ELECTRICAL PRICE PREDICTION

Authors: Weipeng Zhang, Soham Shinde, Mingni Luo, Ziyue Wang

Summary

Forecasting in energy markets is identified as one of the highest leverage contribution areas of Machine Learning towards transitioning to a renewable based electrical infrastructure.

The dataset contains hourly data for Energy Generation, Consumption, Pricing and Weather from years 2015 to 2018 for Spain. There are two CSV files: *energy_dataset.csv* and *weather_features.csv*. Both of them are time-series data having sizes (35064 x 29) and (178396 x 17) respectively. There are four types of features: *date information, energy, weather and price information*, which includes all 43 features.

The goal of the project is to predict the electrical price by the time of the day using generation, consumption and weather data and also analyze what features influence electrical price the most in Spain. After data wrangling and feature engineering, we implemented time-series models, boosting models and also LSTM model to tackle this problem. We compared RMSE of all models and analyzed the result to figure out which model and features are more suitable for this problem.

Methods

EDA

Since the dataset is time series, we assume there are some patterns behind it. For the exploratory data analysis part, our goal is trying to find some clues for feature engineering and selection.

First of all, we checked the distribution of electricity price by year [Fig1], month, week level.

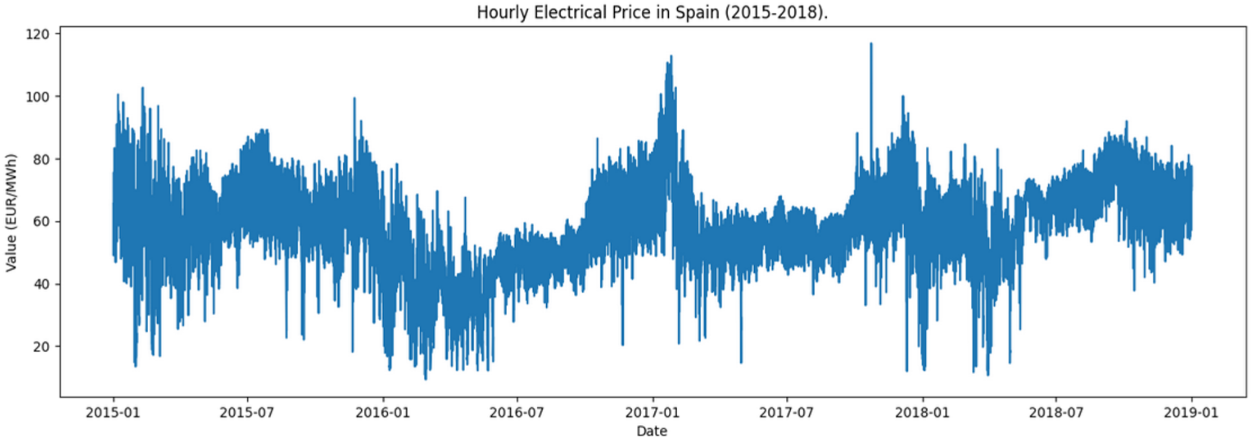


Fig 1. Distribution of price by 3 years

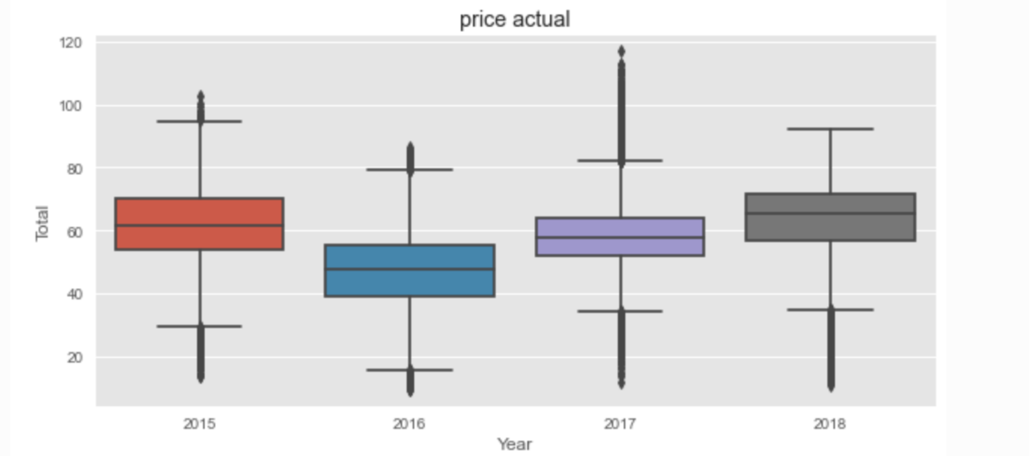


Fig 2. Distribution of mean price by year

It's hard to see every detail of the data if we visualize its distribution by year. But it's clear that the shape of this distribution is like a repetitive curve. We have more confidence that it has some patterns behind it. So we need to split the data into smaller groups.

