SEASONAL ENERGY CONSUMPTION FORECASTING USING SARIMAX

Submitted by:

Name : Priyadharshini R

Register Number : 727723EUCS175

Class : CSE – ‘C’

Year : III

Department : Computer Science and Engineering

1. Abstract

Electricity demand forecasting plays a crucial role in power distribution planning and resource optimization. Energy consumption data typically exhibits strong seasonal patterns due to recurring human activities and environmental factors. This project focuses on forecasting monthly electricity consumption using the Seasonal AutoRegressive Integrated Moving Average (SARIMA) model. Historical household energy consumption data is analyzed, preprocessed, and aggregated to monthly frequency. The SARIMA model is applied to capture trend, seasonality, and temporal dependence in the data. Forecast accuracy is evaluated using statistical error metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). The results demonstrate the effectiveness of seasonal time series modeling for energy demand forecasting.

2. Problem Statement

A power distribution company aims to accurately predict monthly electricity consumption to ensure adequate supply, prevent power outages, and optimize energy distribution. Electricity usage shows strong seasonal variations and temporal dependencies. Ignoring seasonality leads to inaccurate forecasts and inefficient resource planning. Hence, a reliable forecasting model that captures seasonal behavior is essential for effective decision-making in the energy sector.

3. Objectives

- To analyze historical electricity consumption data
- To identify seasonal patterns in energy usage
- To build a SARIMA model for monthly electricity consumption forecasting
- To evaluate forecasting accuracy using statistical metrics
- To validate model performance through residual analysis

4. Scope of the Project

This project focuses on medium-term electricity demand forecasting using classical statistical time series techniques. The scope includes data preprocessing, seasonal modeling, forecast evaluation, and interpretation of results. The study is limited to household-level electricity consumption and does not consider real-time or large-scale grid-level forecasting.

5. Dataset Description

The project uses the Household Energy Consumption dataset obtained from Kaggle. The dataset contains minute-level electricity usage records for a residential household over multiple years.

Dataset Characteristics:

- File format: TXT (semicolon-separated)
- Time granularity: Minute-level
- Duration: Multiple years
- Target variable: Global Active Power

The dataset was aggregated to monthly energy consumption to capture long-term seasonal trends suitable for SARIMA modeling.

6. Data Preprocessing

The following preprocessing steps were applied:

1. Combined the Date and Time columns into a single DateTime column
2. Handled missing values represented by “?”
3. Converted power consumption values to numeric format
4. Removed invalid and missing records
5. Resampled minute-level data to monthly frequency
6. Converted average power values into approximate monthly energy consumption (kWh)

These steps ensured clean, consistent, and analyzable time series data.

7. Exploratory Data Analysis

Exploratory analysis was performed to understand the structure and behavior of the data. Time series visualization of monthly electricity consumption revealed:

- Clear annual seasonality
- Fluctuating demand levels across months
- Recurring consumption patterns

A seasonal boxplot grouped by month further confirmed consistent seasonal behavior, validating the need for a seasonal forecasting model.

8. Stationarity Analysis

The Augmented Dickey-Fuller (ADF) test was conducted to assess stationarity.

- Null Hypothesis: The time series is non-stationary
- Alternative Hypothesis: The time series is stationary

The test results indicated non-stationarity, necessitating differencing. Both non-seasonal and seasonal differencing were incorporated within the SARIMA framework.

9. Methodology

The Seasonal ARIMA (SARIMA) model was selected due to its ability to capture trend, seasonality, and autocorrelation.

Model Configuration:

- Non-seasonal order (p, d, q): (1, 1, 1)
- Seasonal order (P, D, Q, s): (1, 1, 1, 12)

Here, the seasonal period $s = 12$corresponds to yearly seasonality in monthly data. The model was trained on historical data, with the final 12 months reserved for testing.

10. Model Training and Forecasting

The SARIMA model was fitted using the training dataset. After model estimation, electricity consumption was forecasted for the next 12 months. Forecast confidence intervals were generated to quantify uncertainty. Visualization of training data, test data, and forecasted values showed that the model effectively followed observed seasonal trends.

11. Model Evaluation

The forecasting performance was evaluated using the following metrics:

- Mean Absolute Error (MAE)
  Measures the average absolute difference between predicted and actual values.
- Root Mean Squared Error (RMSE)
  Penalizes larger errors more heavily and highlights significant deviations.

Lower MAE and RMSE values indicate better model accuracy.

12. Residual Analysis

Residual diagnostics were performed manually due to limited observations after seasonal differencing. The following analyses were conducted:

- Residuals over time plot showed random fluctuations around zero
- Histogram of residuals indicated approximate normal distribution
- Q–Q plot confirmed residual normality
- Forecast error plot revealed no systematic bias

These results validate the adequacy of the SARIMA model.

13. Results and Discussion

The SARIMA model successfully captured the seasonal behavior of electricity consumption. The forecasts closely matched observed values during the testing period. Error metrics indicated satisfactory accuracy, demonstrating the suitability of seasonal time series models for energy demand forecasting.

14. Business Impact and Practical Relevance

Accurate electricity demand forecasting enables power utilities to:

- Prevent power shortages and outages
- Optimize energy generation and distribution
- Reduce operational and maintenance costs
- Improve long-term planning and sustainability initiatives

The SARIMA-based approach provides a cost-effective and interpretable forecasting solution.

15. Assumptions Made in the Study

- Historical consumption patterns remain consistent over time
- Seasonal behavior does not change significantly across years
- External influences are implicitly captured in past consumption data
- Data quality after preprocessing is sufficient for modeling

16. Comparison with Other Forecasting Techniques

Model	Strengths	Weaknesses
ARIMA	Simple and interpretable	No seasonality
SARIMA	Captures seasonality	No exogenous variables
SARIMAX	Includes external factors	Requires additional data
LSTM	Handles complex patterns	Data-intensive
Prophet	Robust and easy to use	Less flexible tuning

20. Conclusion

This project demonstrates the effectiveness of SARIMA models for forecasting seasonal electricity consumption. By capturing temporal dependencies and recurring seasonal patterns, the model provides reliable forecasts that can aid energy planning and management. Seasonal time series forecasting is a valuable tool for the energy and utilities sector.

Source Code:

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import scipy.stats as stats


from statsmodels.tsa.statespace.sarimax import SARIMAX

from statsmodels.tsa.stattools import adfuller

from sklearn.metrics import mean_absolute_error, mean_squared_error


import warnings

warnings.filterwarnings("ignore")


df = pd.read_csv("dataset.txt", sep=";", na_values="?")


df["Datetime"] = pd.to_datetime(

   df["Date"] + " " + df["Time"],

   format="%d/%m/%Y %H:%M:%S"

