Analytathon 3

Executive summary

The purpose of our project was to develop a model to predict prices by analysing market trends, which could help Energia purchase electricity at the best price among the 5 markets.

To make predictions, the model was trained on a dataset of historical sequence data. During training, the model learns to recognize patterns in the data and use them to make predictions about future values in the sequences. Once the model has been trained, it can be used to make predictions on new and unseen sequences, to help identify the optimal market to purchase electricity at 30-minute intervals for D+1.

However, because of the complexity of the task and time constraints, we only managed to build a base model, that could give predictions but still could be improved. Therefore, further studies need to be carried out in order to explore how the model can be optimized to obtain a better prediction.

Introduction

Energia Group is an Irish energy company that provides electricity and gas to customers in Northern Ireland (NI) and Republic of Ireland (ROI). Energia decided to buy electricity across 5 markets, and they decided to buy the electricity at the best price to optimise the business. The purpose of this report is to develop a Long Short Term Memory (LSTM) model to predict prices by analysing market trends, to guide Energia buying electricity at the lowest price. This report will also show our prediction of our model on the last day to demonstrate the performance of our model.

The report has been organised in the following way. We begins by focusing on discussing data exploration in more detail. The next part is concerned with the methodology used for this study, we will explore the construction of LSTM model and the application of LSTM model on market price forecasting. Finally we will give our recommendations and discuss the potential limitations of our work.

Due to time constraints and the complication of the project, this project could not provide a perfect model to guide the purchase. We have only managed to build a model that can make predictions based on historical time series of the data.

Data Processing

For this project, we have four variables that will affect the market price, which are demand, wind, gas price and electricity price. In the “prices_niv” file, the prices for all five markets of each 30-minute interval of everyday and national imbalance volume(NIV) are saved. We imported these five datasets into a jupyter notebook. Since some of the datasets are started from the second row and column, we formatted all the datasets immediately after importing by removing blank rows and blank columns.

We filter out the missing rows in “demand” dataset, insert the consequent time and date according its position, and set 0 to the demand variables, so now each data date and time period is continuous and uninterrupted.

After inserting the missing rows, we join all the datasets and filter the result dataframe by removing data whose date is after March 31, 2023. This is because the “prices_niv” stops at this data, to make sure we have all four variables for each slot to train the model, we choose to stop at this date. We convert all the data types to numeric as well in this step. The dataframe that merged all the data and contained all columns is called “combined”.

We now have variables for each day and every time slot. Yet, there are still some rows that include 0 or NAN. These are the missing values that will impact the results and we need to replace them with a reasonable value. Since the interpolation could only be applied to 0 but not NAN, NAN was converted to 0 first. Apply `pd.interpolate()` to each variable column of “combined” and set the parameter `method = ‘pad’`. This parameter will copy the value just before the missing entry, this is the most sensible method among all the methods.

To help with training, we created a new column `net demand` that holds the variables created by the given variable, which represents the difference between electricity demand and wind power generation.

Data Exploration

In the data exploration, we worked out how each variable changes along the time, which is the univariate analysis; how each variable is related with others or its past data, which is the multivariate analysis; and outlier detection. We visualized time series data to achieve the above objectives. These visualization graphs can provide informative insights to specify time-related features, such as trends, cycles, and seasonality, that may affect the model selection (Brownlee, 2019).

Univariate Analysis

Here we first draw the demand, wind and net demand in a line plot, shows how the values changed with the time. From the plots, we can clearly see the existence of cycles, roughly one year at a time, with demand and wind being more cyclical and net demand being essentially flat, because it is the difference between the two above, but there is still some small downward arc, because the gap is larger near the beginning and end of the cycle.

Next we plot the gas and electricity price.Even if these two graphs do not have exactly the same shape, we can still observe that prices show great volatility between October 2021 and April 2022. There is another peak between July to October of 2022. The price increased again and reached peak at around the end of the 2022. The trend and the position that has the peak values are nearly the same. This suggests that gas and electricity prices always have similar movements, which could be due to some reason or they are affected in the same way.

Then we plottted the markett prices. The plots include 5 markets and the Market Net Imbalance Volume (NIV).The plot shows the fluctuation of energy prices and NIV over time, which can be useful for understanding trends and making informed decisions regarding energy consumption and cost management. The plot also provides a visual representation of the relationship between the different energy prices and the Market Net Imbalance Volume over the selected time period.

If we increase the size of each plot to obtain a bigger and more detailed plot, we can observe that nearly all five markets share the same trend, they increase and decrease with similar slope and reach peak around the same time. This means the price of these five markets are actually closely correlated with each other, and we will talk about this later in the multivariate analysis secton.

Outlier Detection

Outliers are data points that lie far away from the majority and dramatically deviates from other observations in a dataset (2023). They are usually considerably larger or smaller than other values (Bonthu, 2023). The existence of outliers may have an impact on the result and could result from things like measurement errors or odd occurrences.

The graph shows the trend of the electricity demand over time, with each data point representing the demand for electricity at a specific point in time.The graph can help to identify patterns in electricity demand, such as seasonal fluctuations or changes in demand due to specific events as well as detect for outliers. Since we know the outliers are usually larger or smaller than other values, and from this graph we observed that the outliers are all in the lower part of the graph, so we try to detect outliers by finding values that is the smallest of all numbers.

By using the 'nsmallest()' method to display the "Demand (MW)" column's 10 smallest values, the code may spot outliers. These 10 examples had the lowest demand in comparison to the rest of the dataset. These 10 data points reflect an extremely low demand compared to the rest of the dataset, and outliers are data points that drastically depart from the rest of the data.

We plot gas to check for the outliers as well. We can see some values that are much higher than the rest of the data, indicating a potential anomaly or unusual event, hence these values are considered outliers.

Similar to above, we use the `nlargest()` function to get the 10 maximum values of the gas price column and treat them as anomalies.

As what we did before, we plot the electricity prices and we can observe that there are some abnormal values in the first season of 2021, from October 2021 to February 2022, and around December 2022. All these values are significantly larger than others. So we keep using `nlargest()` function to get the five largest values as anomalies.

Multivariate analysis

There are three stages we took while analysing relations between variables of time series data: stationarity, autocorrelation and decomposition. Stationarity is a way to measure if the data has varied across time, or on the other hand, if the data shows a strong trend or seasonality. Autocorrelation means the feature data is linear related to its own values in the past. Decomposition helps you to plot the trend, seasonality and residuals of your data. (Pierre, 2022)

We first drew a correlation matrix to see the correlation between any two variables. The darker the square, the stronger correlation the two variables have. Expect the squares on the diagonal, we found that there are few variables are strongly correlated, such as DAM market and IDA1 market, gas price and electricity price which we observed from previous analysis as well, IDA2 market and IDA3 market.

We work on the autocorrelation of each variable with `pd.autocorr()` function. The parameter `lag` means the period of time that compare with itself, more specifically is the number of month. We selected 4 different numbers represents the autocorrelation between different period of time:
- lag = 1: calculate the correlation between month M and month M-1.
- lag = 3: calculate the correlation between month M and month M-3, which is every season
- lag = 6: calculate the correlation between month M and month M-6, which is every half year
- lag = 12: calculate the correlation between month M and month M-12, which is every year

For stationality, we use Dickey Fuller test to help. This test will apply statistic hypothesis on the value, if the alternative hypothesis is rejected, then the data is stationary. (Pierre, 2022)

By spliting a time series data into several components, each representing an underlying pattern category.We can see the decomposition plot comprising three components: a trend-cycle component, a seasonal component, and a remainder component (containing anything else in the time series) ().

Methodology

To handle time sequence data, LSTM is a good choice. LSTM models are a type of recurrent neural network that are designed to capture these temporal dependencies in time series data. The unique architecture of LSTM models allows them to selectively remember and forget information from previous time steps, making them well-suited for processing long sequences of data. It can effectively capture dependencies and patterns that exist over long time lags in the data.

The input data was split into train, validation and test. The test data contains the last 48 time slots of a day to evaluate the predictions. We splited the rest of the data into 80/20, which train data accounts for 80% and validation accounts for 20%.

Initially, we decided to use a single model to predict prices of all markets together, however due to the different opening times of IDA2 and IDA3 market, the model results in extremely

high loss during training and bad predictions. Therefore, we determined to build 3 models - one for DAM, IDA1, BM since they are all open 24 hours, one for IDA2 and another for IDA3. For the closed period of IDA2 and IDA3, we decided to fill the NAN with the median values of that day after several attempts.

The training loss of each model is plotted out in one graph. These curves decrease as the training process proceeds and then reaches a stable point. After this, we apply the model to the test dataset and creates a list of prediction. By calculating the Root Mean Squared Error(RMSE) between the prediction and our true values, we will know the performance of our model on test set, and the smaller the RMSE is, the closer the prediction is to the real data, the better result we have. The average RMSE of these three tests are within the range 40-50, which means the model works but still could be improved. This is thelimitation of out model and we will discuss this in the later part.

After finishing testing the data of the last day, we append our predictions to the “combined” dataframe which saves all our variables, and the columns name add “prediction” at the start. So now we have both predictions and true values in the same data frame and we can compare them easily.

Results and Discussion

We apply the model on 31 March 2023’s data and measure its performance. From the plots that we can observed that the curve stays still until 11am, at which point it starts to increase and starts to follow the real trend. The model first selected IDA1 until 8am, and then BM until 11am, after that we choose IDA2. This result shows that the model under estimate the cost of IDA1 energy prices during this time, whilst overestimating DAM and BM. This illustrates a drawback of a basic LSTM model, which relies subsequent decisions in the time series on memory from prior data. Therefore, the best plan of action would be to only help purchasing selections made after 11 a.m.

Conclusion & Recommendation

In conclusion, the model has been successfully created to help predict market prices according to the given variables, however, for this project, it still doesn't adequately address Energia's business issues with the necessary level of realism. The shortage of time and complexity of the project make us not have adequate time on this project, therefore, to optimize the model and give better predictions, we give the following recommendations:

1. Try to apply normalization to the data as pre-processing, since it will lead to significantly better results than the unnormalized data. (Hou et al., 2019)
2. We only focus on building a model that could give predictions rather than a perfect model, hence we can apply optimization to the model to improve the performance.

Reference

1. Bonthu, H. (2023) *Detecting and treating outliers: Treating the odd one out!*, *Analytics Vidhya*. Available at: https://www.analyticsvidhya.com/blog/2021/05/detecting-and-treating-outliers-treating-the-odd-one-out/ (Accessed: May 7, 2023).
2. Brownlee, J. (2019) *Time series data visualization with python*, *MachineLearningMastery.com*. Available at: https://machinelearningmastery.com/time-series-data-visualization-with-python/ (Accessed: May 7, 2023).
3. *Forecasting: Principles and practice (2nd ed)* (no date) *Chapter 6 Time series decomposition*. Available at: https://otexts.com/fpp2/decomposition.html (Accessed: May 7, 2023).
4. Hou, L., Zhu, J., T. Kwok, J., Gao, F., Qin, T. and Liu, T. (2019). Normalization Helps Training of Quantized LSTM. *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*. [online] Available at: https://proceedings.neurips.cc/paper/2019/hash/f8eb278a8bce873ef365b45e939da38a-Abstract.html [Accessed 7 May 2023].
5. *Outlier* (2023) *Wikipedia*. Wikimedia Foundation. Available at: https://en.wikipedia.org/wiki/Outlier (Accessed: May 7, 2023).
6. Pierre, S. (2022) *A guide to time series analysis in Python*, *Built In*. Available at: https://builtin.com/data-science/time-series-python (Accessed: May 7, 2023).