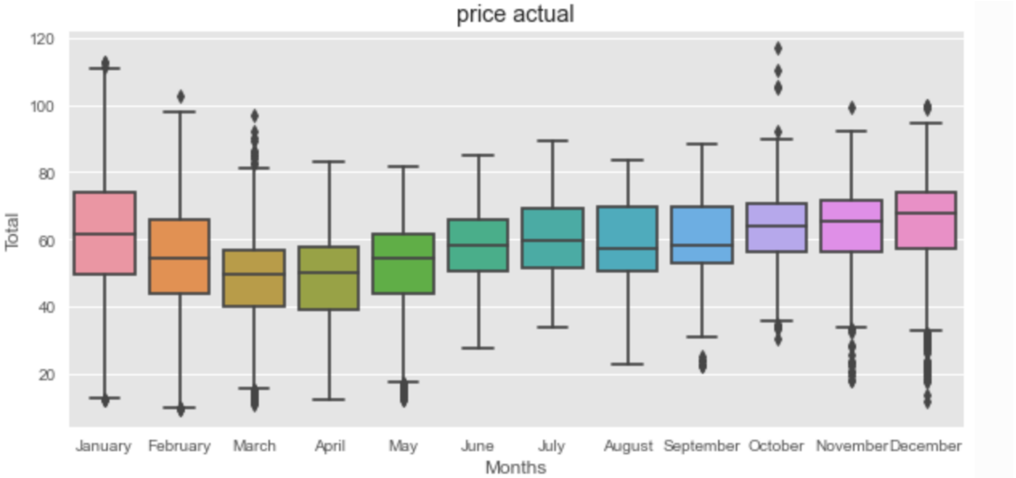


Fig 3. Distribution of mean price by month

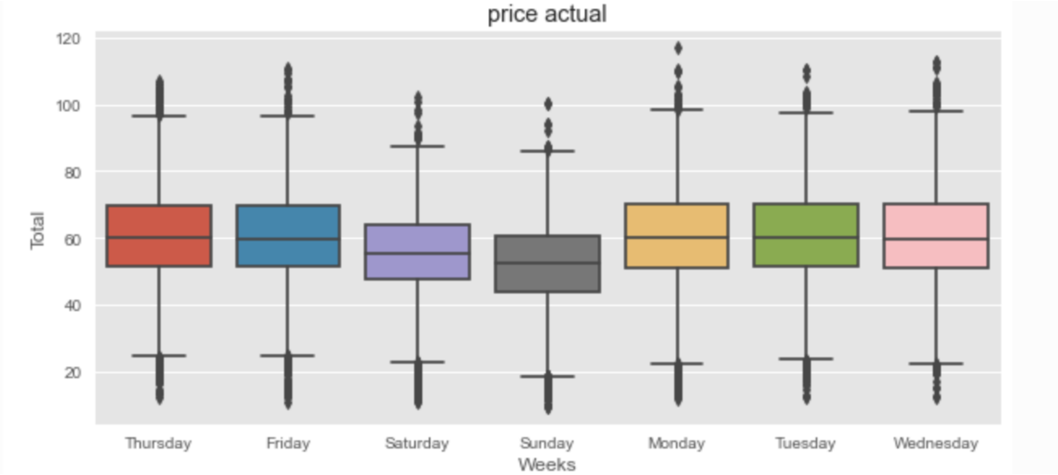


Fig 4. Distribution of mean price by week

Now we have some assumptions by these plots:

1. Price is affected by seasonality
2. Price is affected by temperature

To verify our assumptions, we need to compare the distribution of price and temperature.

