UNIVERSITY OF PIRAEUS
DEPARTMENT OF DIGITAL SYSTEMS

MSc in Climate Crisis and Information and Communication Technologies

Master Thesis

Evaluation of Machine Learning Models for Predicting the System Marginal Price of an Electricity System – Spanish SMP Day-Ahead forecasting

Athanasios Psaltis

Supervisor: Dr.M.Philippakis

Piraeus, 2023

Δήλωση Πνευματικών Δικαιωμάτων

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν. 1599/1986 και τα άρθρα 2,4,6 παρ. 3 του Ν. 1256/1982, η παρούσα Διπλωματική Εργασία με τίτλο :

“Αξιολόγηση μοντέλων μηχανικής μάθησης για την πρόβλεψη της οριακής τιμής συστήματος ηλεκτρικής ενέργειας - Περίπτωση Ισπανίας.’’ καθώς και τα ηλεκτρονικά αρχεία και οι πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας και αναφέρονται ρητώς μέσα στο κείμενο που συνοδεύουν και η οποία έχει εκπονηθεί στο Τμήμα Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιώς αποτελεί αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή / και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και μόνο.

Copyright (C) Αθανάσιος Ψάλτης, 2023, Αθήνα

Acknowledgements

I am profoundly thankful for the unwavering support and guidance provided by the individuals who have been instrumental in my master's thesis journey. Foremost among them is my thesis professor, Dr. Michael Philippakis, whose invaluable expertise and feedback have played a crucial role in shaping this thesis. My gratitude also extends to my friends and colleagues, whose moral support, encouragement, and engaging discussions made the research process both enjoyable and memorable. To my family, I am deeply appreciative of their love, encouragement, and unwavering belief in my abilities, serving as a solid foundation throughout my academic pursuit. Additionally, I extend my heartfelt thanks to all the participants and contributors to my research study; your willingness to share experiences and insights significantly enhanced the quality of my work. I am profoundly grateful to everyone who has been a part of this journey. Your contributions have been indispensable, and I deeply cherish your support.

Summary

Electricity market price prediction plays a crucial role in ensuring the efficient operation of power systems, helping market participants make informed choices and promoting sustainable energy consumption practices. This thesis offers an in-depth examination of three distinct time series forecasting models: AutoRegressive Integrated Moving Average (ARIMA), Facebook Prophet, and Light Gradient Boosting Machine (LightGBM). These models are applied to forecast Short-Term Market Prices (SMP) in Spanish energy system. The research starts by providing an overview of the electricity market in Spain, emphasizing the significance of precise price predictions for market players, grid operators, and policymakers. Additionally, the thesis underscores the significance of data analysis, manipulation, and preprocessing in preparing the SMP dataset for modeling. This involves techniques like data decomposition, stationarity testing, and Fourier transform to uncover underlying patterns and improve the quality of input data. The thesis provides a thorough review of each forecasting model, explaining their fundamental principles, strengths, and limitations. It then delves into the data preprocessing phase, illustrating how data decomposition techniques such as Seasonal Decomposition of Time Series (STL) can be used to separate trend, seasonality, and residual components. Stationarity tests like the Augmented Dickey-Fuller (ADF) test are employed to ensure the data is suitable for modeling, ensuring consistent statistical properties over time. A comparative analysis is conducted using historical SMP data to assess the performance of ARIMA, Facebook Prophet, and LightGBM models in terms of forecasting accuracy, computational efficiency, and robustness. Key metrics like Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are used to evaluate the predictive abilities of the models. The comparative analysis of the ARIMA, Facebook Prophet, and LightGBM models extends to assessing their ability to capture the decomposed components, seasonality, and volatility patterns within SMP data.Furthermore, the research explores the incorporation of Fourier transform to identify and model cyclical patterns in the SMP data, enhancing the models' capability to capture underlying periodic behaviors. This demonstrates the adaptability of the models to various data manipulation techniques. In conclusion, this thesis offers a comprehensive approach to electricity market price forecasting by integrating data analysis and manipulation techniques with ARIMA, Facebook Prophet, and LightGBM models for SMP price prediction in Spain. The results empower market participants and researchers to choose suitable forecasting strategies based on data characteristics, facilitating more informed decisions in the dynamic electricity market environment, while considering the inherent seasonality, stationarity, and cyclicality of the data.

