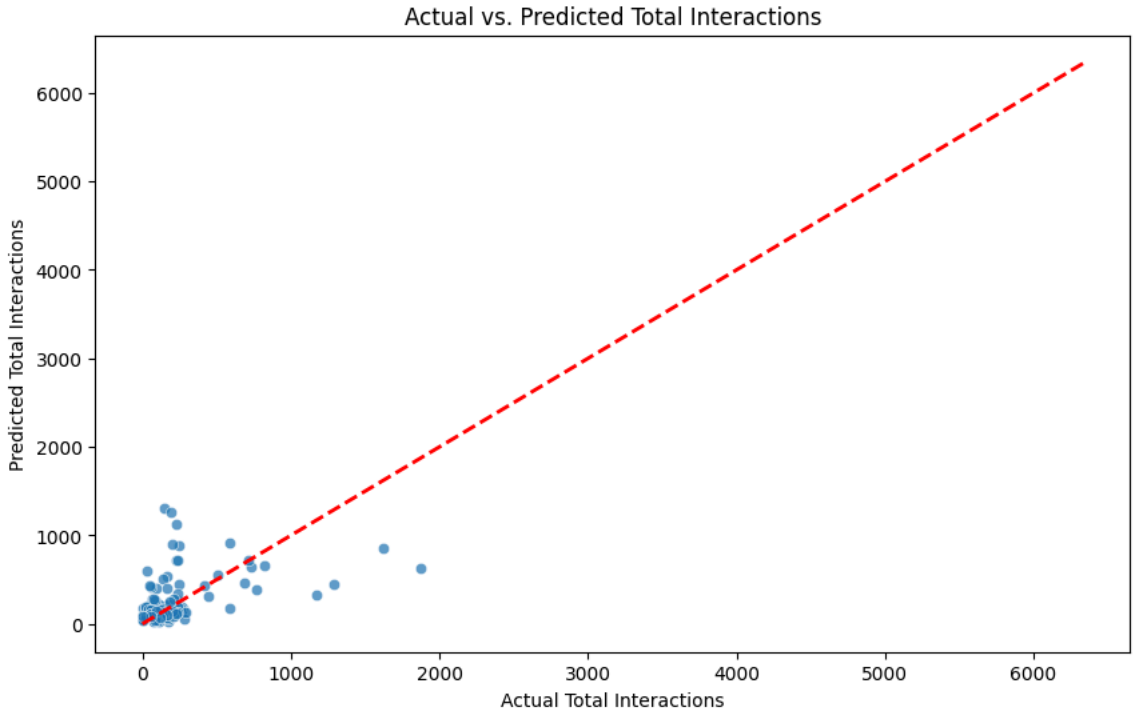
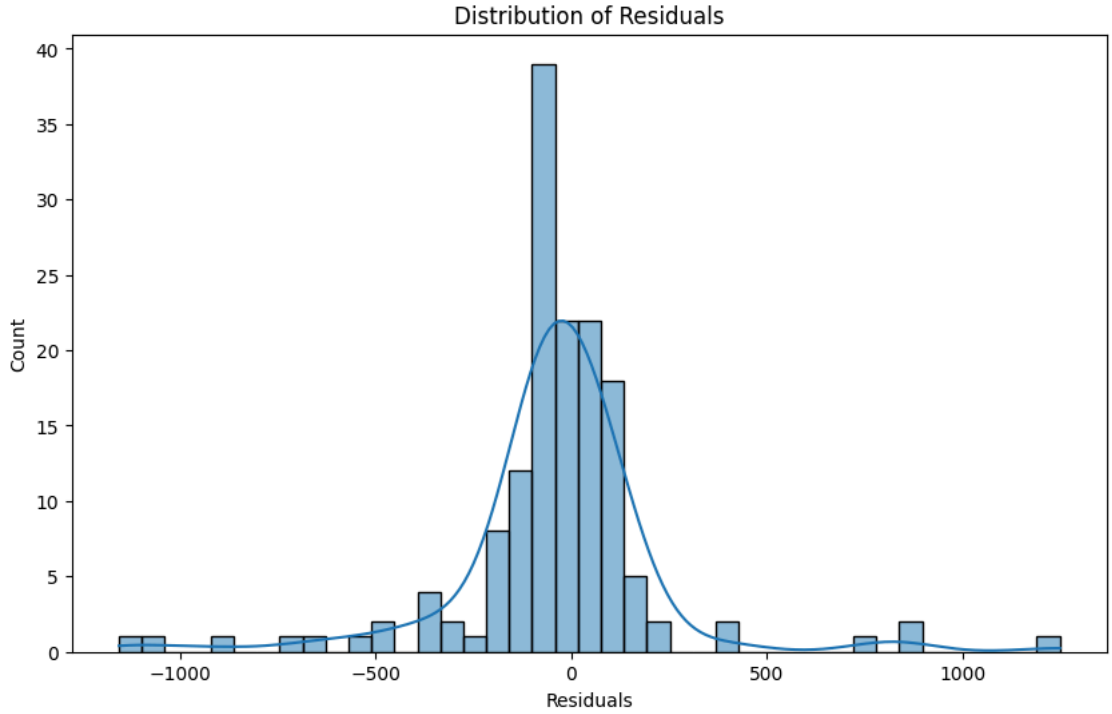
Notebook ValueError

