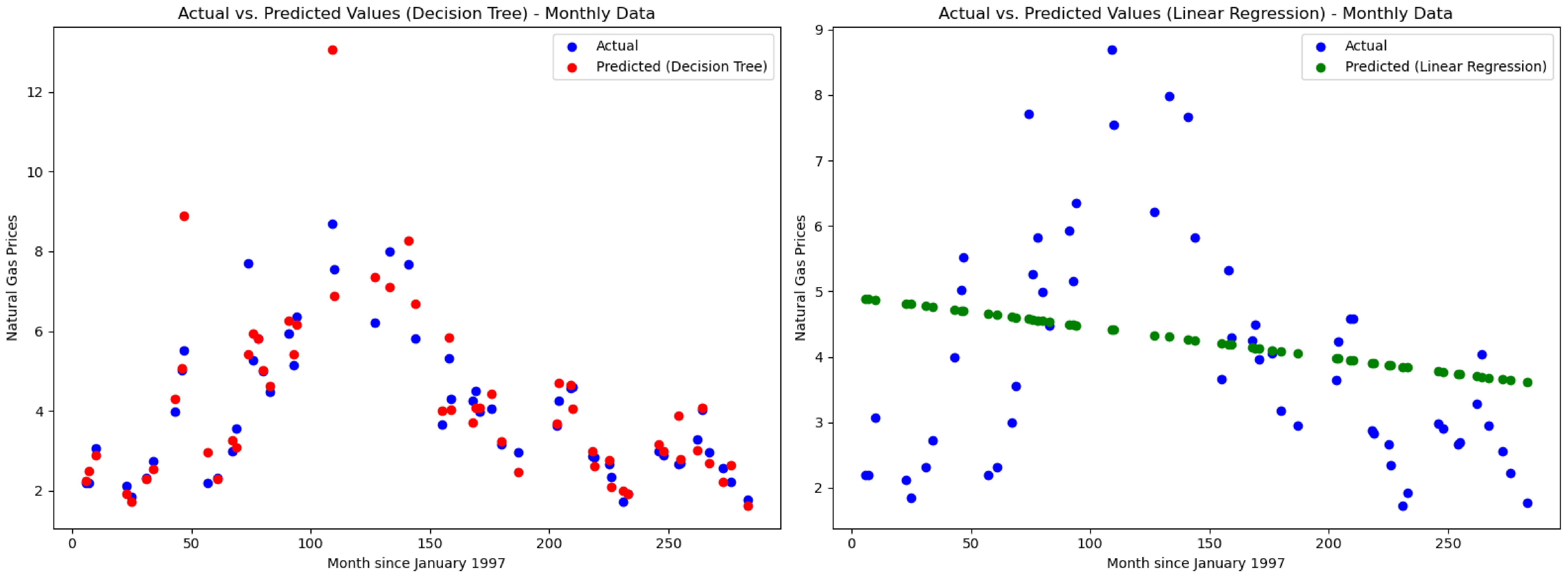
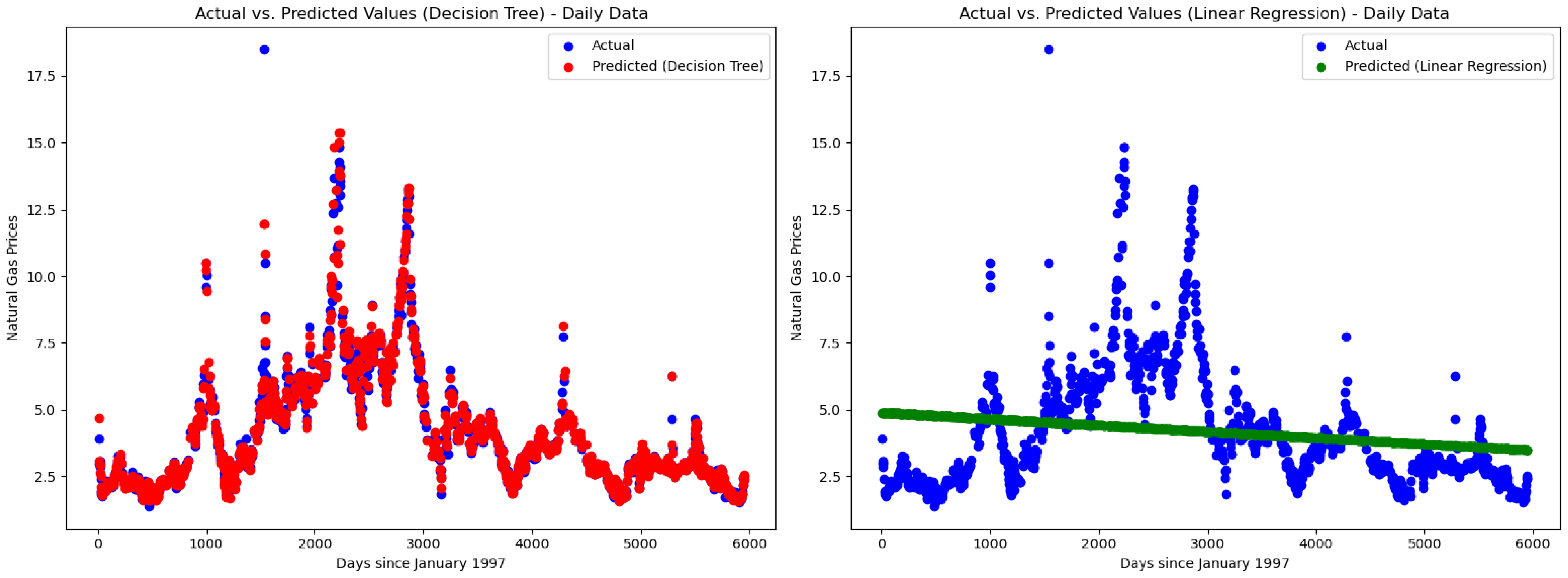
In [110…

```python
import numpy as np
from sklearn.tree import DecisionTreeRegressor
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error
import pandas as pd
import matplotlib.pyplot as plt

# Read the monthly data
df_monthly = pd.read_csv('monthly_csv.csv')

# Drop the 'Month' column
df_monthly = df_monthly.drop(columns=['Month'])

# Create a pandas Series from 1 to 284
X_monthly = pd.Series(np.arange(1, 285))

# Reshape X to a 2D array
X_reshaped_monthly = X_monthly.values.reshape(-1, 1)

# Selecting the 'Price' column as the target variable (y)
y_monthly = df_monthly['Price']

# Split the monthly data
X_train_monthly, X_test_monthly, y_train_monthly, y_test_monthly = train_test_split(
    X_reshaped_monthly, y_monthly, test_size=0.2, random_state=42
)

# Create and fit the DecisionTreeRegressor model for monthly regression
decision_tree_model_monthly = DecisionTreeRegressor()
decision_tree_model_monthly.fit(X_train_monthly, y_train_monthly)
tree_predictions_monthly = decision_tree_model_monthly.predict(X_test_monthly)
tree_mae_monthly = mean_absolute_error(y_test_monthly, tree_predictions_monthly)
print("Mean Absolute Error (Decision Tree) - Monthly Data:", tree_mae_monthly)

# Create and fit the Linear Regression model for monthly regression
linear_model_monthly = LinearRegression()
linear_model_monthly.fit(X_train_monthly, y_train_monthly)
linear_predictions_monthly = linear_model_monthly.predict(X_test_monthly)
linear_mae_monthly = mean_absolute_error(y_test_monthly, linear_predictions_monthly)
print("Mean Absolute Error (Linear Regression) - Monthly Data:", linear_mae_monthly)

# Plot actual vs. predicted values for both models - Monthly Data
plt.figure(figsize=(16, 6))

# Plot for Decision Tree - Monthly Data
plt.subplot(1, 2, 1)
plt.scatter(X_test_monthly, y_test_monthly, label='Actual', color='blue')
