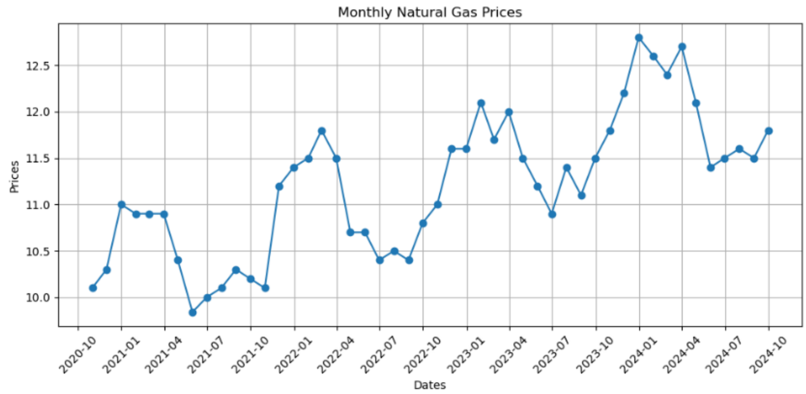
Nat Gas Report

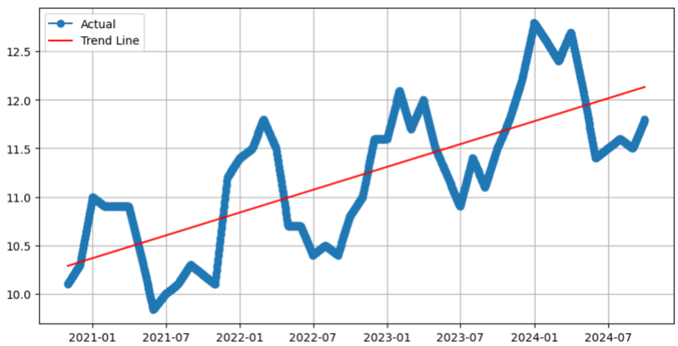
The project involved working with a dataset of natural gas prices from 2020 to 2024, where only the last two days of each month had recorded prices. The goal was to fill in missing prices for the days between recorded data and to predict future prices, so that an estimated gas price could be provided for any given date.

The first step was to download and import the CSV file, read its contents, and extract basic information about the dataset. The date column was converted from object to datetime format and sorted chronologically. Summary statistics were generated to understand the minimum, maximum, and percentile values of the prices.

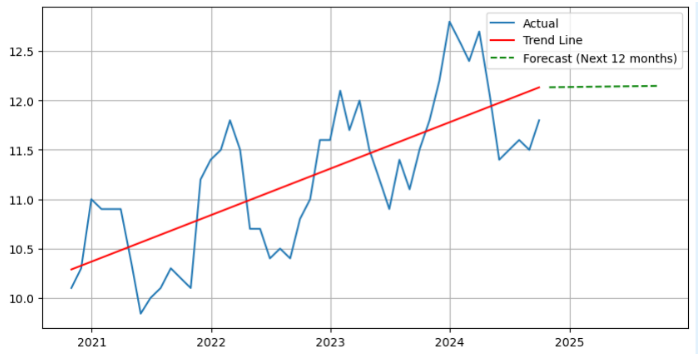
Visualisation was then used to analyse trends in the monthly gas prices. Using matplotlib, recurring patterns were observed, such as prices increasing at the end of each month. This pattern suggests that gas demand rises during the winter months, likely due to increased heating and other seasonal needs. Additional plots also revealed an overall upward trend in prices over time, reflecting general market growth and seasonal fluctuations.



To generate daily estimates, the date column was set as the index, and interpolation was applied to fill in prices for all days of each month, not just the last two. For forecasting, a numerical time index was created, as NumPy cannot directly operate on datetime values. A simple linear regression model was fitted to predict future prices. While the model captured the general trend, predicted prices deviated from actual values due to seasonal variations.



The red trend line in the plot shows the model's predicted prices. While it follows the overall upward trend, it does not reflect the seasonal variations seen in the actual prices, which rise and fall depending on the month.



And it also forecasted for the upcoming months in the same manner as for the previous months.

Validating the evaluation involved inputting specific dates to see the predicted prices. For example, predictions for July and November returned $10.36 and $10.37, respectively, showing minimal difference, whereas actual prices vary significantly between these months. Interpolation produced more accurate results ($10.42 for July and $11.10 for November), confirming that the initial model failed to capture seasonality. The R^2 score was 0.53, indicating that the model explained only 53% of the variability in the data. R^2 score was also checked, which was 0.53, although it captures 53 percentage of the data variability, there is still 47% of the data still needs to be captured by the model.

To improve accuracy, the linear regression model was modified to incorporate both trend and seasonality. After retraining, predictions for the same months yielded $9.50 for July and $10.28 for November, capturing seasonal effects more accurately. The R^2 score also improved to 0.942, demonstrating a much stronger fit to the data.