

Predicting The Energy Market's Day-Ahead Prices

University School of Management and Entrepreneurship
Delhi Technological University
MA Economics

Abstract

This research project offers an in-depth look at the dynamic European energy markets. The dataset used in the research provides an excellent analysis of Europe's energy systems and power prices on an hourly basis. It can be used in many ways to examine the electricity market of Europe, including correlations between different energy systems, implications for prices in specific markets, and much more. Here, we are analysing historical trends in pricing behaviour to predict future prices for energy markets in Europe using the R.

Introduction

The dynamics of the electricity trade have been completely reshaped in recent times. In particular, electricity has become a commodity that displays a set of characteristics that are uncommon to other markets: a constant balance between production and consumption, load and generation that are influenced by external weather conditions, and dependence of the consumption on the hour of the day, day of the week, and time of the year. Due to these facts, the dynamics of electricity prices exhibit behaviour unseen in other markets, e.g. sudden and unexpected price peaks or seasonality of prices at three different levels (daily, weekly, and yearly).

As a result of this unique behaviour, electricity markets have become a central point of research in the energy sector and accurate electricity price forecasting has emerged as one of the biggest challenges faced by the different market entities. The usual motivation behind these efforts is purely economic. Accuracy in forecasting electricity prices reduces the risk of under/overestimating the revenue for the participants and provides better risk management. As forecasting accuracy increases, the negative economic effects of price uncertainty are mitigated and the market players make an economic profit.

In addition, another important fact to consider is that electricity markets are established to keep the grid stable. In particular, as prices become more volatile, the grid balance is compromised, strategic reserves may have to be used, and the risk of blackout increases. Therefore, by accurately forecasting electricity prices, not only can economic profits be made, but also the system stability is improved.

Methodology

Our research includes time series analysis and forecasting of European energy prices using the R programming language.

Time series analysis is a specific way of analysing a sequence of data points collected over time. The dataset we used contained energy prices from July 2010 to January 2022. For the forecasting to be reliable we wanted to have at least 10 years of data and use that for better model learning but our software was incapable of processing such large amounts of data hence for simplification purposes we have considered only 3 years of data for forecasting.

For forecasting future prices we have used the ARIMA model, short for 'AutoRegressive Integrated Moving Average', which is a forecasting algorithm based on the idea that the information in the past values of the time series can alone be used to predict future values. The Arima model uses stationary data because it looks at past data to predict future values, unlike non-stationary data which might produce unreliable and spurious results and leads to

poor understanding and forecasting. Hence, we have used stationarity time series data which means that the mean, variance, and autocorrelation structure do not change over time.

Data analysis

Since we had data for multiple regions, we decided to focus our attention on the HU region. The data set columns -

Column name	Description
fecha	Date of the power prices in DD/MM/YYYY format. (Date)
hora	Hour that corresponds with each set of power prices listed by minute. (Time)
sistema	Numeric code for system identifier for each set of reported price points for a specific hour across EU countries. (Numeric)
bandera	Indicator of whether or not electricity is green (Y) or non-green/conventional electricity (N). (Boolean)
precio	Cost per Megawatt Hour expressed in Euro €/MWh currency format. (Currency)

We have considered all the columns in our analysis except the bandera data as it was not useful to our analysis and time series model creation.

To continue the analysis we will consider only one system at a time. First, let's take a look at the HU system from 1st January 2019 to 1st January 2022¹. We will subset from our original data set to carry out the analysis.