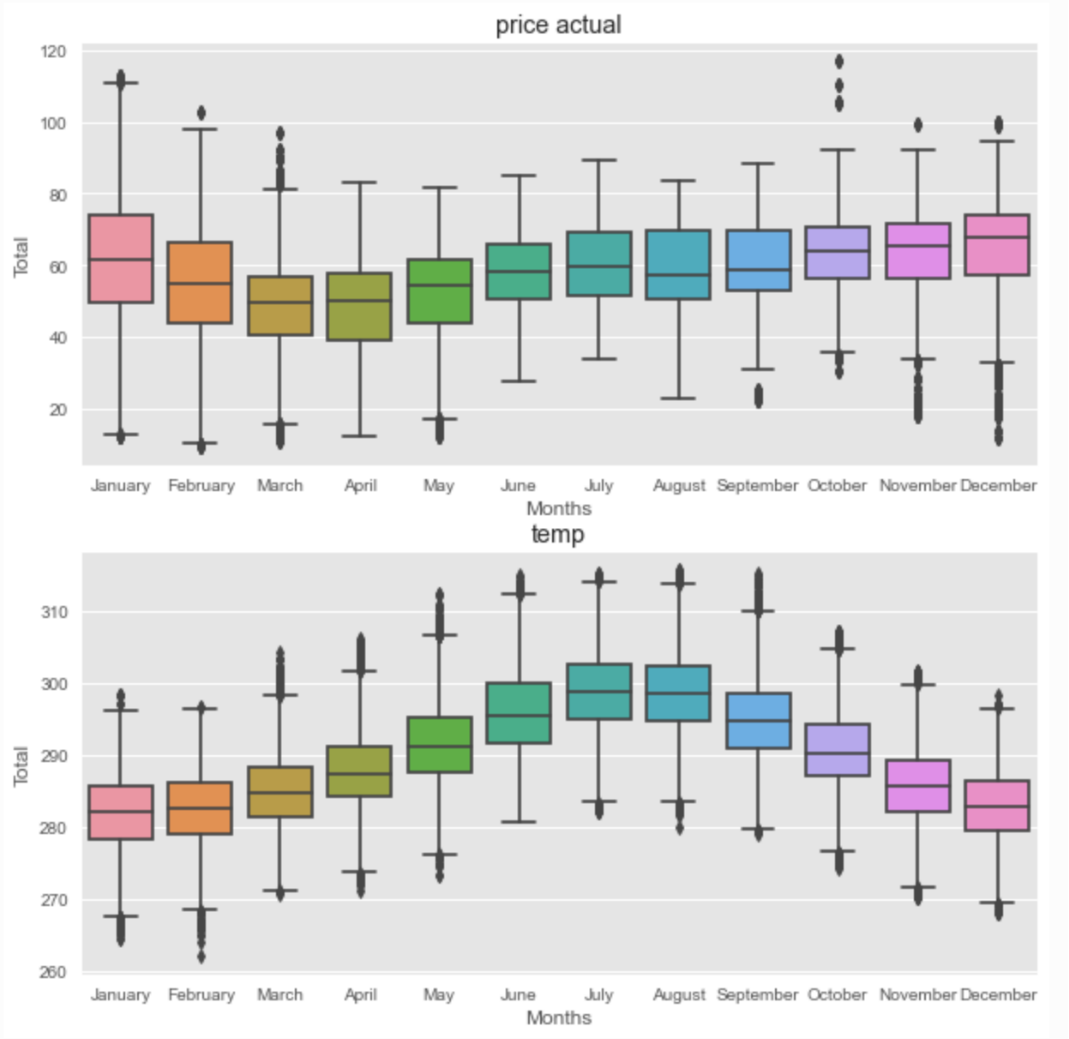


Fig 5. Distribution of price and temperature by month

The price will become more expensive when the temperature is lower or higher, which means temperature is a vital feature for our goal. Also we plot the correlation plot to help us decide better later [Fig A1].

Preprocessing

Outliers Detection

We detected the outliers in our weather data. By looking at the distribution of individual features, it shows that there are extremely large values in pressure and wind speed [*Fig A2*]. In addition, there are heavy rain and snow during the summer of 2016 and winter of 2015. We replaced these abnormal values with NAs and then imputed them.

NA Imputation

First, we checked the NA values in the energy data and weather data. It shows that *generation hydro pumped storage aggregated* and *forecast wind offshore eday ahead* in energy data have 100% NA values. So we marked these two columns as invalid features and will drop them later. In terms of other columns which do not have that many missing values, we imputed them with certain strategies. The reason we did not drop those rows is that since this is time-series data, we need to guarantee the continuity of each row, to prevent any issues that will potentially damage the prediction. First, we tried imputing them by the value of the last valid row. And then we also tried imputing those values linearly, which is imputing them with the values that make the context values increase or decrease linearly with equal intervals. In the model-building part, the results demonstrated that linear imputation is better so we kept that.

Column Drop

Besides the columns that we mentioned in the NA imputation, we also found invalid columns like 'generation fossil coal-derived gas', 'generation wind offshore', etc. which contain 99% zero values. Additionally, the energy data contains forecasted features like 'total load forecast', 'price day ahead', 'forecast solar day ahead', 'forecast wind onshore day ahead' from TSO. In order to compare our prediction results with theirs, and prevent data leakage problems, we dropped these columns as well.

Data Merge

As mentioned, the data we used to predict the model contains two files, one is energy data, the other is weather data. The goal of our project is to predict the electricity price based on energy-related data and also weather data. So we need to combine them together. We can do that by joining these two data frames together by the time column. But since the weather data file contains weather information in 5 cities in Spain, we need to handle them first. The easiest way could be directly calculating the mean value of all the features. But we also tried utilizing the data in different cities and pivoting wider with extra columns. In the modeling part, it shows longer training time and worse RMSE value, so we dropped this approach.

