Time Series Analysis Project

Topic: Spain Electricity Shortfall Challenge(Kaggle)

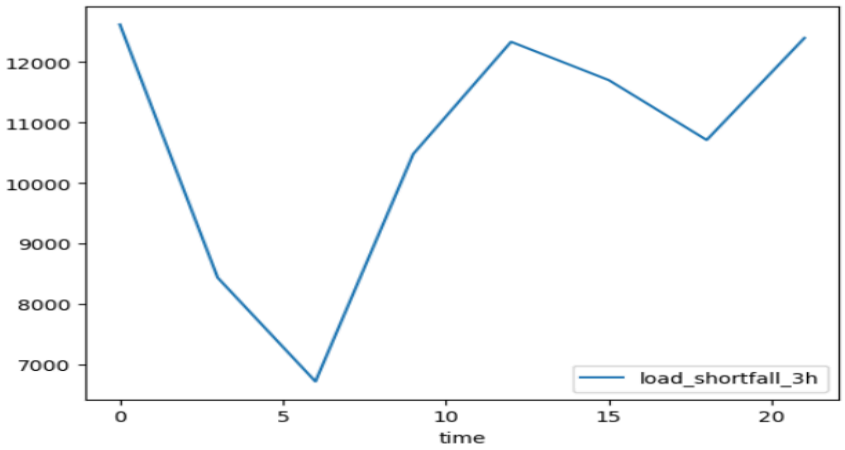
Task: Predict the expected electricity shortfall between the energy generated by means of fossil fuels and renewable sources.

1. Data Description

I was given a dataset consisting of various atmospheric statistics across five different cities: Madrid, Barcelona, Seville, Valencia, and Bilbao, marked on the map below. The data contained features describing wind, rainfall, humidity, cloudiness, pressure, and temperature. Moreover, 4 out of the 5 cities also had a weather_id feature, which was quite enigmatic, as it was not explained in the documentation.

2. Exploratory Data Analysis

At first, I performed exploratory data analysis, searching for outliers, null values, and examining data seasonality. The data seemed to be highly dependent on the time of day. The shortfall rose in the morning (10 AM – 1 PM), then slightly decreased between 3 PM and 7 PM. After that, the shortfall spiked, usually reaching the day's maximum.



Besides this, I examined the correlation among all features and removed a few due to excessively high correlation.

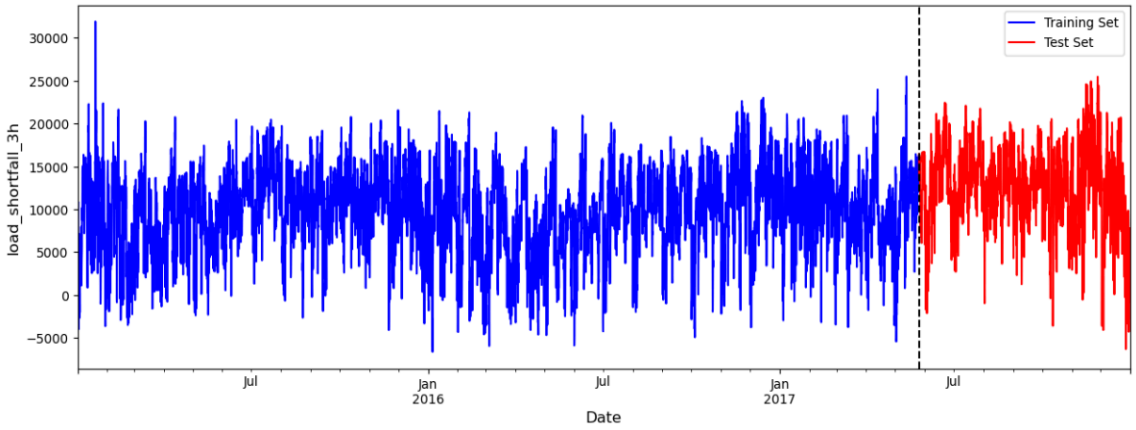
The most interesting part of exploring the data was discovering that all four weather_id columns contained identifiers from OpenWeatherMap. Now that I knew where they came from and what they meant, I could leverage them in the next section of my project.

3. Feature Engineering

At first, I transformed all weather_id columns into multiple new features based on ID groups from OpenWeatherMap. This resulted in approximately 35 new features. Next, I added seasonality to the data by creating seven time-based features. The final step in this section was scaling the data using StandardScaler. After many rounds of modeling, I also added lag features, which completely changed the outcome of the project.

4. Modelling

This section began with splitting the data into training and testing datasets. The key aspect was ensuring that dates did not overlap.



Next, I trained four different models on the data, using XGBoost, Decision Tree, Random Forest, and an LSTM neural network. At first, each of the models failed to make good predictions, with XGBoost being the best model. XGBoost achieved an RMSE of around 4000 and an R^2 of around 40%. I tried tuning the XGBoost hyperparameters, but this resulted in only a slight improvement in the model's performance.

After many trials, a complete game-changer turned out to be the lag features of the energy shortfall. This reduced every model's RMSE significantly and allowed XGBoost to achieve a score of 1731 RMSE and 93% R^2. It is worth noting that Random Forest also performed well and achieved a very similar RMSE. However, when it comes to R^2, the difference between the models was more pronounced. Taking all statistics into account, XGBoost outperforms in all of them, and therefore, I chose it as the best model for predicting energy shortfall.

5. Benchmark statistics

	Test RMSE
Decision Tree	2502.348578
Random Forest	1759.008402
XGBoost	1731.532697
LSTM	2999.513971

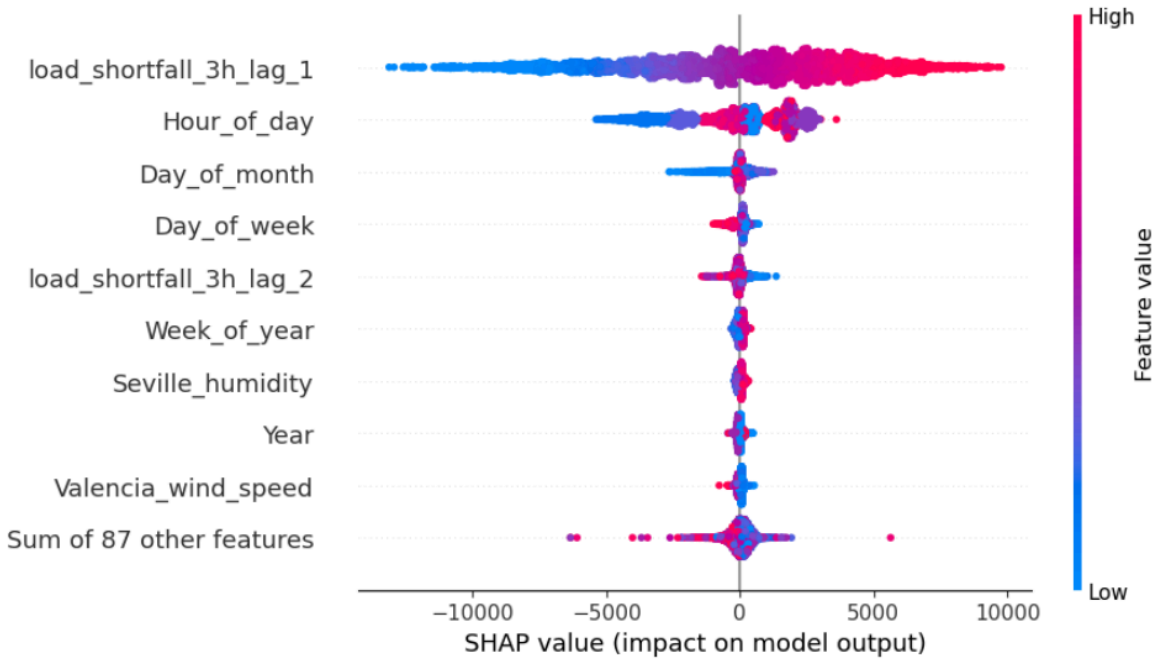
	Test R^2
Decision Tree	0.735971
Random Forest	0.869536
XGBoost	0.934655
LSTM	0.788000

	MAE
Decision Tree	41.22
Random Forest	35.55
XGBoost	35.54
LSTM	46.53

	MSE
Decision Tree	2502.35
Random Forest	1759.01
XGBoost	1731.53
LSTM	2999.51

6. Model Explanation

The final part of my project was identifying key features used by the best model to predict energy shortfall. For this purpose, I used SHAP values (which are also the topic of my Bachelor's thesis).



Conclusions below

7. Conclusions

The first lag feature of shortfall was definitely the most important feature throughout the entire dataset. In second place is the hour of the day feature, where we can observe the same trend described in the exploratory data analysis (EDA) section. Additionally, the shortfall seems to be lower in the first half and at the end of the month. Similarly, the day of the week feature tells us that the shortfall was lower on average in the second half of the week.

Next, we have the second lag feature, which seems to be inversely proportional to the first lag feature; however, its influence on the model's predictions is significantly smaller. These were the most important features used by XGBoost.