)


df.set_index("Datetime", inplace=True)

df.drop(["Date", "Time"], axis=1, inplace=True)


df["Global_active_power"] = pd.to_numeric(
```

```
    df["Global_active_power"],
    errors="coerce"
)

df = df.dropna()

monthly_df = df["Global_active_power"].resample("M").mean()
monthly_df = monthly_df * 30 * 24

plt.figure(figsize=(10,4))
plt.plot(monthly_df)
plt.title("Monthly Electricity Consumption")
plt.xlabel("Date")
plt.ylabel("Energy Consumption (kWh)")
plt.grid(alpha=0.3)
plt.show()

adf_stat, p_value, *_ = adfuller(monthly_df)
print("ADF Statistic:", adf_stat)
print("p-value:", p_value)

train = monthly_df.iloc[:-12]
test = monthly_df.iloc[-12:]

model = SARIMAX(
    train,
    order=(1,1,1),
    seasonal_order=(1,1,1,12),
    enforce_stationarity=False,
```

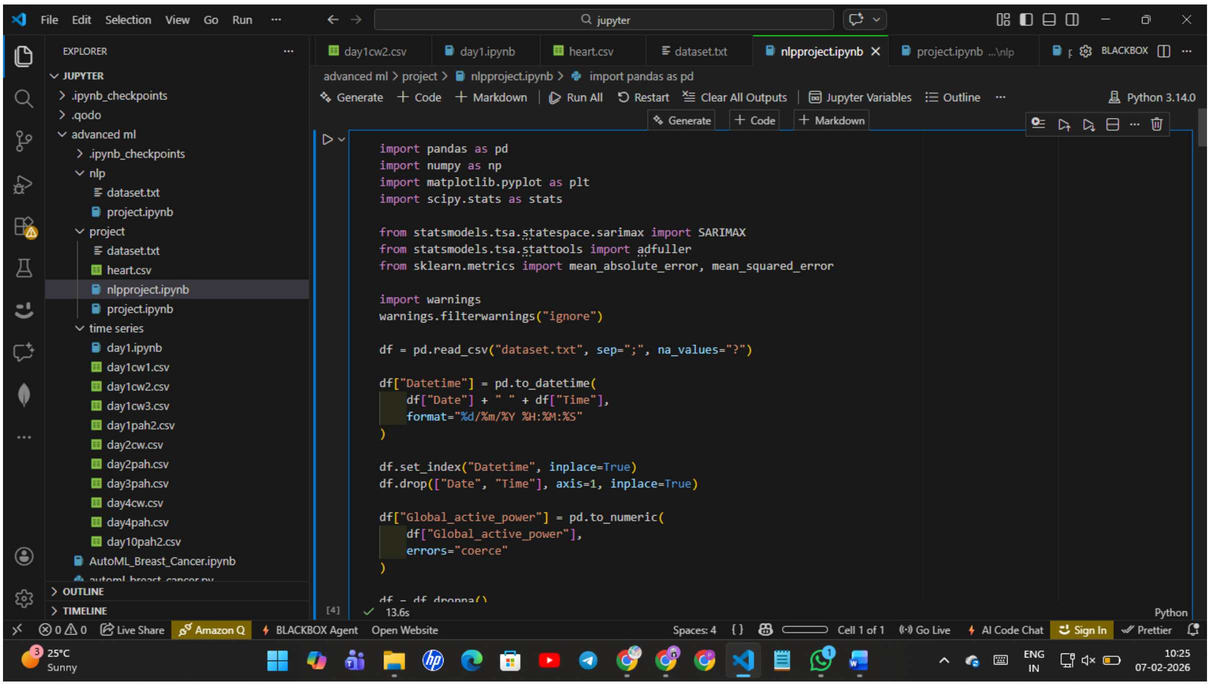
```
    enforce_invertibility=False
)

results = model.fit()
print(results.summary())

fitted = results.fittedvalues

plt.figure(figsize=(10,5))
plt.plot(train, label="Actual")
plt.plot(fitted, linestyle="--", label="Fitted")
plt.title("Actual vs Fitted Values (SARIMA)")
plt.xlabel("Date")
plt.ylabel("Energy Consumption")
plt.legend()
plt.grid(alpha=0.3)
plt.show()

forecast = results.get_forecast(steps=12)
pred = forecast.predicted_mean
conf_int = forecast.conf_int()

mae = mean_absolute_error(test, pred)
rmse = np.sqrt(mean_squared_error(test, pred))

print("MAE:", mae)
print("RMSE:", rmse)

plt.figure(figsize=(11,5))
```

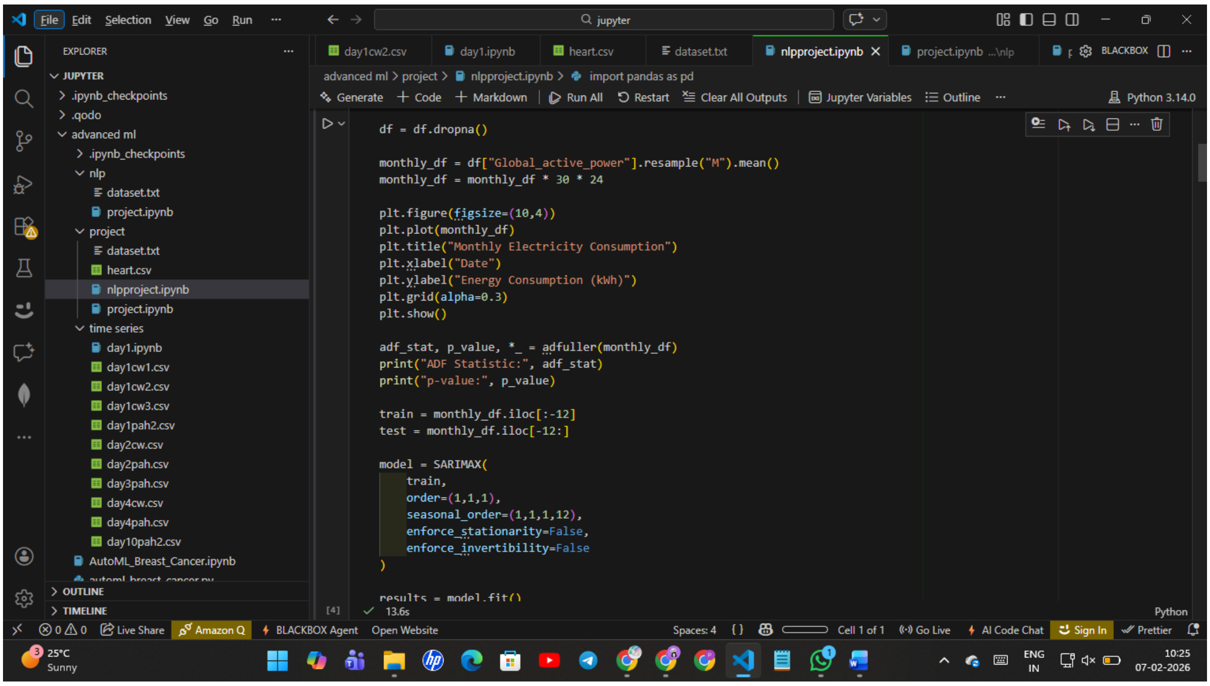
```
plt.plot(train, label="Training Data")
plt.plot(test, label="Test Data")
plt.plot(pred, linestyle="--", label="Forecast")
plt.fill_between(
    conf_int.index,
    conf_int.iloc[:,0],
    conf_int.iloc[:,1],
    alpha=0.3
)
plt.title("Train–Test–Forecast Comparison")
plt.xlabel("Date")
plt.ylabel("Energy Consumption")
plt.legend()
plt.grid(alpha=0.3)
plt.show()

monthly_df_df = monthly_df.to_frame(name="Consumption")
monthly_df_df["Month"] = monthly_df_df.index.month