Keywords: Machine Learning, SMP, Forecasting, Spanish Energy System, Sarima, FBProphet, LightGBM

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 System Marginal Price

In the realm of any market, the establishment of a price is contingent upon a consensus reached between the parties involved in the supply (sellers) and demand (buyers) of a specific quantity of a commodity or service[1]. Within wholesale electricity markets, this principle manifests as a harmonization of interests between sellers, exemplified by electricity producers and buyers, represented by entities such as retail companies catering to domestic or industrial consumers. This alignment culminates in the fixation of a price, expressed in EUR, for a designated volume of electricity, measured in MWh.

Conventionally, sellers exhibit a keen interest in optimizing the price to maximize their returns, while buyers are primarily inclined to minimize the price to mitigate costs. This synchronization of objectives is achieved through negotiation mechanisms, which may take the form of verbal or electronic exchanges, enabling both the buyer and the seller to arrive at a price point that is mutually acceptable.

In the context of market transactions between participants, the determination of a mutually agreeable price necessitates a structured approach. This can be accomplished through two primary methods:

Individualized, Bilateral Agreements: This approach entails the establishment of a bespoke agreement directly between two parties. On occasion, such agreements may involve intermediaries known as "brokers" who facilitate the negotiation process.

Auction Mechanisms: Alternatively, the pricing mechanism can involve the consideration of multiple buyers and sellers concurrently, congregated at a specific point in time through an auction. These auctions are typically conducted on platforms administered by power exchanges.

In the context of auctions, the process involves the aggregation of demand offers and the ranking of supply offers, arranged in ascending order from the least expensive to the most expensive. This ranking, often referred to as the "merit order," guides the selection of the cheapest selling offers, which are then matched with demand offers until the cumulative demand is fulfilled. In most cases within such auctions, all transactions are subsequently settled at the price determined by the last (or "marginal") selected selling offer. This pricing mechanism is known as marginal pricing or "pay-as-cleared."

Marginal pricing is a method used to price electricity in the wholesale market, more specifically in the day-ahead auction. The term System Marginal Price (SMP) denotes the price at which the wholesale electricity market establishes equilibrium between supply and demand during a specific timeframe and in a particular location[2]. It reflects the cost of electricity in real-time and hinges on the marginal cost incurred by the most expensive generator required to meet the prevailing demand, known as the marginal unit. Additionally, factors like transmission limitations, weather conditions, and fluctuations in demand exert influence on the SMP. In essence, the SMP signifies the instantaneous cost of electricity in a specific area and plays a pivotal role in

[1] https://efet.org/files/documents/20220222%20EFET_Insight_02_marginal_pricing.pdf

[2] https://www.lnrg.technology/2023/03/31/what-is-system-marginal-price-smp/

determining the profitability of power generation and demand-side management strategies.

The process of SMP forecasting in electricity markets involves predicting the SMP for future timeframes, ranging from a few hours or days ahead in the short-term to months or even years. This forecasting holds significance for various electricity market participants, including power producers, electricity suppliers, and large energy consumers, as it enables them to make informed decisions concerning their energy production, consumption, and trading activities.

Typically, SMP forecasting employs mathematical models and statistical algorithms that consider a multitude of factors affecting electricity prices, such as demand patterns, weather conditions, fuel costs, and the availability of generation resources. These models typically rely on historical data and are continuously updated as new information becomes available.

Accurate SMP forecasting empowers market participants to optimize their bidding strategies, manage risk exposure, and make well-informed investment choices. Furthermore, it plays a crucial role in the efficient and dependable operation of the electricity grid by allowing grid operators to anticipate and respond promptly and effectively to changes in electricity supply and demand.