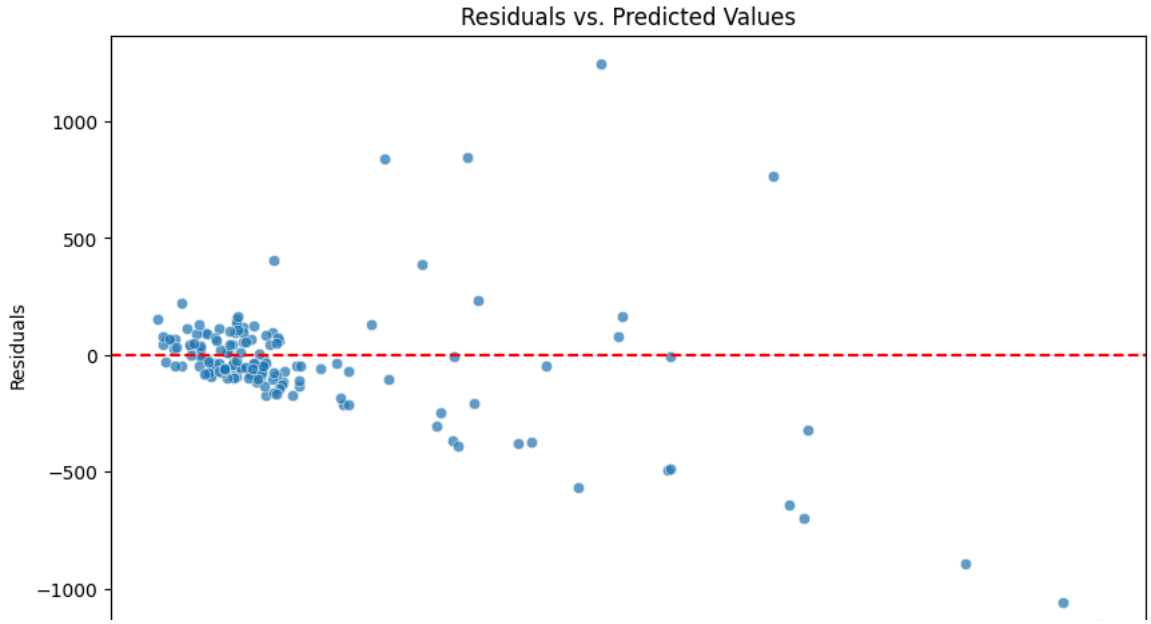
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_squared_error
from sklearn.preprocessing import StandardScaler, LabelEncoder