Data Shift

To convert the data into the format that fits the classical regression models, we created a new column called `price_tomorrow` by shifting our target column ‘price’ one day ahead. By doing this, we can train the model properly. However, for our time-series models, we predicted the rows by previous data, so it’s not necessary to perform this.

Feature Engineering

In the feature engineering part, we generated five different types of features.

- The first type is statistical indexes in groups. These features were generated by calculating certain statistical values of specified variables of specified groups, such as the mean price of this week, mean price of this month.
- The second type is historical value. We took the historical value of a specified variable as a new feature, such as price of an hour ago, price of a week ago.
- The third type is date features like day of week and day of month. These features were generated directly from the date-time of the row.
- The fourth type is statistical indexes in sliding windows. We calculated statistical indexes in a sliding window, such as the mean price of the last 6 hours.
- The last type is the trend of variables, such as the growth rate of price in the last hour and the growth rate of price of this month.

We finally generated 50 new features, making the number of features used for training become 85.

Modeling

Classical Regression Model

We used the features generated from the feature engineering part as the input of 7 different models: linear regressor, ridge regressor, lasso regressor, random forest regressor, LightGBM, XGBoost, and CatBoost. We used the last ¼ part of the train set as the valid set for the training process. The gradient boosting models performed better among all models and CatBoost was the best. We took the average of the predictions of these 7 different models as our final prediction and the RMSE came down to 6.0847.

Then we performed feature selection on the input. We got three different feature importance tables from the trained LightGBM, XGBoost, and CatBoost models and then filtered out the last 10 features with the lowest feature importance. We used the filtered features to train the models again and lowered the final RMSE down to 6.0187.

Time Series Model

Time series data is a collection of observation values obtained through repeated measurements over time. The frequency for the same can be Annual, Monthly, Weekly, Daily or even second-wise. They have a natural temporal ordering. Being complex, the time series models make an assumption that the data is suitably prepared before fitting.

There are two types of analysis that can be done. Firstly, the univariate analysis in which we considered the series with a single time-dependent variable. Eg. AR, ARIMA, Prophet model. Whereas in Multivariate analysis, the time series is dependent on more than one time-dependent variable. Eg. VAR, Prophet model.

ARIMA

ARIMA stands for Auto-Regressive Integrated Moving Average. It explains a given time series based on its own lags and lagged forecast errors. This model is characterised by 3 terms (p,d,q):

- p is the order of the AR term ie. The number of lags of Y to be used as predictors.
- d is the minimum number of differencing required to make the time.
- q is the order of the MA term ie. number of lagged forecast errors.

This model assumes that the data is stationary which makes it mandatory to check for stationarity of the data and correct it. Kwiatkowski–Phillips–Schmidt–Shin (KPSS) test is used for hypothesis tests. We accepted the null hypothesis for the stationary data.

H0: Null hypothesis: the series is stationary.

H1: Alternative Hypothesis: the series is not stationary.

If the data is not found to be stationary, then we need to do some minimum number of necessary transformations like differencing so as to make it stationary (d). The value for p and q parameters is determined using Partial Autocorrelation Function (PACF) and Auto Correlation Function (ACF) plots. Then, we can manually pass the parameters to model for predictions. Auto-ARIMA variation can be used to automatically calculate the best combination of (p,d,q) based on Akaike's Information Criteria (AIC) value.

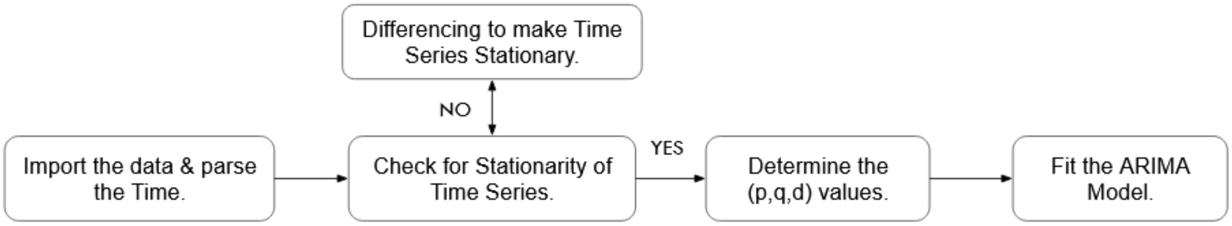

Differencing to make Time Series Stationary.
NO
Import the data & parse the Time.
Check for Stationarity of Time Series.
YES
Determine the (p,q,d) values.
Fit the ARIMA Model.

Fig 6. ARIMA model

We can extend the ARIMA model to multivariate analysis by using the Vector AutoRegression (VAR) model which includes other time-series data along with target series. Besides the generic steps of ARIMA, we also need to perform Granger Causality Test which determines whether a time series causes another time-series and accordingly consider the variables.

H0: Null Hypothesis: xt does not granger-causes yt

H1: Alternative Hypothesis: xt granger-causes yt

PROPHET

PROPHET is based on an additive model released by Facebook. It provides for both types of analysis. In Univariate analysis, the response variable to ‘y’ response parameter as well as to ‘ds’ predictor parameter. The Prophet Model needs to be given a period to make its predictions by initializing the ‘future’ parameter. In our case, the model is trained on data for years 2015-2017 and values for 2018 are predicted. We can visualize the components of the time series using this model [Fig A4]. In Multivariate analysis, we added other correlated variables as regressors to the model to get the prediction [Fig A5].

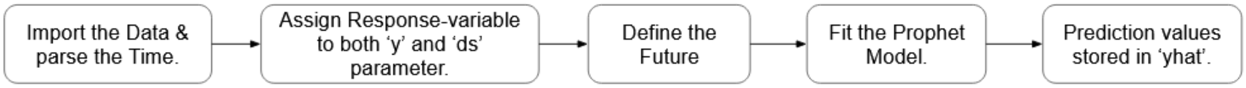

Import the Data & parse the Time.
Assign Response-variable to both ‘y’ and ‘ds’ parameter.
Define the Future
Fit the Prophet Model.
Prediction values stored in ‘yhat’.

Fig 7. PROPHET model

LSTM

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning.[1] Unlike standard feedforward neural networks, LSTM has feedback connections. So it would be perfect for time series data prediction.

For training and validation data, we split them with 3:1. As one of the RNN models, LSTM is super sensitive to the Input. It is important to scale features before training the model. Here we used MinMaxScaler in SKlearn to do the normalization.

$$f(x) = (x - min) / (max - min)$$

The scaled data should only be computed using the training data so we only fit the scaler with training data. And then we transform the data into a 3d array with the shape of [samples, timesteps, features]. Here we need to predict the electricity price the next day, so the timesteps would be 24. After training the model, we got RMSE 4.4456 for the test data, which is the best we got so far [Fig A6].

Results

Model	Linear Regressor	Ridge Regressor	Random Forest	LightGBM	XGBoost	CatBoost
RMSE	6.982	6.980	6.47	6.27	6.39	6.21

Table 1. RMSE value for classical regression models

Model	ARIMA	VAR	Prophet Uni.	Prophet Multi.	LSTM
RMSE	14.15	14.08	13.21	9.90	4.45

Table 2. RMSE value for time series models

In the case of Time-Series models, the LSTM model has the most accuracy followed by the Prophet and then the ARIMA models. For classical regression models, CatBoost has the best performance among all. We further compared our prediction result with the forecast value from the Transmission Service Operator (TSO). It shows that the RMSE value of our best model is 45% less than TSO, which verified that our prediction is better than theirs [Fig A7].

The variables having the most importance for the prediction of electricity price are ‘day of the week’, ‘fossil oil generation’, ‘biomass generation’, ‘hour of the day’ along with other feature engineered variables. In weather data, it’s interesting to note that volume of rain and snow do not have a large impact on the actual price.

Discussion

In the power market, the electricity price is an essential element because it influences the behavior of power generation enterprises, power supply enterprises, and other buyers. [2]
First of all, for the entire power system, electricity price forecasting solves the problem of shortage of capacity in certain periods and large amounts of surplus in certain periods to a certain extent.
For the users, if we can get an accurate prediction of electricity price, it can allow users to better control budgets. They can formulate a reasonable electricity consumption plan according to actual needs.
From the perspective of generators, accurate electricity price forecasting is conductive to accurately grasp the market trend and market opportunities, so as to construct the optimal electricity and electricity price bidding strategy in order to obtain the maximum profit. From the perspective of power investors, long-term electricity price forecasting helps them make correct investment decisions and reduce investment risks.
An accurate forecasting electricity price model will be required by governments, enterprises, and users in various countries.

In future work, to improve the Time-Series Models, SARIMA (an ARIMA model which handles the seasonality) can be used. Also, the reason for the underfitting of time series models can be explained. To improve the performance of the classical regression models, further feature engineering operations can be conducted. We believe that applying stepwise feature selection can also help to improve the performance and efficiency of the classical regression models.

Statement of Contributions

Ziyue Wang: Performed extra data preprocessing methods, and implemented the LSTM model.
Soham Shinde: Time Series analysis with implementation of specific models of ARIMA, VAR and Prophet.
Mingni Luo: exploratory data analysis part and ARIMA model.
Weipeng Zhang: Responsible for the framework of the code, including exploratory data analysis, data preprocessing, feature engineering, building and training classical regression models.