Potential customers for SMP forecasting services in the electricity market encompass a wide array of stakeholders involved in energy production, consumption, and trading. Examples of potential customers include:

- Power Generators: These entities can utilize SMP forecasting to fine-tune their bidding strategies and make informed decisions regarding the scheduling and dispatch of their generation assets. Marginal pricing also allows generators to recover investment costs.
- Retail Electricity Providers: Retail electricity providers can leverage SMP forecasting to mitigate their exposure to price fluctuations and develop pricing strategies that adapt to shifts in market conditions.
- Large Energy Consumers: Entities such as industrial facilities and data centers can employ SMP forecasting to optimize their energy consumption patterns and reduce energy expenses.
- Energy Traders: Energy traders can utilize SMP forecasting to identify market trends, exploit arbitrage opportunities, and manage their market-related risks effectively.
- Grid Operators: Grid operators can rely on SMP forecasting to anticipate changes in electricity supply and demand, allowing them to proactively maintain grid stability and reliability.

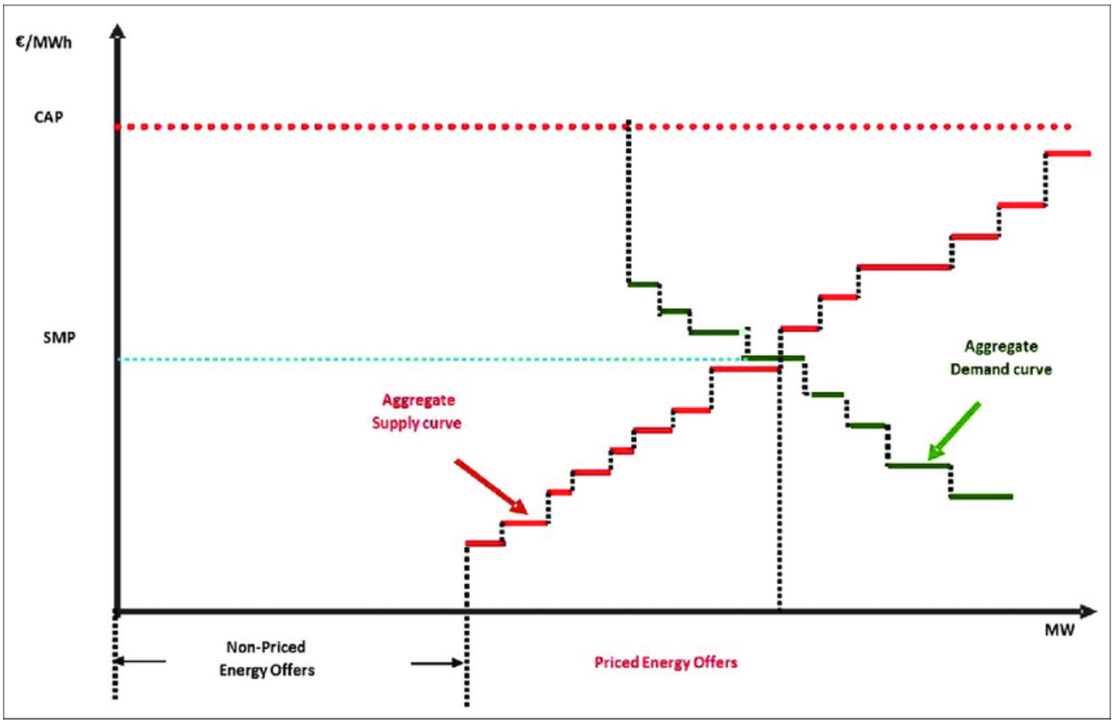


Figure 1-1 Determination of System Marginal Price (SMP), where the aggregate Supply and Demand curves intersect

Marginal pricing within the wholesale electricity market serves the purpose of allowing operators of electricity generation facilities with low operational expenses to gradually recoup a portion of their initial capital investments. Particularly notable among the beneficiaries of marginal pricing are operators of renewable energy generation units, as the operational costs associated with most renewable sources are nearly negligible. Consequently, these operators can realize significant portions of their investment outlays through market-driven mechanisms, thereby diminishing the reliance on financial assistance from public budgets.