plt.scatter(X_test_monthly, tree_predictions_monthly, label='Predicted (Decision Tree)', color='red')
plt.xlabel('Month since January 1997')
plt.ylabel('Natural Gas Prices')
plt.title('Actual vs. Predicted Values (Decision Tree) - Monthly Data')
plt.legend()

# Plot for Linear Regression - Monthly Data
plt.subplot(1, 2, 2)
plt.scatter(X_test_monthly, y_test_monthly, label='Actual', color='blue')
plt.scatter(X_test_monthly, linear_predictions_monthly, label='Predicted (Linear Regression)', color='green')
plt.xlabel('Month since January 1997')
plt.ylabel('Natural Gas Prices')
plt.title('Actual vs. Predicted Values (Linear Regression) - Monthly Data')
plt.legend()

plt.tight_layout()
plt.show()

# Read the daily data
df_daily = pd.read_csv('daily_csv.csv')

# Drop the 'Month' column
df_daily = df_daily.drop(columns=['Date'])

# Create a pandas Series from 1 to 5953
X_daily = pd.Series(np.arange(1, 5954))

# Reshape X to a 2D array
X_reshaped_daily = X_daily.values.reshape(-1, 1)

# Selecting the 'Price' column as the target variable (y)
y_daily = df_daily['Price']

# Handling missing values by removing rows with NaN in y
y_no_missing = y_daily.dropna()
X_no_missing = X_daily[y_no_missing.index]

# Split the data without missing values
X_train_no_missing, X_test_no_missing, y_train_no_missing, y_test_no_missing = train_test_split(
    X_no_missing.values.reshape(-1, 1), y_no_missing, test_size=0.2, random_state=42
)

# Create and fit the DecisionTreeRegressor model for daily regression without missing values in y
decision_tree_model_no_missing = DecisionTreeRegressor()
decision_tree_model_no_missing.fit(X_train_no_missing, y_train_no_missing)
tree_predictions_no_missing = decision_tree_model_no_missing.predict(X_test_no_missing)
tree_mae_no_missing = mean_absolute_error(y_test_no_missing, tree_predictions_no_missing)
print("Mean Absolute Error (Decision Tree) - Daily Data (No Missing Values):", tree_mae_no_missing)

