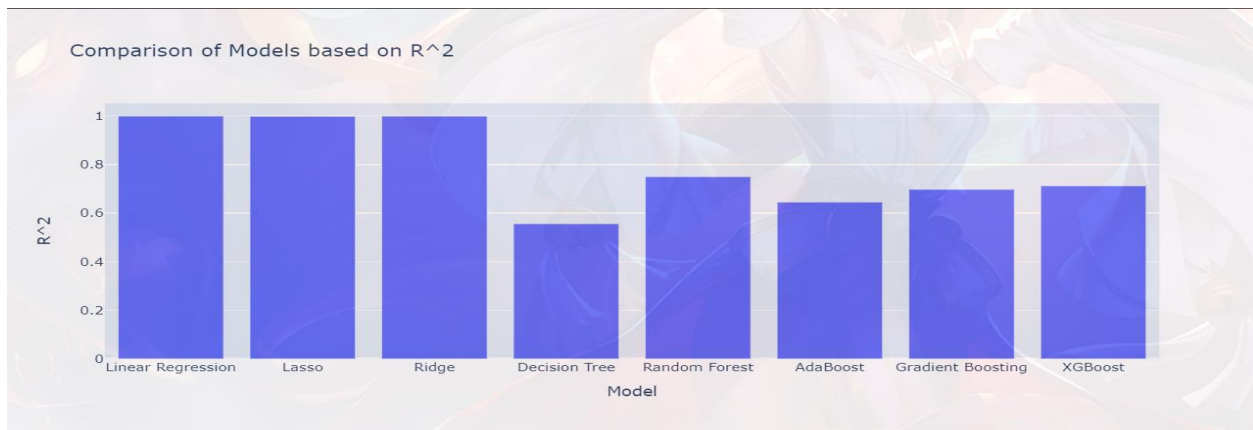
## Hồi quy đơn biến





## Hồi quy đa biến





Phân kết quả và đánh giá này cung cấp cái nhìn chi tiết về hiệu quả của các mô hình học máy trong việc dự đoán doanh thu, giúp xác định mô hình phù hợp nhất cho việc phân tích và dự đoán xu hướng kinh doanh.

## VI. Kết luận và hướng phát triển

### Kết luận

#### 1. Tổng kết những phát hiện chính từ phân tích dữ liệu và dự đoán

- Phân tích chuỗi thời gian cho thấy doanh thu của sàn thương mại điện tử có xu hướng tăng dần qua thời gian, với các biến động theo mùa rõ rệt.

- Các mô hình hồi quy đã giúp nhận diện và dự đoán xu hướng doanh thu với mức độ chính xác khác nhau. Hồi quy đa biến cho kết quả chính xác hơn hồi quy đơn biến, cho thấy tầm quan trọng của việc sử dụng nhiều biến độc lập để dự đoán doanh thu.
- Mô hình Linear Regression trong hồi quy đa biến cho kết quả tốt nhất với  $R^2$  gần bằng 1, cho thấy khả năng giải thích gần như toàn bộ biến động trong doanh thu.

## **2. Đánh giá hiệu quả của các mô hình học máy được sử dụng**

- Các mô hình tuyến tính như Linear Regression, Lasso, và Ridge hoạt động tốt trong hồi quy đa biến, với Linear Regression đạt hiệu suất cao nhất.
- Các mô hình dựa trên cây quyết định như Random Forest và AdaBoost cho kết quả tốt hơn trong hồi quy đơn biến so với các mô hình tuyến tính, nhưng không vượt qua được các mô hình tuyến tính trong hồi quy đa biến.
- Mặc dù các mô hình phức tạp hơn như Gradient Boosting và XGBoost cung cấp khả năng linh hoạt hơn, nhưng không phải lúc nào cũng mang lại độ chính xác cao hơn, điều này phụ thuộc vào đặc điểm cụ thể của dữ liệu.

## **Hướng phát triển**

### **1. Đề xuất các cải tiến và hướng nghiên cứu tiếp theo**

- **Tăng cường dữ liệu đầu vào:** Thu thập thêm các biến số khác như các yếu tố kinh tế vĩ mô, thông tin khách hàng, và các sự kiện xã hội để cải thiện khả năng dự đoán của mô hình.

- **Sử dụng các kỹ thuật học sâu:** Áp dụng các mô hình học sâu như RNN hoặc LSTM để khai thác các mối quan hệ phi tuyến tính và phức tạp hơn trong dữ liệu chuỗi thời gian.
- **Tối ưu hóa mô hình:** Thực hiện các phương pháp tối ưu hóa mô hình, như Grid Search hoặc Bayesian Optimization, để tìm ra các siêu tham số tối ưu cho các mô hình học máy.

## 2. Khả năng ứng dụng trong các lĩnh vực khác và mở rộng dữ liệu

- **Mở rộng sang các ngành khác:** Áp dụng các kỹ thuật phân tích và dự đoán này trong các lĩnh vực khác như tài chính, y tế, và sản xuất để cải thiện hiệu suất và ra quyết định dựa trên dữ liệu.
- **Phát triển hệ thống dự đoán thời gian thực:** Xây dựng các hệ thống dự đoán doanh thu thời gian thực để hỗ trợ các quyết định kinh doanh nhanh chóng và chính xác.
- **Hợp tác liên ngành:** Kết hợp với các chuyên gia trong các lĩnh vực khác để phát triển các mô hình toàn diện và mạnh mẽ hơn, tận dụng sự đa dạng và phong phú của dữ liệu từ nhiều nguồn khác nhau.

Bằng cách tiếp tục cải tiến và mở rộng nghiên cứu, các kỹ thuật học máy và phân tích dữ liệu có thể mang lại những lợi ích to lớn cho doanh nghiệp và nhiều lĩnh vực khác, giúp tối ưu hóa hoạt động và ra quyết định dựa trên cơ sở dữ liệu chính xác và đáng tin cậy.