With the introduction of adaptive public support mechanisms, financial aid is automatically scaled back during periods of elevated electricity prices. During such periods, the financing of renewable energy sources is shifted to electricity consumers, who contribute through the energy component of their utility bills rather than relying on tax-based funding.

As the prominence of renewable electricity generation continues to grow, there arises a heightened demand for alternative solutions when renewable sources are not producing power due to factors such as calm winds or cloudy skies. In these instances, alternative energy sources or strategies must be employed to either generate or conserve electricity. Marginal pricing ensures that backup technologies, such as peak electricity generation, battery storage, or the utilization of alternative energy sources like hydrogen, which may operate only for a limited number of hours per year, can at least partially recoup their initial investment costs. Moreover, it plays a pivotal role in incentivizing consumers to adapt their energy consumption patterns and serves as a signal for encouraging investments in novel technologies and services.

2 Spanish Energy System

Europe has created a system of pan-European auction in the day-ahead market since electrical networks are interconnected across the continent. As an integral part of the electrical energy production market, the day-ahead market, also called single day-ahead coupling (SDAC), aims to carry out electrical energy transactions by submitting selling and takeover bids for electrical energy on behalf of the market agents for the twenty-four hours of the following day. This market, coupled with Europe since 2014, is one of the crucial pieces in achieving the objective of the European Internal Energy Market.

Every day of the year at 12:00 CET is the day-ahead market session where prices and electrical energies are set for all across Europe for the twenty-four hours of the next day. The price and volume of energy at a specific hour are established by where supply and demand intersect, following the model agreed upon and approved by all of the European markets that is currently applied in Spain, Portugal, Germany, Austria, Belgium, Bulgaria, Croatia, Slovakia, Slovenia, Estonia, France, Holland, Hungary, Ireland, Italy, Latvia, Lithuania, Luxembourg, Finland, Sweden, Denmark, Norway, Poland, the United Kingdom, the Czech Republic, and Romania.

The buying and selling agents that are found in Spain or Portugal will present their bids to the day-ahead market through OMIE, which is the only NEMO (Nominated electricity market operator) in those countries. Their buying and selling bids are accepted based on their economic merit and depending on the available capacity for interconnection between price zones. If, at a certain time of day, the capacity for interconnection between two zones is sufficient to allow the flow of electricity resulting from negotiation, the price of electricity at that time will be the same in both zones. If, on the other hand, interconnection at that time is maxed out, at that moment the algorithm for setting prices results in a different price in each zone. The mechanism described for setting electricity prices is called market coupling.

The results from the day-ahead market, based on free contracting between buying and selling agents, represent the most efficient solution from an economic point of view, but given electricity's characteristics, it is also necessary for it to be viable from a physical point of view. As such, once these results are obtained, they are sent to the System Operator for validation with perspective on their technical viability. This process is known as managing the system's technical limitations and ensures that the market results can be technically accommodated on the transportation network. As such, results from the day-ahead market may be altered slightly because of the analysis of technical limitations done by the System Operator, giving rise to a viable daily program.

All the aforementioned benefits associated with marginal pricing contribute significantly to reducing the expenses incurred by end-consumers when purchasing electricity, as well as the suppliers who serve them. Within the various components encompassed in an end-consumer's electricity bill, the cost attributed to electricity itself is the sole component that has witnessed a decline over the past decade. Specifically, during the period from 2011 to 2020, the electricity component of the average EU household's bill experienced a noteworthy reduction of 10%.