plt.figure(figsize=(10,5))
monthly_df_df.boxplot(column="Consumption", by="Month")
plt.title("Monthly Seasonality in Energy Consumption")
plt.suptitle("")
plt.xlabel("Month")
plt.ylabel("Energy Consumption")
plt.grid(alpha=0.3)
plt.show()

residuals = results.resid
```

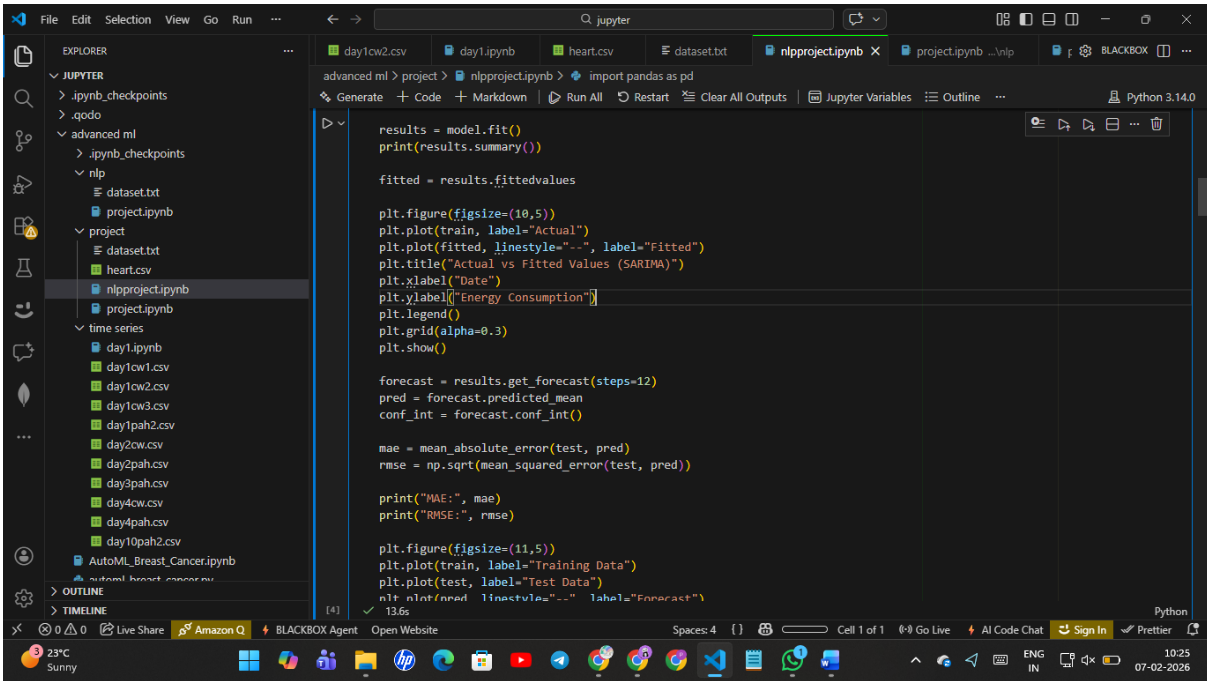
```
plt.figure(figsize=(10,4))
plt.plot(residuals)
plt.title("Residuals Over Time")
plt.xlabel("Date")
plt.ylabel("Residual")
plt.grid(alpha=0.3)
plt.show()

plt.figure(figsize=(6,4))
plt.hist(residuals, bins=20)
plt.title("Residual Distribution")
plt.xlabel("Residual")
plt.ylabel("Frequency")
plt.grid(alpha=0.3)
plt.show()

plt.figure(figsize=(6,6))
stats.probplot(residuals, dist="norm", plot=plt)
plt.title("Q–Q Plot of Residuals")
plt.grid(alpha=0.3)
plt.show()

forecast_error = test - pred

plt.figure(figsize=(10,4))
plt.plot(forecast_error, marker="o")
plt.axhline(0, linestyle="--")
plt.title("Forecast Error Over Time")
```

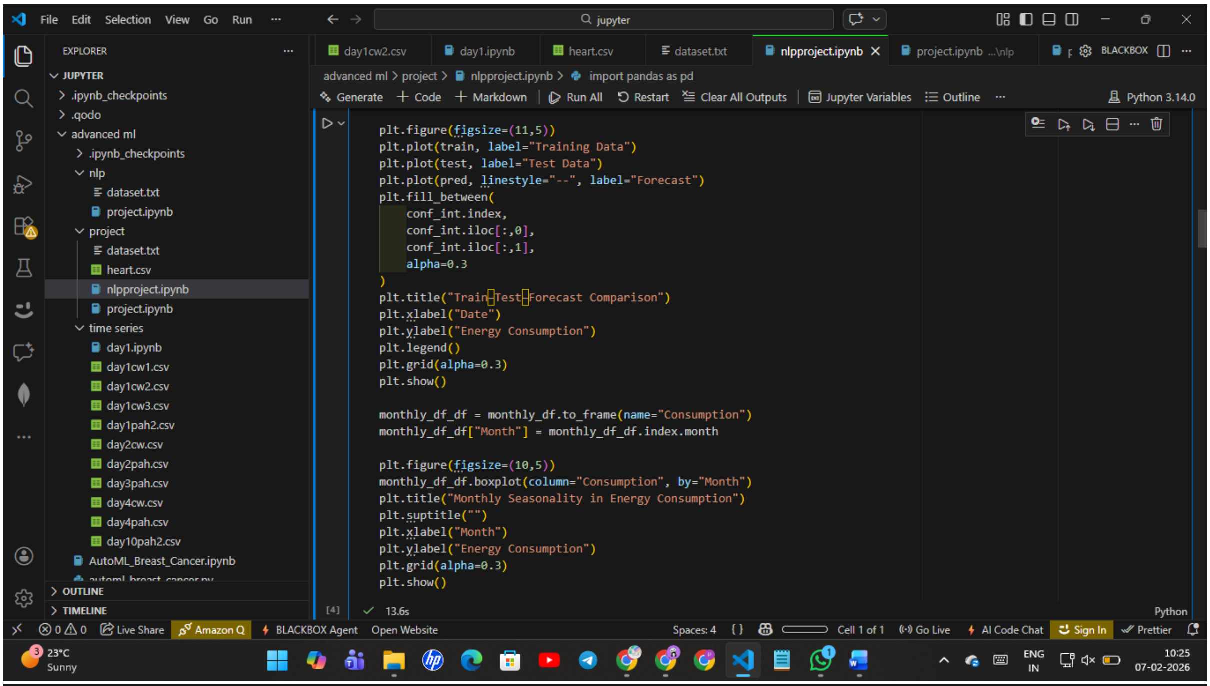
```
plt.xlabel("Date")

plt.ylabel("Error")

plt.grid(alpha=0.3)