So to start we have subsetting the data for the HU system and extracted the price column after deleting the data that wasn't needed.

```
#removing some observation from the start in order to make the analysis easier
#taking prices from year 2019
energy1[c(-(1:1126351)),] -> energy1

#renaming the columns
colnames(energy1)[1] = "Date"
colnames(energy1)[2] = "Hour"
colnames(energy1)[3] = "System"
colnames(energy1)[4] = "Price"

#subsetting on the basis of system = HU
energy1[energy1$System=="HU",] -> huenergy

#separating price column from hu data
huenergy1 <- huenergy[,4]
```

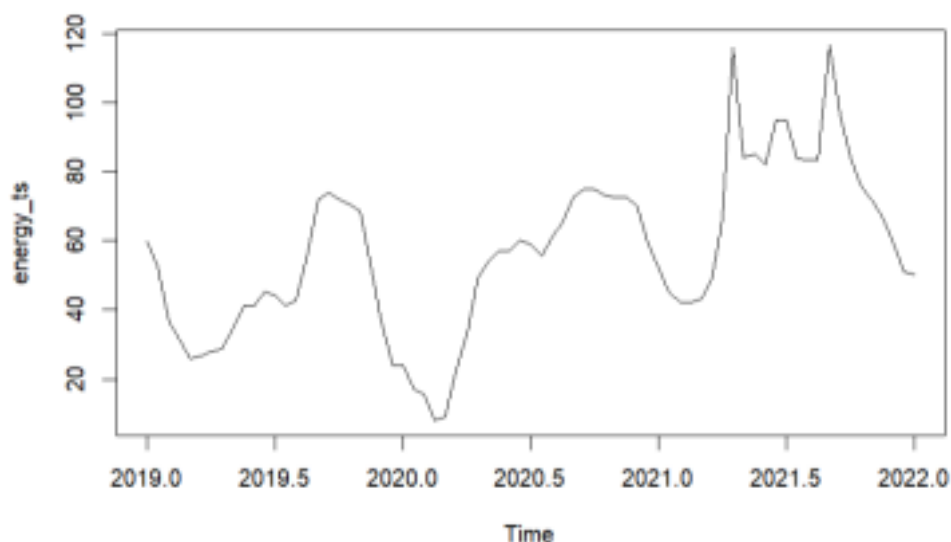
¹We have used 3 years of data because of the limitation in R software.

Then we will convert the price column into a time series data type using the ts() function.

```
#downloading time series package
install.packages("lubridate")
library(lubridate)

#converting prices into time series data type
energy_ts <- ts(huenergy1, decimal_date(ymd("2019-01-01")),
                decimal_date(ymd("2022-01-01")), frequency = 24)
```

Plotting this time series data we will get a graph like this



Now we can see the trend of how prices have changed over 3 years for the HU system. We will use this time series to forecast the next 24 values.

```
##Using Arima model to forecast
#checking if the data is stationary
install.packages("tseries")
library(tseries)
adf.test(energy_ts)

#setting up the Arima model
install.packages("forecast")
library(forecast)
energy_arima = auto.arima(energy_ts, ic="aic", trace= TRUE)

#checking if this model is stationary
adf.test(ts(energy_arima$residuals))
```

For forecasting purposes, we will use the ARIMA model. First, to apply ARIMA we need to check that the data is stationary which we will check using an ADF test. Running this command we will learn that the data is stationary. In time series analysis, stationary refers to a property of a stochastic process generating a sequence of observations over time, where the statistical properties of the process do not change over time. Specifically, a stationary time series is one whose statistical properties such as mean, variance, autocorrelation, etc. are all constant over time.