However, when confronted with rapid increases in spot prices, as witnessed in 2021/2022, questions arise regarding whether end-consumers are directly exposed to these volatile day-ahead prices. In the case of most household consumers, fluctuations in day-ahead prices do not result in an immediate alteration of the retail price they encounter. This divergence arises from the fact that companies engage in electricity buying and selling activities well in advance of the actual generation and consumption of electricity. This proactive approach serves to manage the daily volatility associated with the marginal price in the day-ahead auction. Employing these strategies, commonly referred to as "hedging strategies," enables firms to offer fixed-price contracts to households or to adjust prices less frequently.

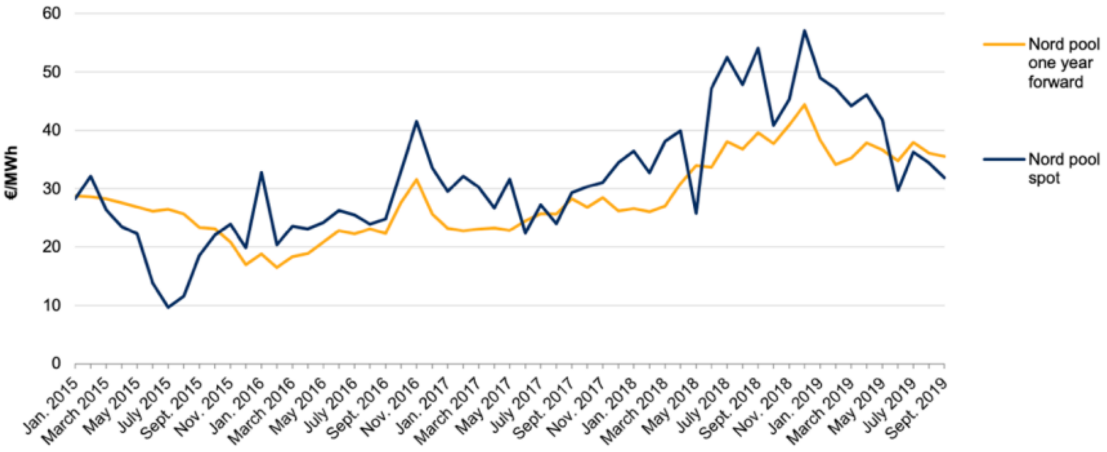


Figure 2-1: Day-ahead spot prices and year-ahead forward prices in the Nordic countries

In Spain, the average electricity price is considerably high. Although the marginal system European market is based on has a number of advantages as mentioned above, it is very sensitive to fluctuations in the prices of energy sources. When it becomes feasible to meet the entirety of the demand exclusively through renewable energy sources, the resulting price remains at a minimum. Conversely, when the utilization of gas or other non-renewable resources becomes imperative (on days when there is no sun or wind and RES production cannot reach the demand), the price experiences a substantial escalation.