plt.show()
```

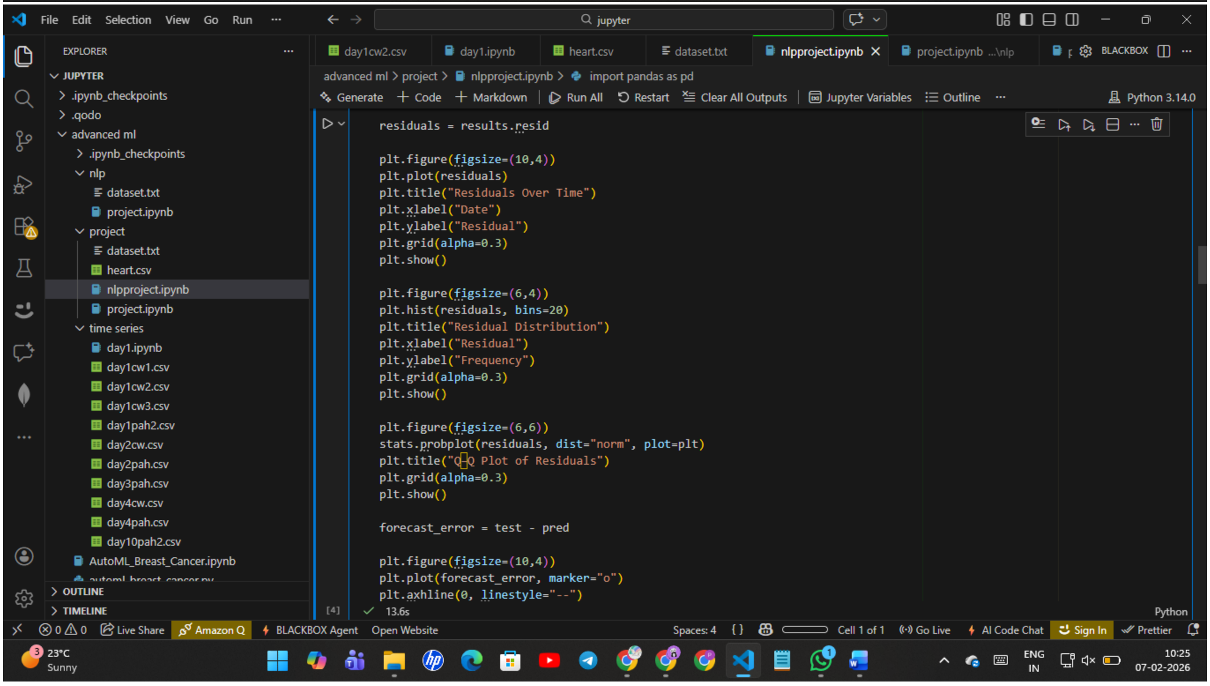
Screenshots:

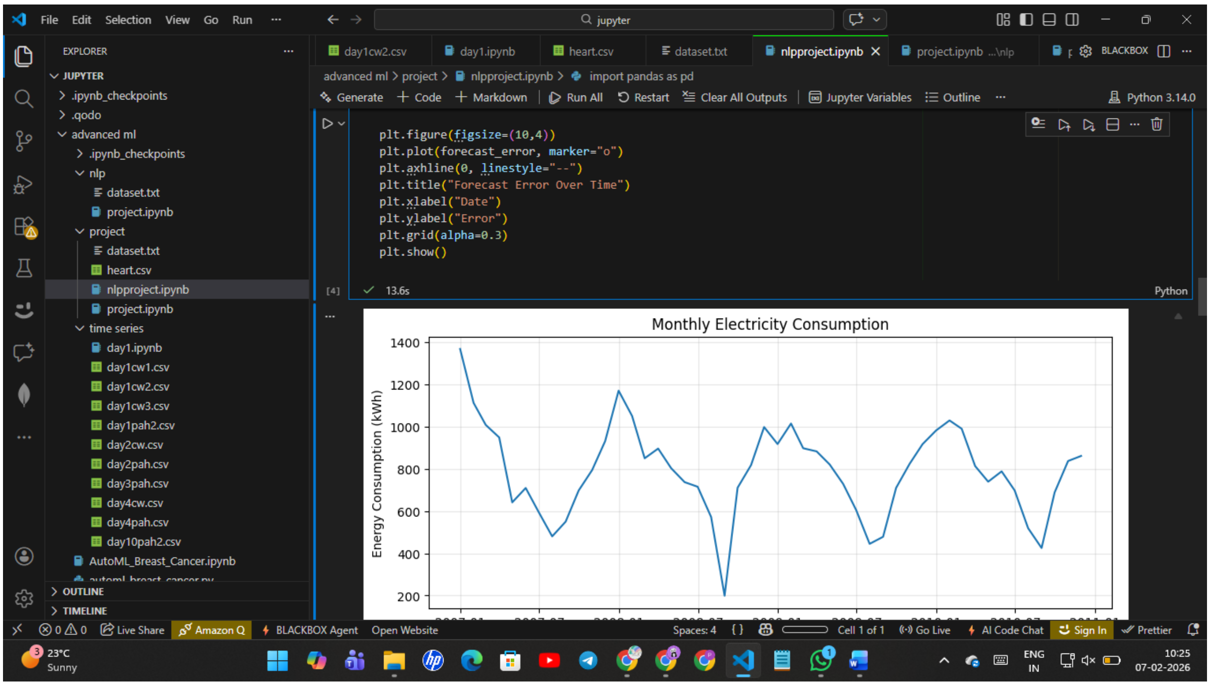


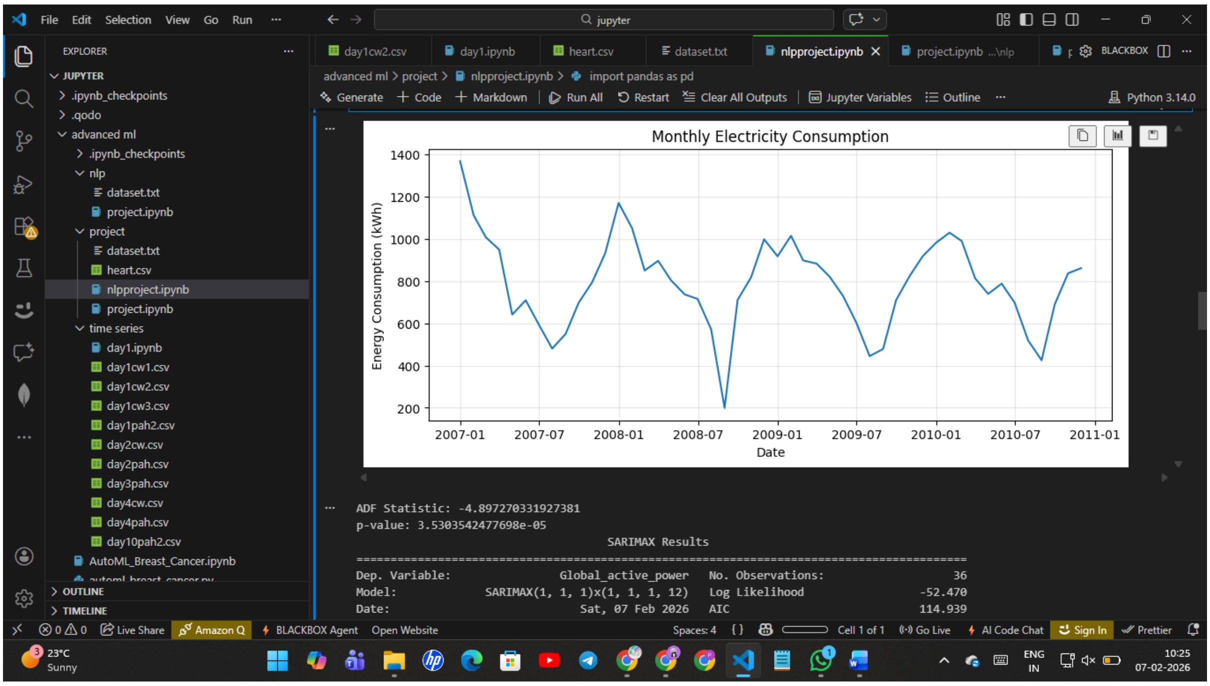

df = df.dropna()
monthly_df = df["Global_active_power"].resample("M").mean()
monthly_df = monthly_df * 30 * 24
plt.figure(figsize=(10,4))
plt.plot(monthly_df)
plt.title("Monthly Electricity Consumption")
plt.xlabel("Date")
plt.ylabel("Energy Consumption (kWh)")
plt.grid(alpha=0.3)
plt.show()
adf_stat, p_value, *_ = adfuller(monthly_df)
print("ADF Statistic:", adf_stat)
print("p-value:", p_value)
train = monthly_df.iloc[:-12]
test = monthly_df.iloc[-12:]
model = SARIMAX(
train,
order=(1,1,1),
seasonal_order=(1,1,1,12),
enforce_stationarity=False,
enforce_invertibility=False
)

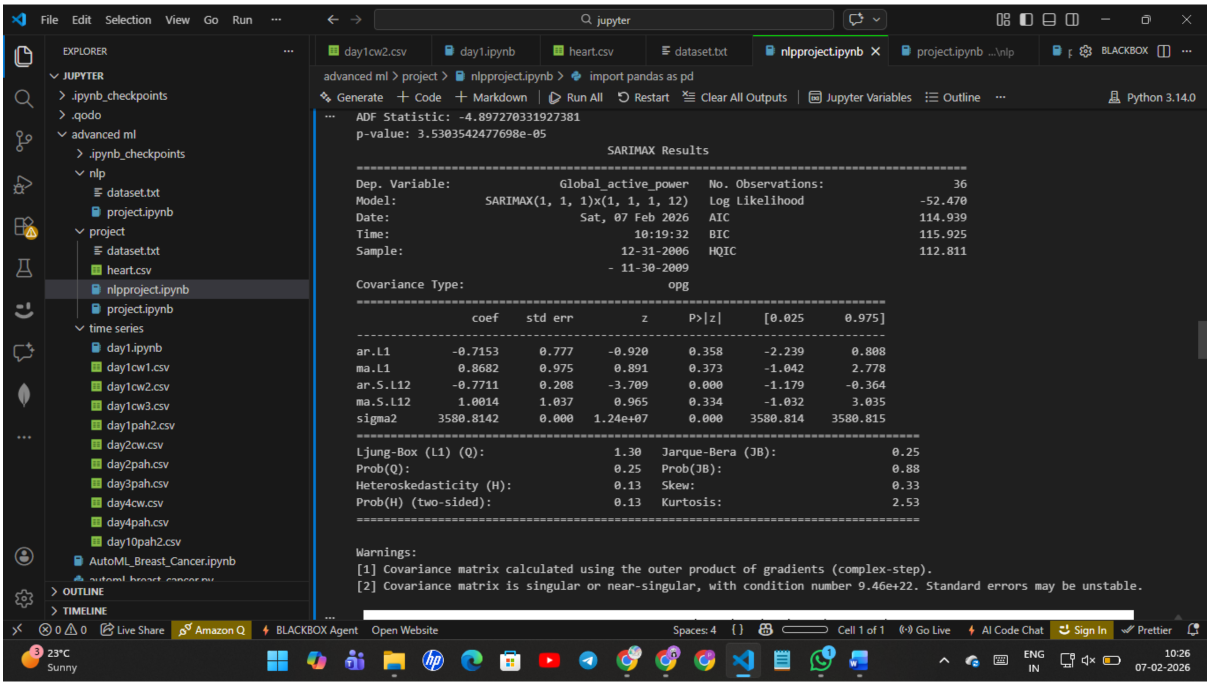

results = model.fit()
print(results.summary())
fitted = results.fittedvalues
plt.figure(figsize=(10,5))
plt.plot(train, label="Actual")
plt.plot(fitted, linestyle="--", label="Fitted")
plt.title("Actual vs Fitted Values (SARIMA)")
plt.xlabel("Date")
plt.ylabel("Energy Consumption")
plt.legend()
plt.grid(alpha=0.3)
plt.show()
forecast = results.get_forecast(steps=12)
pred = forecast.predicted_mean