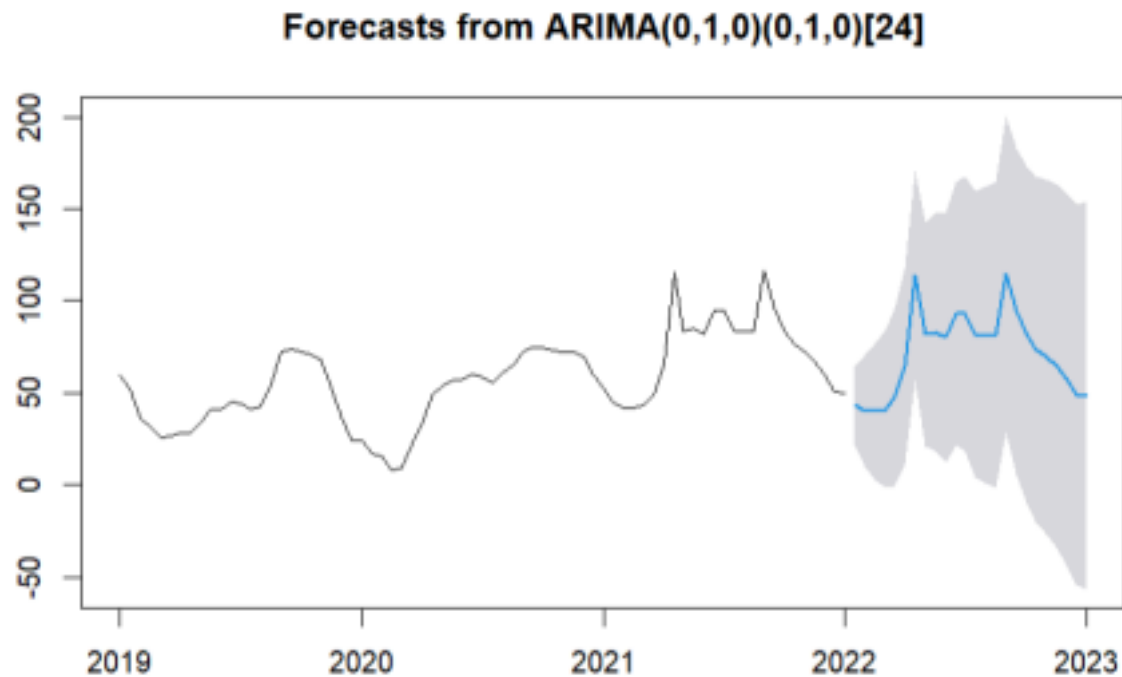
We will use the `auto.arima()` function to find the best-fit model. Here we have used the type AIC. The Akaike information criterion (AIC) is a statistical tool used to assess the quality of models for a given dataset by estimating their prediction error. It is a mathematical method that evaluates how well a model fits the data it was generated from. The AIC is often reported in structural equation modelling (SEM) software as a measure of the model's fit to the data. In summary, AIC helps researchers choose the best model for a given dataset by comparing the relative quality of different models.

Once R has run the command and found the best fit ARIMA model we will then check if this model is stationary using the same `adf.test()` function and learn that data is stationary still.

Now we will directly use the `forecast()` function to forecast the next values where the level is the 95% confidence interval we want to form.

```
#forecast
price_forecast = forecast(energy_arima, level = c(95), h = 24)
price_forecast
plot(price_forecast)
```

After which we will get this graph. The blue line indicates the forecasted values -



Conclusion

Comparing the Arima forecast with a simple forecast

We will use the forecast function to simply forecast the next 24 values of the function and then we will compare them to the predicted prices from the ARIMA model.