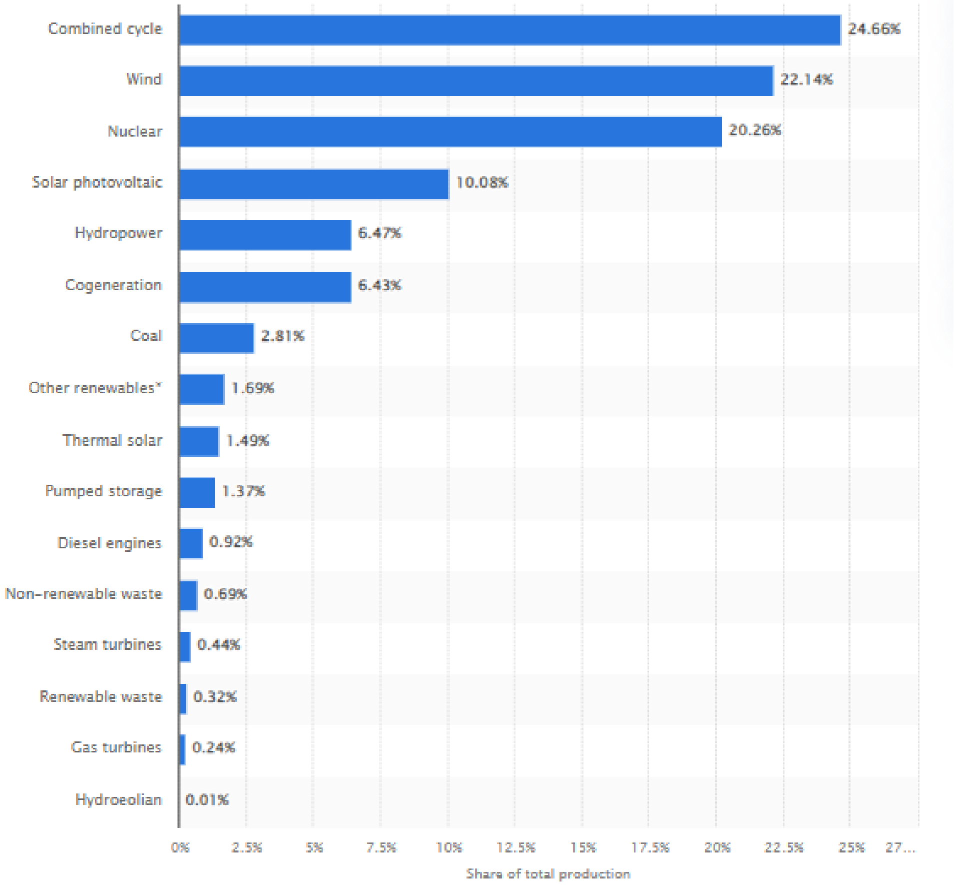


Figure 2-2 Share of electricity generation in Spain in 2022, by source

Historically, the most expensive price per MWh in Spain was recorded on the afternoon of 8 March 2022. In the 8pm time band that day, the price per MWh reached 700 €, due to the high gas prices as a result of the Ukrainian War. The average price per MWh that day was 544 €.

Electricity demand in Spain saw a 3.5% increase in 2021 but remained below the levels observed before the pandemic in 2019. However, in 2022, electricity demand dropped by 2.6% to approximately 243 terawatt-hours (TWh), primarily due to reduced consumption caused by high prices. Despite this decline in demand, power generation in 2022 rose by 6%, amounting to an increase of 16 TWh compared to the previous year, reaching a total of over 285 TWh. The introduction of Royal Decree-Law 10/2022, also known as the "gas cap" or "Iberian exception," on June 15, after its presentation to the European Council on March 26, played a significant role in these developments. In 2022, the Iberian exception led to lower wholesale electricity prices in Spain but at the expense of increased electricity exports and a 25% rise in gas-fired generation, equivalent to 18 TWh more than in 2021.Coal-fired power generation increased by approximately 50% (3 TWh) in 2022 but still represented only 3% of the total electricity generation. On the other hand, electricity generation from solar photovoltaic (PV) sources grew by approximately 23% year-on-year, amounting to an additional 5 TWh in 2022. Solar PV achieved an installed capacity of 20 gigawatts (GW), surpassing hydroelectric power as the technology with the third-highest installed capacity. An unusual event occurred during the year when a significant drop in solar PV electricity output was observed in March due to haze from the Sahara Desert, resulting in a 14%

decrease (-0.2 TWh) compared to the same month in the previous year.Furthermore, record drought conditions in Spain caused hydroelectric generation to decrease by more than 30% year-on-year in 2022. The cap on gas prices led to a further year-on-year decline in power generation from combined heat and power (CHP) plants, amounting to a 23% decrease (3 TWh) for the January-August 2022 period. CHP plants were not eligible for the cap until the Spanish Government announced it on September 6. Additional measures introduced in the recent More Energy Security Plan aimed to protect consumers, including electricity-intensive industries. These measures included an 80% reduction in access charges and a temporary decrease in VAT for gas prices from 21% to 5%. Additionally, a new mechanism was established to manage excess electricity consumption. Looking ahead, it is expected that electricity usage will decrease by approximately 0.5% in 2023, followed by a growth rate of 1% annually in both 2024 and 2025. Over this three-year period, total renewable energy output is forecasted to increase by more than 15% annually, accompanied by a moderate rise in electricity imports. As a result of lower utilization of thermal generation plants due to increased renewable energy output and reduced gas-fired exports, CO2 emissions are projected to decrease by nearly 60% in 2025 compared to 2022 levels.