conf_int = forecast.conf_int()
mae = mean_absolute_error(test, pred)
rmse = np.sqrt(mean_squared_error(test, pred))
print("MAE:", mae)
print("RMSE:", rmse)
plt.figure(figsize=(11,5))
plt.plot(train, label="Training Data")
plt.plot(test, label="Test Data")

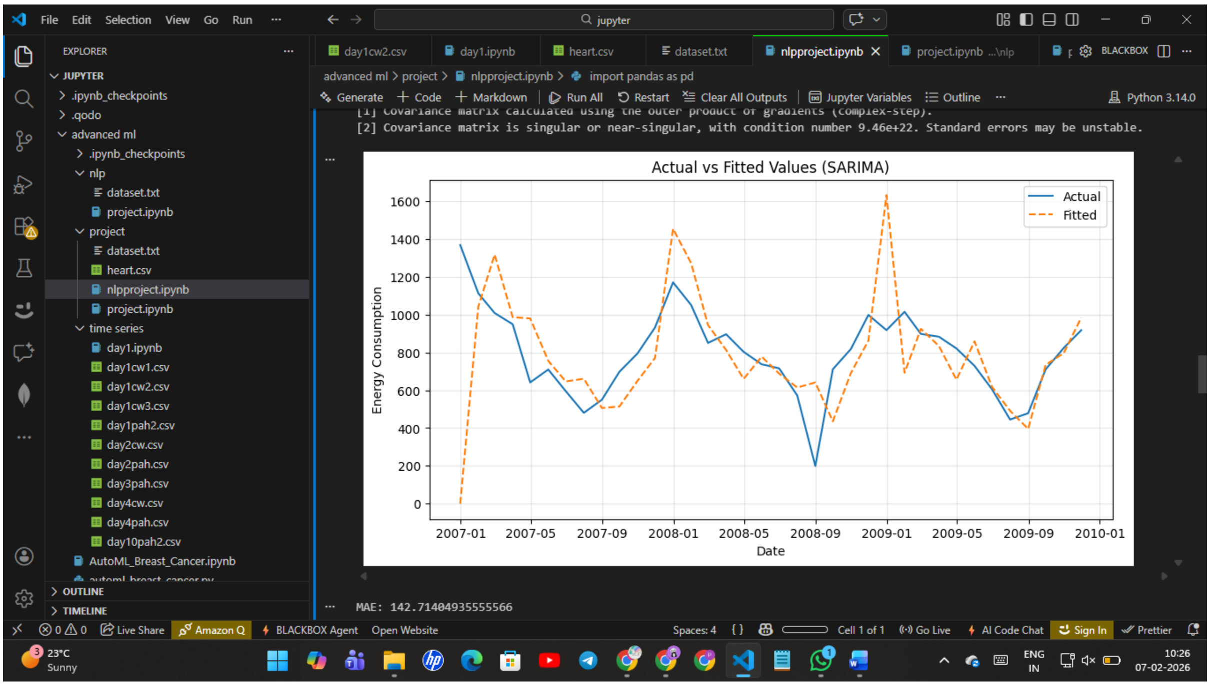

plt.figure(figsize=(11,5))
plt.plot(train, label="Training Data")
plt.plot(test, label="Test Data")
plt.plot(pred, linestyle="--", label="Forecast")
plt.fill_between(
conf_int.index,
conf_int.iloc[:,0],
conf_int.iloc[:,1],
alpha=0.3
)
plt.title("Train–Test–Forecast Comparison")
plt.xlabel("Date")
plt.ylabel("Energy Consumption")
plt.legend()
plt.grid(alpha=0.3)
plt.show()
monthly_df_df = monthly_df.to_frame(name="Consumption")
monthly_df_df["Month"] = monthly_df_df.index.month
plt.figure(figsize=(10,5))
monthly_df_df.boxplot(column="Consumption", by="Month")
plt.title("Monthly Seasonality in Energy Consumption")
plt.suptitle("")
plt.xlabel("Month")
plt.ylabel("Energy Consumption")
plt.grid(alpha=0.3)
plt.show()

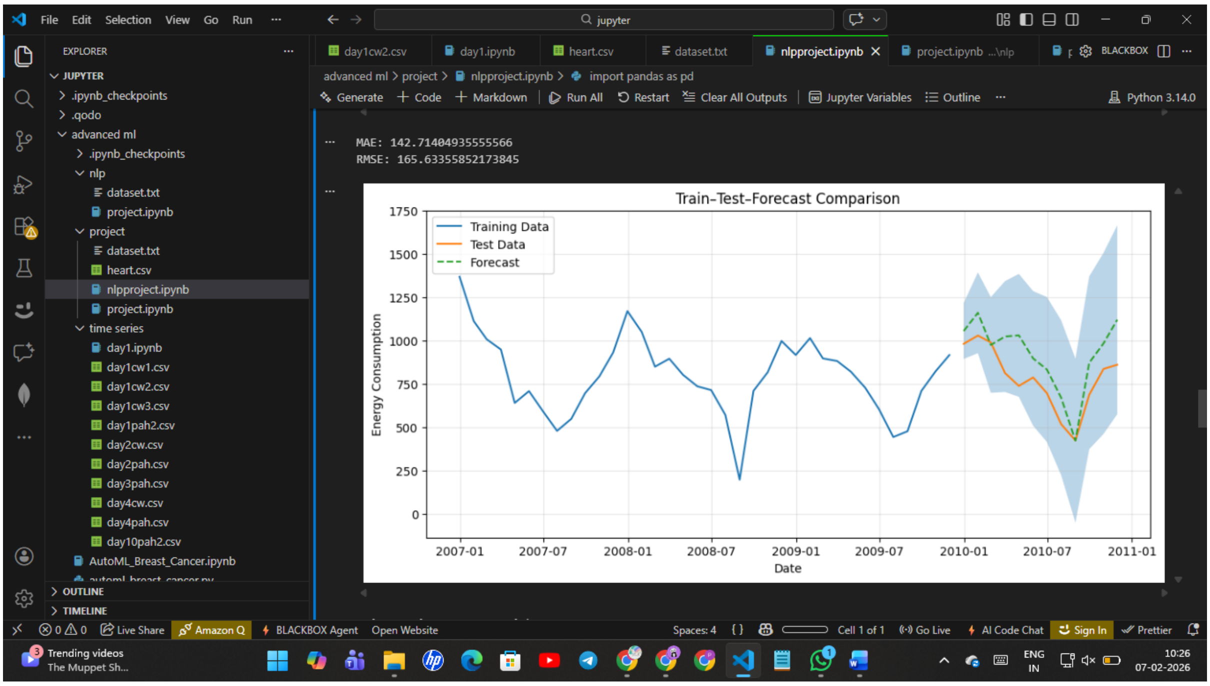

residuals = results.resid
plt.figure(figsize=(10,4))
plt.plot(residuals)
plt.title("Residuals Over Time")
plt.xlabel("Date")
