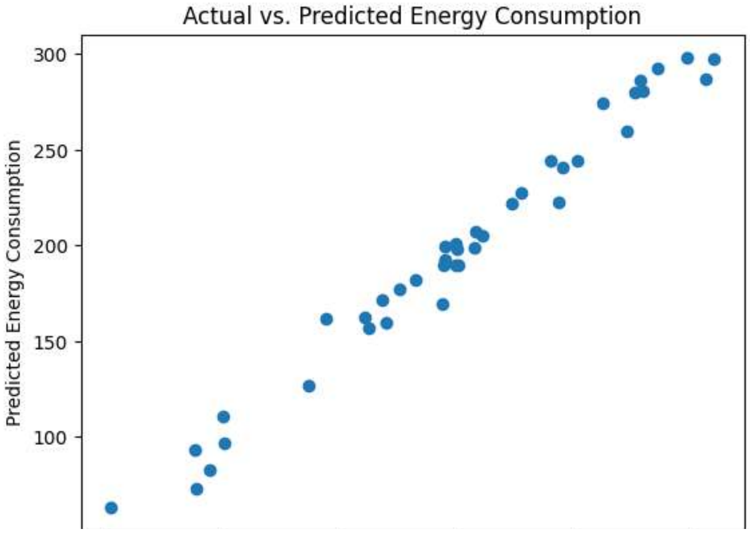
Building Energy Consumption Prediction with Linear Regression

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

# Generate synthetic data
np.random.seed(42)
# Generate random data for a building's energy consumption
n_samples = 200
occupants = np.random.randint(1, 100, n_samples)
outside_temperature = np.random.uniform(-10, 35, n_samples)
time_of_day = np.random.uniform(0, 24, n_samples)
energy_consumption = 50 + 2 * occupants + 3 * outside_temperature - 1.5 * time_of_day + np.random.normal(0, 10, n_samples)

# Create a DataFrame
data = pd.DataFrame({'Occupants': occupants, 'Outside_Temperature': outside_temperature, 'Time_of_Day': time_of_day, 'Energy_Consumption': en

# Split the data into training and testing sets
X = data[['Occupants', 'Outside_Temperature', 'Time_of_Day']]
y = data['Energy_Consumption']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create a Linear Regression model
model = LinearRegression()

# Train the model on the training data
model.fit(X_train, y_train)

# Make predictions on the test data
y_pred = model.predict(X_test)

# Calculate and print the model's performance
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f'Mean Squared Error: {mse:.2f}')
print(f'R-squared (R2) Score: {r2:.2f}')

# Plot the actual vs. predicted energy consumption
plt.scatter(y_test, y_pred)
plt.title('Actual vs. Predicted Energy Consumption')
plt.xlabel('Actual Energy Consumption')
plt.ylabel('Predicted Energy Consumption')
plt.show()
```

```
Mean Squared Error: 91.74
R-squared (R2) Score: 0.98
```


Actual vs. Predicted Energy Consumption
Predicted Energy Consumption