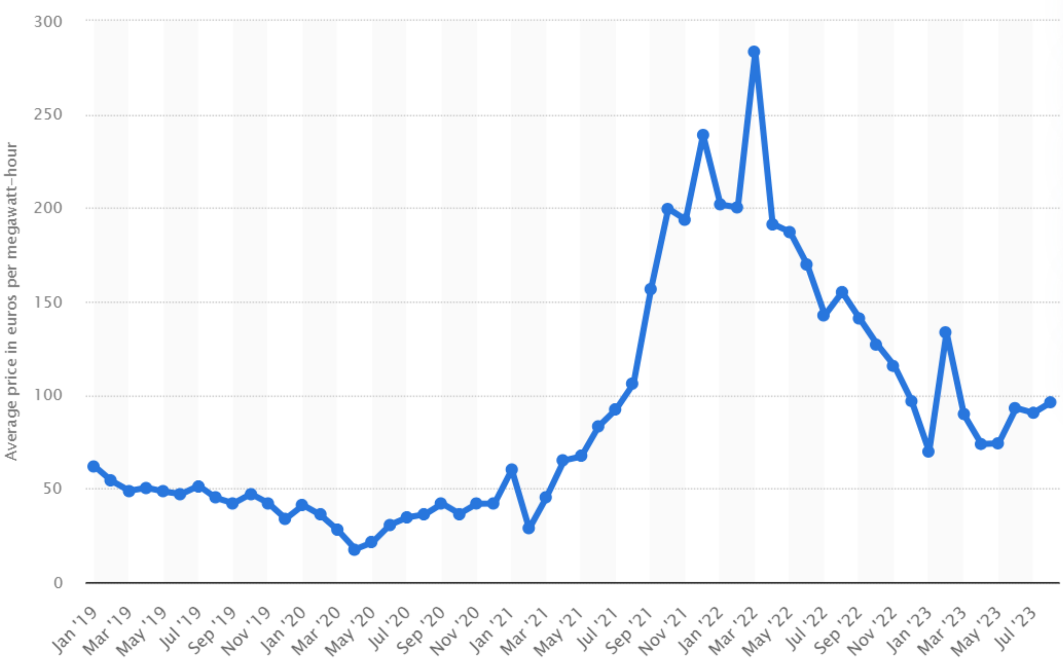
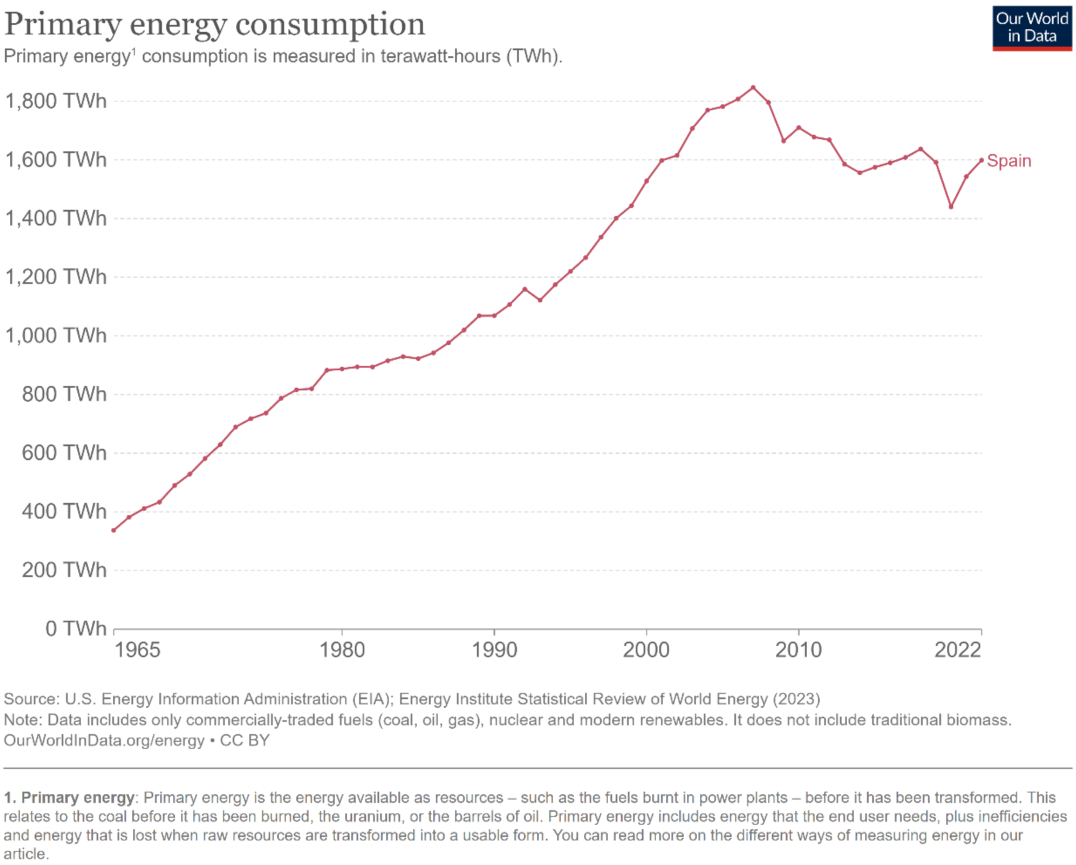