plt.ylabel("Residual")
plt.grid(alpha=0.3)
plt.show()
plt.figure(figsize=(6,4))
plt.hist(residuals, bins=20)
plt.title("Residual Distribution")
plt.xlabel("Residual")
plt.ylabel("Frequency")
plt.grid(alpha=0.3)
plt.show()
plt.figure(figsize=(6,6))
stats.probplot(residuals, dist="norm", plot=plt)
plt.title("Q–Q Plot of Residuals")
plt.grid(alpha=0.3)
plt.show()
forecast_error = test - pred
plt.figure(figsize=(10,4))
plt.plot(forecast_error, marker="o")
plt.axhline(0, linestyle="--")

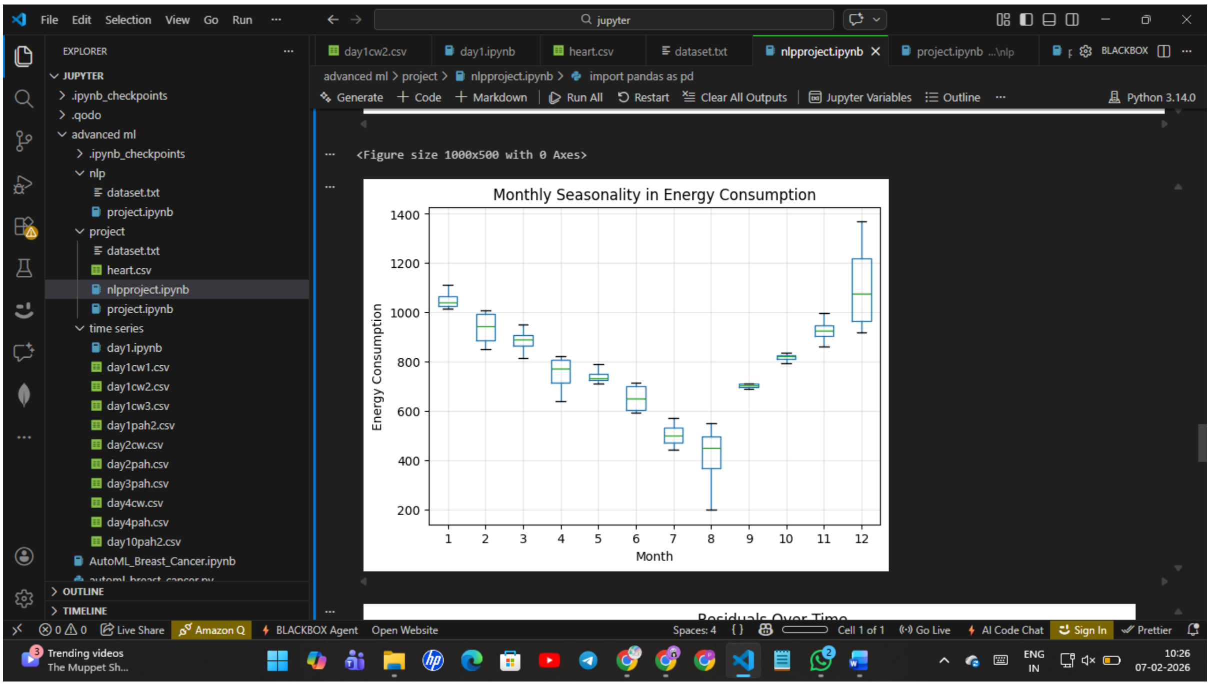

plt.figure(figsize=(10,4))
plt.plot(forecast_error, marker="o")
plt.axhline(0, linestyle="--")
plt.title("Forecast Error Over Time")
plt.xlabel("Date")
plt.ylabel("Error")
plt.grid(alpha=0.3)
plt.show()
Monthly Electricity Consumption
Energy Consumption (kWh)

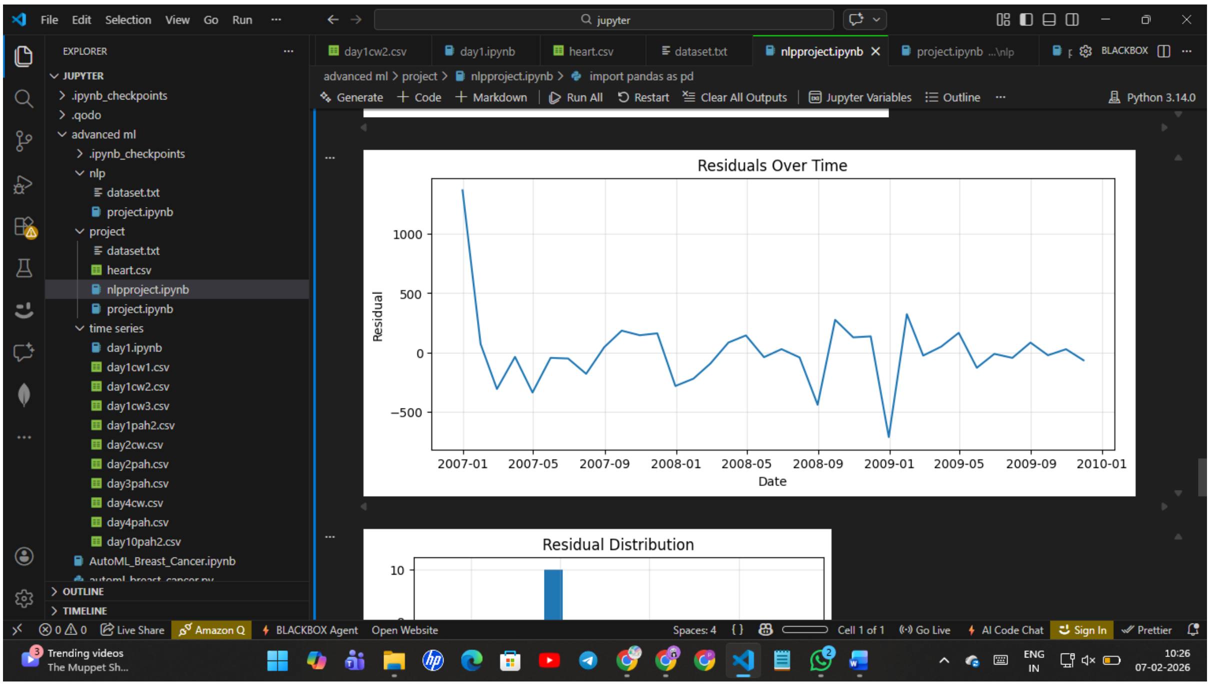

Monthly Electricity Consumption
Energy Consumption (kWh)
Date
ADF Statistic: -4.897270331927381
p-value: 3.5303542477698e-05
SARIMAX Results
Dep. Variable: Global_active_power No. Observations: 36
Model: SARIMAX(1, 1, 1)x(1, 1, 1, 12) Log Likelihood -52.470
Date: Sat, 07 Feb 2026 AIC 114.939

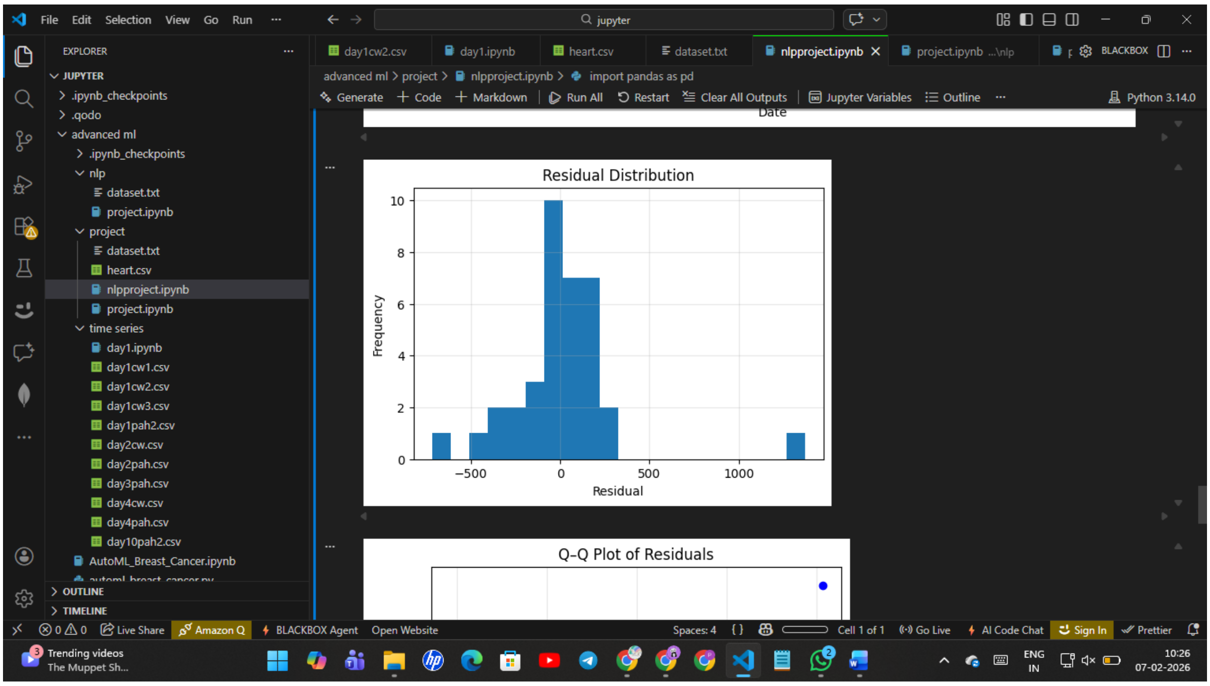

ADF Statistic: -4.897270331927381