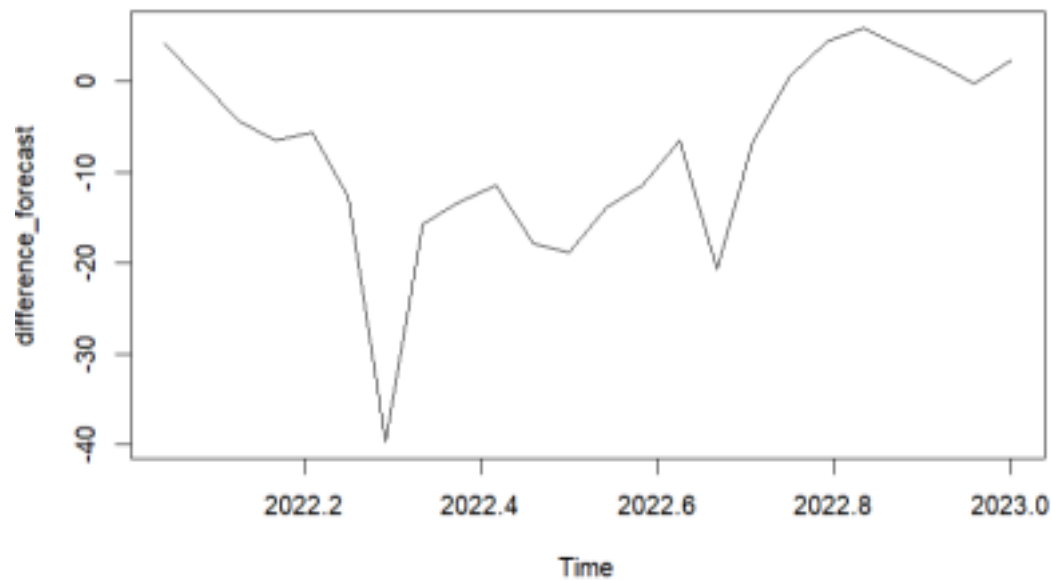
We have extracted the point forecast into 2 different vectors as seen below

```
#forecasting using simple forecast function
forecast(energy_ts,24) -> simple_forecast_hu

#storing the mean forecast values as a vector
simple_forecast_hu[["mean"]] -> simple_forecast_mean_values
price_forecast[["mean"]] -> arima_forecast_mean_values

#comparing simple forecast with ARIMA forecast
(simple_forecast_mean_values) - (arima_forecast_mean_values) -> difference_forecast
plot(difference_forecast)
```

After plotting the difference between the simple forecast function and the ARIMA model we will get this graph



Here we can see that most of the differences are not 0 i.e. there is a significant difference in the values forecasted using the forecast function and those forecasted from the ARIMA model. We observe that since all the differences are negative the simple forecasts underestimate the true forecasted value assuming that the true forecasted values are the ones obtained from the ARIMA model.

Limitations of the Arima model and alternatives

The ARIMA (Autoregressive Integrated Moving Average) model is a popular statistical method for time series forecasting. However, there are several limitations to this model that may make it less effective in certain situations. Some of these limitations include:

1. ARIMA models assume linearity: ARIMA models assume that the relationship between past and future observations is linear. However, in many real-world applications, this assumption may not hold, and more advanced non-linear models may be required.
2. ARIMA models require stationary data: ARIMA models assume that the data being analysed is stationary, meaning that the statistical properties of the data do not change over time. However, many real-world time series data are non-stationary, requiring more advanced models such as ARIMA with differencing or other methods to transform the data to stationary before modelling.
3. ARIMA models are sensitive to outliers: ARIMA models can be sensitive to outliers, which can significantly affect the forecast accuracy. Outliers can be difficult to detect and remove, and alternative robust models that can handle outliers are needed.
4. ARIMA models may not capture long-term trends: ARIMA models may not be able to capture long-term trends in the data. For example, if a trend is slowly increasing or decreasing over a long period, ARIMA may not be able to capture this trend accurately.
5. ARIMA models do not handle seasonality well: While ARIMA models can capture some forms of seasonality, they may not be suitable for modelling complex

seasonal patterns that require specialised models like Seasonal ARIMA or other seasonal methods.

Some alternatives to the ARIMA model that may be more suitable in certain situations include

1. Machine learning models: Non-linear machine learning models such as neural networks, decision trees, and random forests can handle non-linear relationships and can be effective for modelling complex time series data.
2. Vector Autoregression (VAR) models: VAR models can handle multiple time series variables and can capture relationships between variables, making them useful for multivariate time series analysis.
3. State space models: State-space models are a flexible and powerful class of models that can handle non-linear relationships and incorporate external variables, making them useful for a wide range of time series applications.
4. Exponential smoothing models: Exponential smoothing models are another popular class of models that can handle non-stationary data and capture trends, seasonality, and other patterns in the data.
5. Bayesian structural time series models: Bayesian structural time series models are a flexible and interpretable class of models that can handle complex time series data and can incorporate external variables and interventions.