# Create and fit the Linear Regression model for daily regression without missing values in y
linear_model_no_missing = LinearRegression()
linear_model_no_missing.fit(X_train_no_missing, y_train_no_missing)
linear_predictions_no_missing = linear_model_no_missing.predict(X_test_no_missing)
linear_mae_no_missing = mean_absolute_error(y_test_no_missing, linear_predictions_no_missing)
print("Mean Absolute Error (Linear Regression) - Daily Data (No Missing Values):", linear_mae_no_missing)

# Plot actual vs. predicted values for both models - Daily Data (No Missing Values)
plt.figure(figsize=(16, 6))

# Plot for Decision Tree - Daily Data (No Missing Values)
plt.subplot(1, 2, 1)
plt.scatter(X_test_no_missing, y_test_no_missing, label='Actual', color='blue')
plt.scatter(X_test_no_missing, tree_predictions_no_missing, label='Predicted (Decision Tree)', color='red')
plt.xlabel('Days since January 1997')
plt.ylabel('Natural Gas Prices')
plt.title('Actual vs. Predicted Values (Decision Tree) - Daily Data')
plt.legend()

# Plot for Linear Regression - Daily Data (No Missing Values)
plt.subplot(1, 2, 2)
plt.scatter(X_test_no_missing, y_test_no_missing, label='Actual', color='blue')
plt.scatter(X_test_no_missing, linear_predictions_no_missing, label='Predicted (Linear Regression)', color='green')
plt.xlabel('Days since January 1997')
plt.ylabel('Natural Gas Prices')
plt.title('Actual vs. Predicted Values (Linear Regression) - Daily Data')
plt.legend()

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

```
Mean Absolute Error (Decision Tree) - Monthly Data: 0.47333333333333333
Mean Absolute Error (Linear Regression) - Monthly Data: 1.3868502722473173
```



```
Mean Absolute Error (Decision Tree) - Daily Data (No Missing Values): 0.13120067170445004
Mean Absolute Error (Linear Regression) - Daily Data (No Missing Values): 1.6905562690088114
```



In [ ]:

In [ ]:

In [ ]: