Time Series Regression Assignment

Ryoji Takahashi

December 2, 2024

1 Description of Assignments and Objectives

The time series regression assignments are:

1. Choose an open multivariate time series dataset that is not financial and where the variable to be predicted does not have seasonality (reference source).
2. Check that the time series is non-seasonal
3. Select, train and test the machine learning/time series model(s) that allow: Predicting 100 time periods in the future with the particularity that from the moment where the prediction begins, the value of the regressor variables are not available
4. Graphically represent the results and evaluate the goodness of the model using the appropriate KPIs
5. Draw conclusions from the exercise carried out.
6. Prepare a deliverable with all the necessary files to reproduce the analysis and put the trained model(s) into production (DevOps integration).

1.1 Dataset

1. The dataset household power consumption was selected for this study. It contains 2075259 rows and 8 columns of active power recodes from 16/12/2006 to 26/11/2010 by minutes, and I have selected ”Global active power’ measure by ”date time” as the time series target. In the code, the data was resampled to hourly intervals to reduce noise, and missing values were interpolated.

1.2 Identification of non-seasonal time series

To determine whether time series data is seasonal or non-seasonal, I have examined the following criteria:

- If the dataset is non-seasonal, the seasonal component in the decomposition plot is flat or absent.
- The Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots do not show repeating patterns at regular intervals.
- The Augmented Dickey-Fuller (ADF) test checks if the series has a unit root. A p-value > 0.05 indicates the series is non-stationary, but this does not directly imply seasonality.

Since there are cases where a series can be stationary but still non-seasonal, the ADF test does not directly address the question of non-seasonality. However, it is worth to look at it.

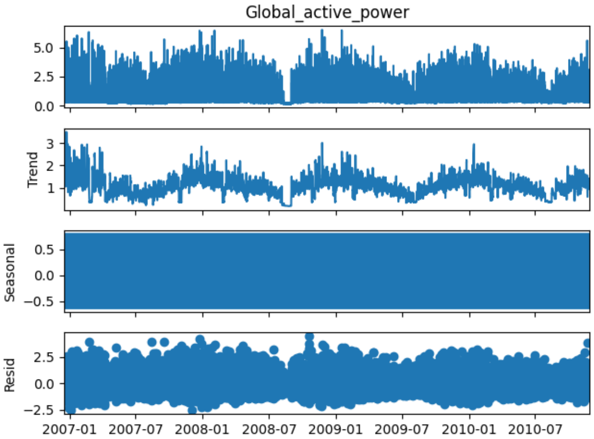

Figure 1: Seasonal Decompose

First, as in Figure 1, the decomposition plot appears flat, and Figure 2 shows that the plots do not exhibit perfect repeating patterns at 240 intervals.

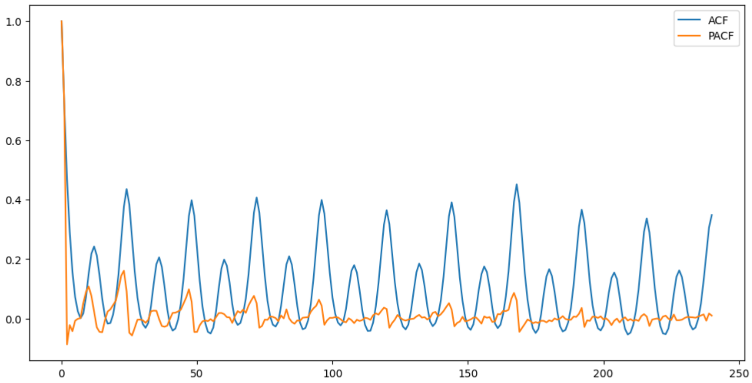

Figure 2: ACF, PACF

Additionally, ADF test shows that ADF Statistic: -14.382 and P-value: $9.101\times\ e^{-27}$. Therefore, the dataset is both stationary and non-seasonal.

1.3 ML Models

I have trained from the beginning of 80 %, and tested and validated for the last 20 %. I have deployed

Seasonal Autoregressive Integrated Moving Average with Exogenous Regressors (SARIMAX), XGBoost with optuna, and Long Short Term Memory (LSTM). The SARIMAX model is a powerful time series forecasting technique that extends the traditional ARIMA model to account for both seasonality and external factors. Since I have seen the better performance using XGBoost with optuna in the classification cases, I have applied in here as well. LSTM is known to provide greater accuracy for demanded forecasts.