p-value: 3.5303542477698e-05
SARIMAX Results
Dep. Variable: Global_active_power No. Observations: 36
Model: SARIMAX(1, 1, 1)x(1, 1, 1, 12) Log Likelihood -52.470
Date: Sat, 07 Feb 2026 AIC 114.939
Time: 10:19:32 BIC 115.925
Sample: 12-31-2006 HQIC 112.811
- 11-30-2009
Covariance Type: opg
coef std err z P>|z| [0.025 0.975]
ar.L1 -0.7153 0.777 -0.920 0.358 -2.239 0.808
ma.L1 0.8682 0.975 0.891 0.373 -1.042 2.778
ar.S.L12 -0.7711 0.208 -3.709 0.000 -1.179 -0.364
ma.S.L12 1.0014 1.037 0.965 0.334 -1.032 3.035
sigma2 3580.8142 0.000 1.24e+07 0.000 3580.814 3580.815
Ljung-Box (L1) (Q): 1.30 Jarque-Bera (JB): 0.25
Prob(Q): 0.25 Prob(JB): 0.88
Heteroskedasticity (H): 0.13 Skew: 0.33
Prob(H) (two-sided): 0.13 Kurtosis: 2.53
Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
[2] Covariance matrix is singular or near-singular, with condition number 9.46e+22. Standard errors may be unstable.

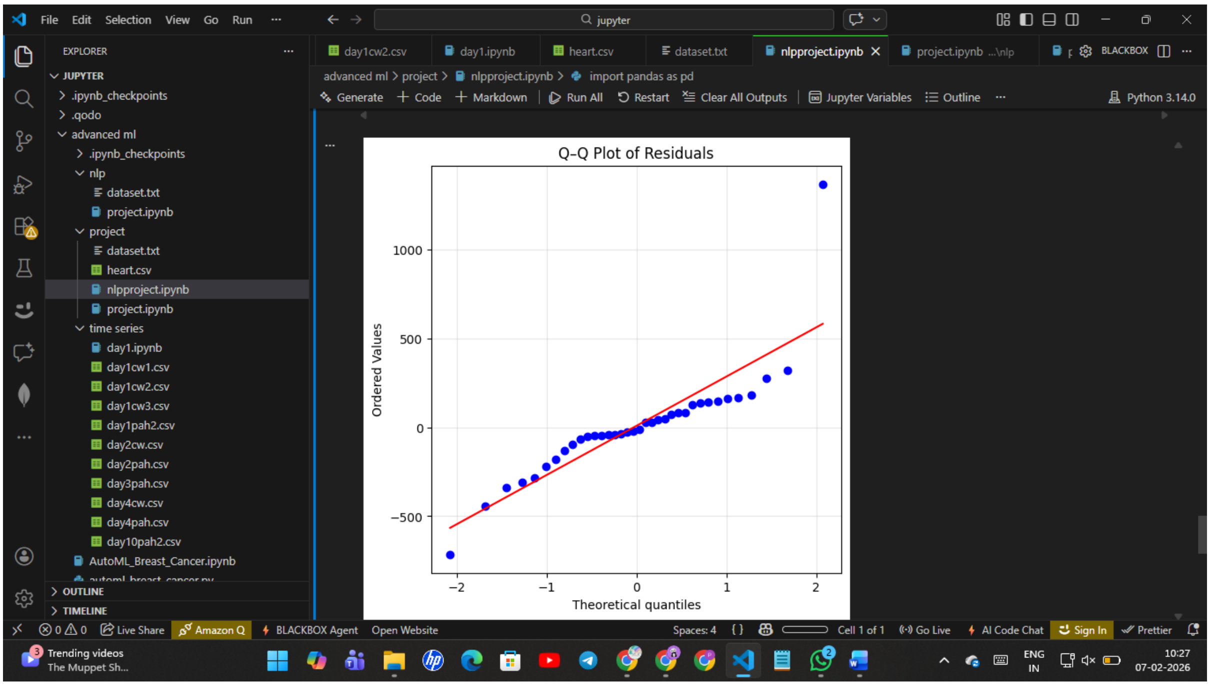

[2] Covariance matrix is singular or near-singular, with condition number 9.46e+22. Standard errors may be unstable.
Actual vs Fitted Values (SARIMA)
Actual
Fitted
Energy Consumption
Date
MAE: 142.71404935555566

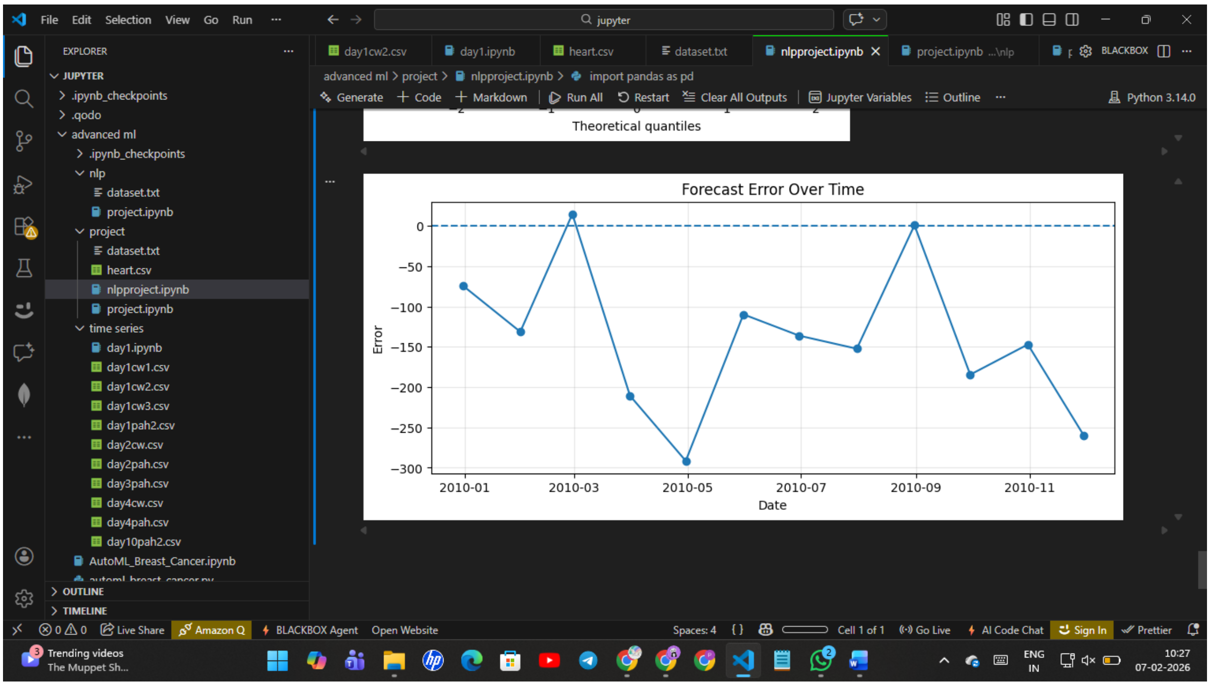

