In [ ]:

```python
# Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
data = pd.read_csv('instagram_data.csv')

# Extract input features (X) and target variables (y)
X = data[['Followers']]  # Assuming 'Followers' is one of the input features
y_likes = data['Likes']
y_time_since_posted = data['Time_Since_Posted']

# Split the data into training and testing sets
X_train, X_test, y_likes_train, y_likes_test, y_time_train, y_time_test = train_test_split(
    X, y_likes, y_time_since_posted, test_size=0.2, random_state=42
)

# Standardize the input features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Build a model for predicting Likes
model_likes = LinearRegression()
model_likes.fit(X_train_scaled, y_likes_train)

# Build a model for predicting Time Since Posted
model_time_since_posted = LinearRegression()
model_time_since_posted.fit(X_train_scaled, y_time_train)

# Make predictions
likes_predictions = model_likes.predict(X_test_scaled)
time_since_posted_predictions = model_time_since_posted.predict(X_test_scaled)

# Evaluate the models
mse_likes = mean_squared_error(y_likes_test, likes_predictions)
mse_time_since_posted = mean_squared_error(y_time_test, time_since_posted_predictions)

print(f'Mean Squared Error (Likes): {mse_likes}')
print(f'Mean Squared Error (Time Since Posted): {mse_time_since_posted}')

# Visualize the predictions
plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)
sns.scatterplot(x=y_likes_test, y=likes_predictions)
plt.title('Likes Prediction Evaluation')

plt.subplot(1, 2, 2)
sns.scatterplot(x=y_time_test, y=time_since_posted_predictions)
plt.title('Time Since Posted Prediction Evaluation')

plt.tight_layout()
plt.show()
```