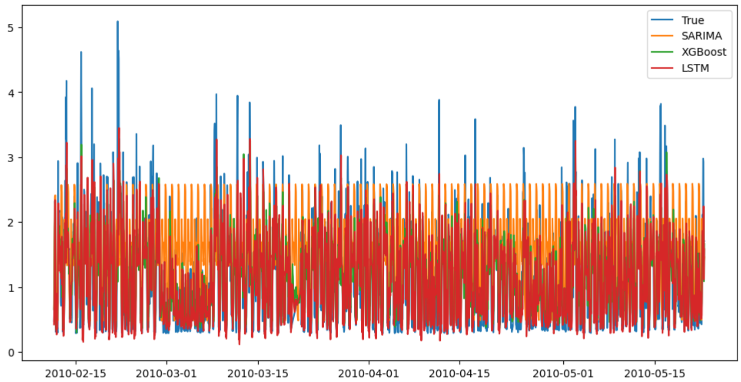

Figure 3: Prediction results of 100 time (24h) periods in the future

As per the assignment, I plotted predictions for 100 time periods (24 hours each) in Figure 3. (It is hard to see the details.) I have utilized Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-Squared (R2) as KPIs shown in Table 1. While Figure 3 does not show clearly the results, Table 1 shows that XGBoost is performing slightly better than LSTM. This may be because I could not perform hyper-parameter tuning (Optuna, Keras Tuner, or GridSearchCV) for the LSTM due to hardware limitations.

Model Performance Metrics			
Model \ KPI	MSE	MAE	R^2
SARIMAX	0.629	0.598	-0.061
XGBoost	0.266	0.358	0.551
LSTM	0.290	0.376	0.510

Table 1: Comparison of Model Performance Metrics 100 time 24-hour periods

Additionally, I plotted the prediction results of 10 time (24h) periods in the future in Figure 4. This provides a clear view compared to Figure 3. As shown, XGBoost, LSTM predictions are performing better than SARIMAX.

Table 2 shows the KPIs number. XGBoost and LSTM results are slightly worse than previous one. This suggests that the accuracy of each model varies within a certain range. To demonstrate the accuracy range of each mode, I experimented the predictions of 1, 5, 10, 20, 50, 100, and 200 time (24h) periods in the future. MSE in SARIMA is about $0.371 \leq \text{MSE} \leq 0.827$, XGBoost is about $0.211 \leq \text{MSE} \leq 0.4287$, and LSTM is about $0.239 \leq \text{MSE} \leq 0.465$. (Other KPIs are not shown here.) Therefore, XGBoost and LSTM are the more accurate models for this study.

Finally, for delivery and reproducibility, I saved the trained models as **pkl** and **keras** files, allowing others to load these files and easily run tests.

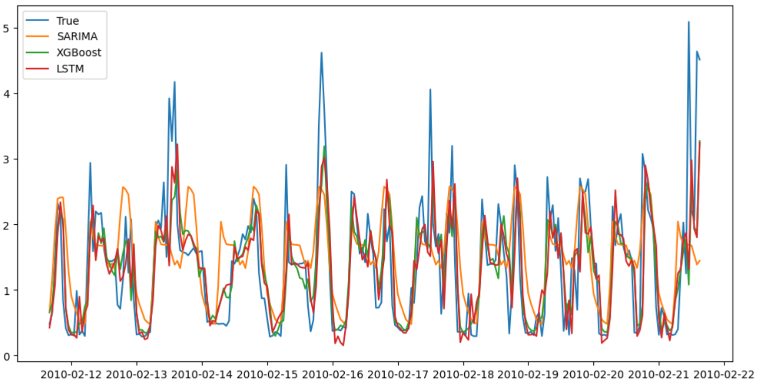

Figure 4: Prediction results of 10 time (24h) periods in the future

Model Performance Metrics			
Model \ KPI	MSE	MAE	R^2
SARIMAX	0.599	0.543	0.344
XGBoost	0.425	0.417	0.535
LSTM	0.459	0.439	0.497

Table 2: Comparison of Model Performance Metrics (10 time 24-hour periods).

2 Conclusion and Further Discussion

Time series analysis is continually evolving, driven by technological advancements and the growing availability of data. I aim to continue exploring automated model selection, nonlinear models, and machine learning with large datasets. Real-time forecasting using generative AI is particularly an intriguing area for further research.