References

Dataset used :
https://www.kaggle.com/nicholasjhana/energy-consumption-generation-prices-and-weather

[1] Sepp Hochreiter; Jürgen Schmidhuber (1997). "Long short-term memory". Neural Computation. 9 (8): 1735–1780. doi:10.1162/neco.1997.9.8.1735. PMID 9377276. S2CID 1915014.
[2] D. Liu and H. H. Qiao, “A review of electricity price forecasting in deregulated electricity markets,” Shaanxi Electric Power, vol. 39, no. 8, pp. 27–32+41, 2011.

Appendix

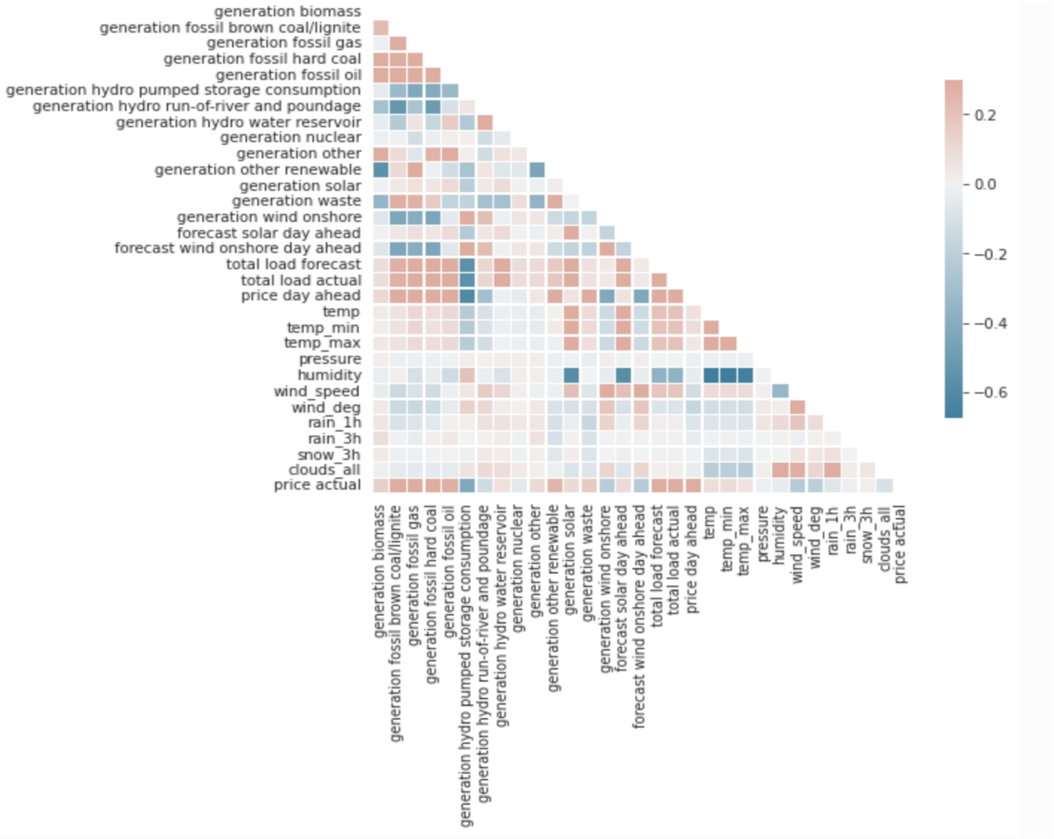


Fig A1. Correlation matrix

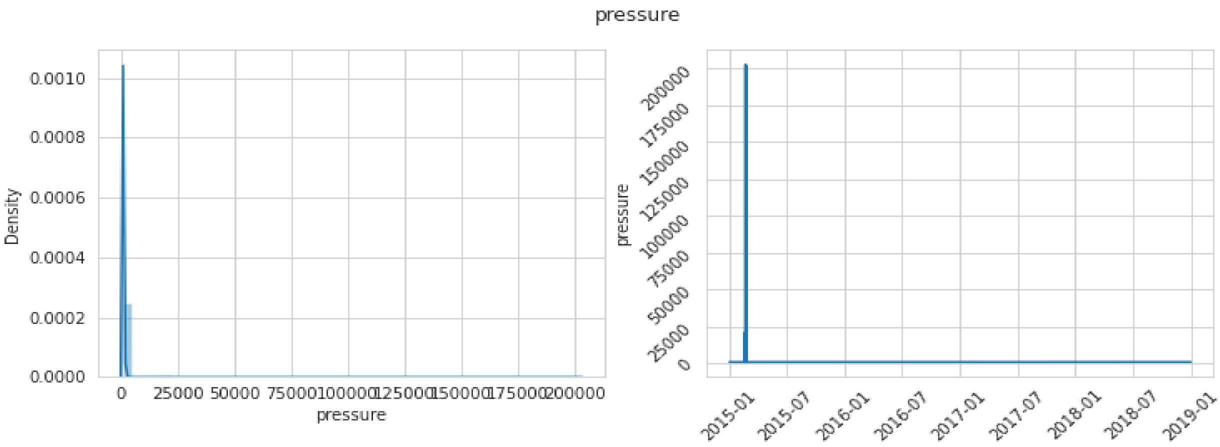
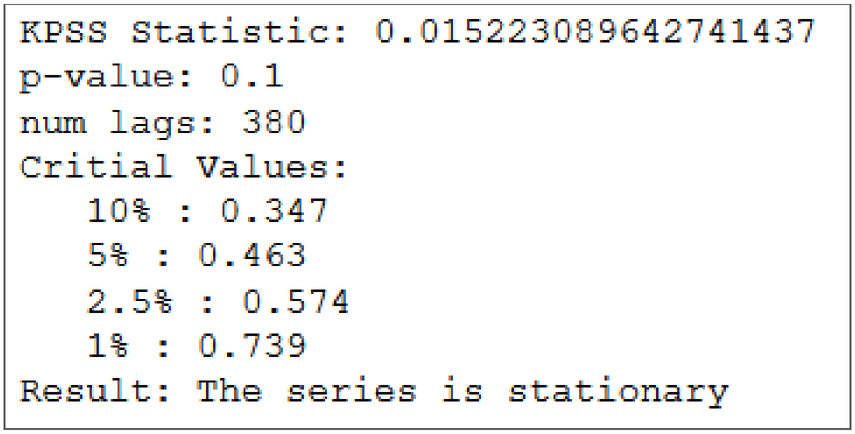


Fig A2. Outliers in pressure



```
KPSS Statistic: 0.015223089642741437
p-value: 0.1
num lags: 380
Critial Values:
   10% : 0.347
   5% : 0.463
   2.5% : 0.574
   1% : 0.739
Result: The series is stationary
```