MAE: 142.71404935555566
RMSE: 165.63355852173845
Train-Test-Forecast Comparison
Training Data
Test Data
Forecast
Energy Consumption
Date


<Figure size 1000x500 with 0 Axes>
Monthly Seasonality in Energy Consumption
Energy Consumption
Month


File Edit Selection View Go Run ...
jupyter
EXPLORER
JUPYTER
advanced ml > project > nlpproject.ipynb > import pandas as pd
Generate + Code + Markdown Run All Restart Clear All Outputs Jupyter Variables Outline
Python 3.14.0
Residuals Over Time
Residual
1000
500
0
−500
2007-01 2007-05 2007-09 2008-01 2008-05 2008-09 2009-01 2009-05 2009-09 2010-01
Date
Residual Distribution
10


File Edit Selection View Go Run ...
jupyter
EXPLORER
JUPYTER
advanced ml > project > nlpproject.ipynb > import pandas as pd
Generate + Code + Markdown Run All Restart Clear All Outputs Jupyter Variables Outline
Python 3.14.0
Residual Distribution
Frequency
10
8
6
4
2
0
−500 0 500 1000
Residual
Q–Q Plot of Residuals


Q-Q Plot of Residuals
Ordered Values
Theoretical quantiles
1000
500
0
−500
−2
−1
0
1
2


Theoretical quantiles
Forecast Error Over Time
Error
Date
0
−50
−100
−150
−200
−250
−300
2010-01
2010-03
2010-05
2010-07
2010-09
2010-11