```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, r2_score

# Load the dataset
file_path = "datasetmain.xlsx"
dataset = pd.read_excel(file_path)

# Preprocess the dataset
# Select relevant columns
relevant_data = dataset[['INSOLATION (Wh/m^2/day)', 'INPUT POWER(kW)', 'OUTPUT POWER(kW)']].copy()

# Add a constant column for daily demand
daily_demand_kw = 8.333333  # Given daily demand in kW
relevant_data['DAILY DEMAND (kW)'] = daily_demand_kw

# Calculate the EB supply (demand - output power)
relevant_data['EB SUPPLY (kW)'] = relevant_data['DAILY DEMAND (kW)'] - relevant_data['OUTPUT POWER(kW)']

# Assume an electricity tariff rate (e.g., $0.12 per kWh)
tariff_rate = 4  # Example rate in Rs per kWh
relevant_data['TARIFF PER DAY (Rs)'] = relevant_data['EB SUPPLY (kW)'] * tariff_rate

# Define features (X) and target (y)
X = relevant_data[['INSOLATION (Wh/m^2/day)', 'INPUT POWER(kW)', 'OUTPUT POWER(kW)']]
y = relevant_data['TARIFF PER DAY (Rs)']

# Split the dataset into training (90%) and testing (10%) sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=42)

# Train a Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Predict on the test set
y_pred = model.predict(X_test)

# Evaluate the model
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

# Display evaluation metrics
print(f"Mean Absolute Error (MAE): {mae}")
print(f"R-squared (R²): {r2}")

# Display predictions
predictions = pd.DataFrame({'Actual Tariff (RS)': y_test, 'Predicted Tariff (RS)': y_pred})
print(predictions)
```

```
Mean Absolute Error (MAE): 5.329070518200751e-15
R-squared (R²): 1.0
    Actual Tariff (RS)  Predicted Tariff (RS)
0            24.862932              24.862932
17           23.325332              23.325332
```