Fig A3. KPSS test

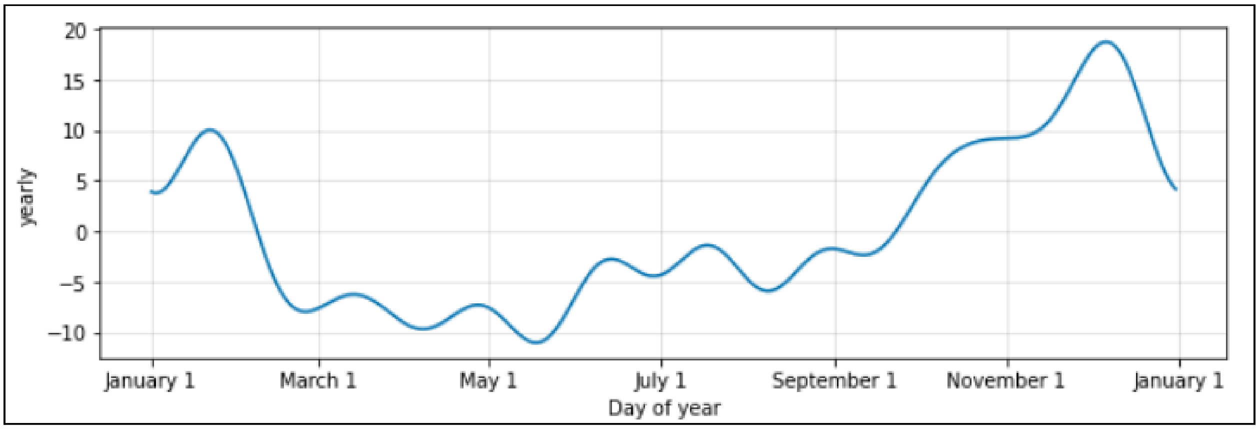


Fig A4. Annual component of Prophet forecast

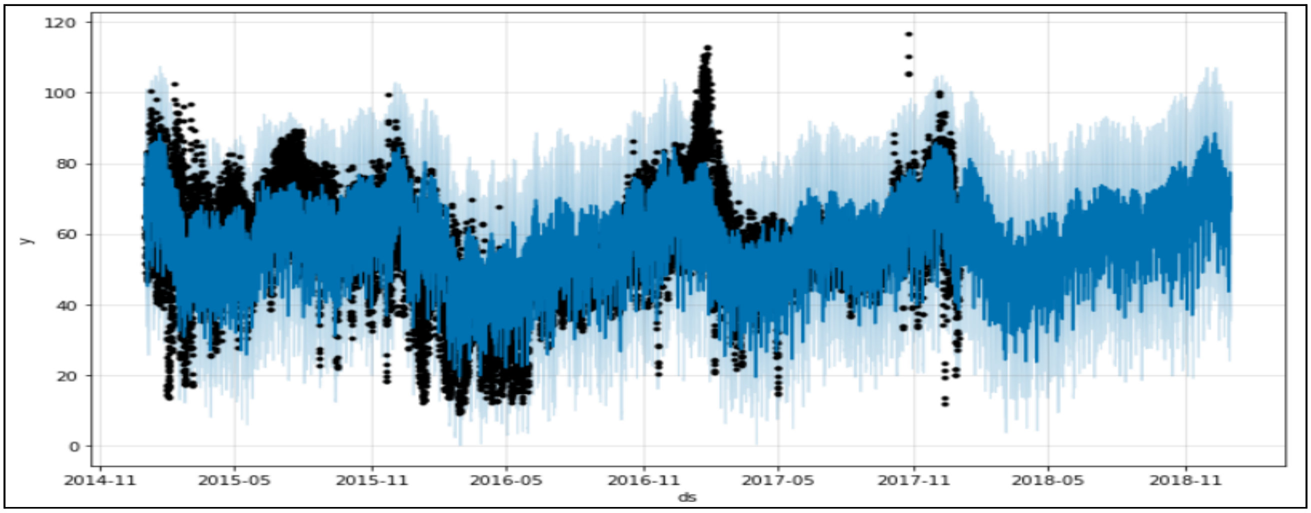


Fig A5. Electrical price prediction using multivariate Prophet modeling