# Import dataset
data = pd.read_csv("/content/Facebook.csv")
print(data)

# Analyze the dataset
print(data.columns.unique())  # View column names
print(data.info())  # Check data types and missing values
print(data.describe())  # Get summary statistics

# Handle missing values (consider your specific needs)
data = data.dropna()  # Dropping rows with missing values

print(data.shape)  # Check data dimensions after handling missing values
print(data.isnull().sum())  # Ensure there are no missing values left

# Preprocess the data
df = pd.DataFrame(data, columns=['Type', 'Post Month', 'Post Weekday', 'Post Hour', 'Lifetime Post Total Reach', 'Total Interactions'])

# Apply label encoding to the 'Type' column
le = LabelEncoder()
df['Type_Encoded'] = le.fit_transform(df['Type'])

# Drop the original 'Type' column as it is no longer needed
df = df.drop(columns=['Type'])  # Drop the original 'Type' column

print(df)

# Split the dataset into independent and dependent variables
X = df.drop(columns='Total Interactions')  # Use the dataframe with 'Type' encoded
y = df['Total Interactions']

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Create a Linear Regression model
model = LinearRegression()

# Train the model
model.fit(X_train, y_train)  # Now X_train contains numerical data only

# Test the model
y_pred = model.predict(X_test)

# Calculate the Mean Squared Error , rmse ,r2
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print(f'Mean Squared Error: {mse}')
print(f'Root Mean Squared Error: {rmse}')
print(f'R² Score: {r2}')
```

```
Lifetime Post Impressions by people who have liked your Page            0
Lifetime Post reach by people who like your Page                        0
Lifetime People who have liked your Page and engaged with your post     0
comment                                                                 0
like                                                                    0
share                                                                   0
Total Interactions                                                      0
dtype: int64
     Post Month  Post Weekday  Post Hour  Lifetime Post Total Reach  \
0            12             4          3                       2752
1            12             3         10                      10460
2            12             3          3                       2413
3            12             2         10                      50128
4            12             2          3                       7244
..          ...           ...        ...                        ...
494           1             7         10                       5400
495           1             7          2                       4684
496           1             5          8                       3480
497           1             5          2                       3778
498           1             4         11                       4156

     Total Interactions  Type_Encoded
0                   100             1
1                   164             2
2                    80             1
3                  1777             1
4                   393             1
..                  ...           ...
494                 176             1
495                  84             1
496                  75             1
497                 115             1
498                 136             1

[495 rows x 6 columns]
Mean Squared Error: 72353.85002969483
Root Mean Squared Error: 268.9867097640603
R² Score: -0.016055772925068323
```

```
plt.figure(figsize=(10, 6))
sns.scatterplot(x=y_test, y=y_pred, alpha=0.7)
plt.plot([y.min(), y.max()], [y.min(), y.max()], 'r--', lw=2)
plt.xlabel('Actual Total Interactions')
plt.ylabel('Predicted Total Interactions')
plt.title('Actual vs. Predicted Total Interactions')
plt.show()

# Plot Residuals
residuals = y_test - y_pred
plt.figure(figsize=(10, 6))
sns.histplot(residuals, kde=True)
plt.xlabel('Residuals')
plt.title('Distribution of Residuals')
plt.show()

# Residuals vs. Fitted plot
plt.figure(figsize=(10, 6))
sns.scatterplot(x=y_pred, y=residuals, alpha=0.7)
plt.axhline(y=0, color='r', linestyle='--')
plt.xlabel('Predicted Total Interactions')
plt.ylabel('Residuals')
plt.title('Residuals vs. Predicted Values')
plt.show()
```


Actual vs. Predicted Total Interactions
Predicted Total Interactions
Actual Total Interactions
0
1000
2000
3000
4000
5000
6000


Distribution of Residuals
Count
Residuals
0
5
10
15
20
25
30
35
40
−1000
−500
0
500
1000


Residuals vs. Predicted Values
Residuals
1000
500
0
−500
−1000