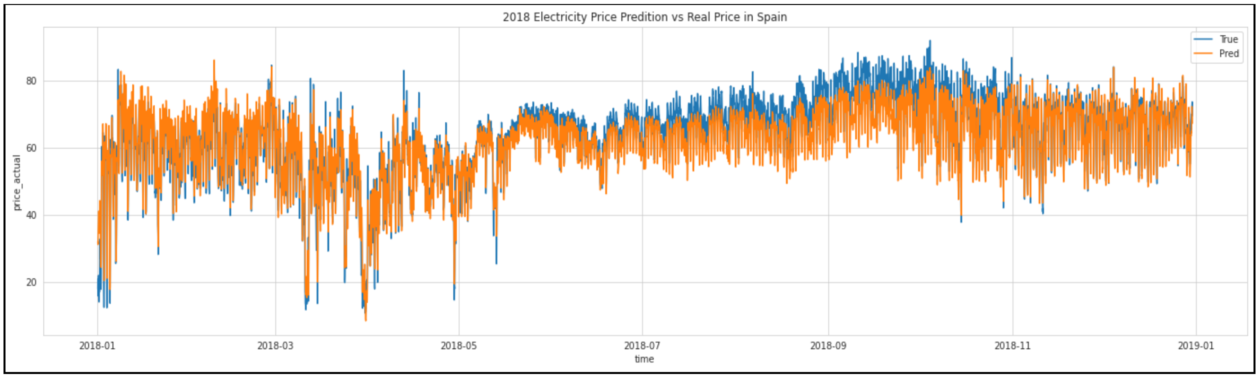


Fig A6. Electricity Price Prediction VS Real Price in Spain in 2018

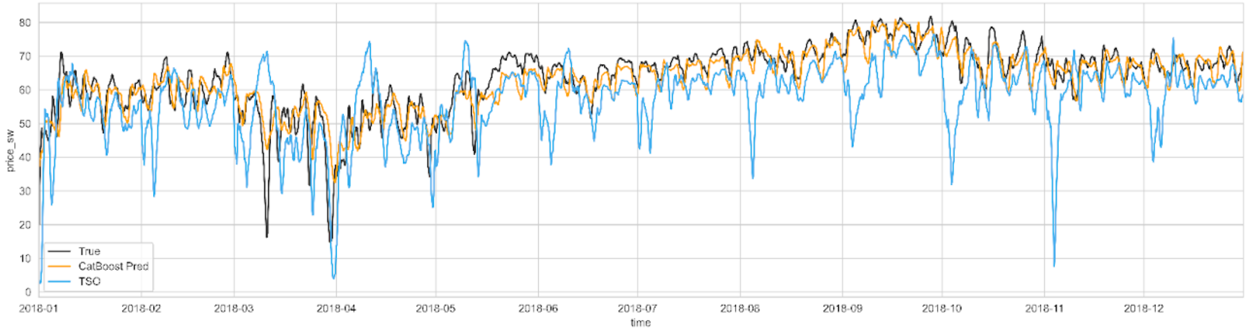


Fig A7. Price of TSO, real and our predicted comparison