Figure 2-3: Average monthly electricity wholesale price in Spain from January 2019 to August 2023 (€/MWh)



Primary energy consumption

Primary energy[1] consumption is measured in terawatt-hours (TWh).

Source: U.S. Energy Information Administration (EIA); Energy Institute Statistical Review of World Energy (2023)
Note: Data includes only commercially-traded fuels (coal, oil, gas), nuclear and modern renewables. It does not include traditional biomass.
OurWorldInData.org/energy • CC BY

1. **Primary energy**: Primary energy is the energy available as resources – such as the fuels burnt in power plants – before it has been transformed. This relates to the coal before it has been burned, the uranium, or the barrels of oil. Primary energy includes energy that the end user needs, plus inefficiencies and energy that is lost when raw resources are transformed into a usable form. You can read more on the different ways of measuring energy in our article.

Figure 2-4 Primary energy consumption (TWh)

3 Research problem

Accurate SMP prediction in Spain is a challenging subject since it is dependent on a variety of factors such as weather, electricity demand, electricity generation, and market prices. The main problems on accurately predicting the SMP consists of the following factors:

Power Grid Complexity: The electricity grid is a highly complex and dynamic system. It is made up of various generators, loads, transmission lines, and other devices that interact in a non-linear fashion. Weather conditions, fuel pricing, demand, and generating availability all have an impact on the power grid's behavior. Because of this intricacy, properly predicting the SMP is difficult.

Lack of Data: Predicting the SMP accurately necessitates a considerable amount of data, such as historical pricing, generator availability, and transmission constraints. Unfortunately, this data is not always easily available, and there may be gaps or inconsistencies in the data that impair the prediction's accuracy.

Uncertainty: Future forecasting involves a great level of uncertainty. Unexpected outages, fluctuations in demand, and unexpected weather occurrences can all have an impact on the pricing. Furthermore, market players' behavior may be unpredictable, adding to the uncertainty.

Market Design: The design of the power market can also affect SMP prediction accuracy. Certain markets, for example, may have intricate bidding processes or other regulations that make predicting the SMP problematic.

Model Complexity: SMP predictive models can be highly complicated, requiring sophisticated algorithms and vast volumes of data. This can make it harder to understand and interpret the results, as well as increase the likelihood of model errors or biases.

The system marginal price demonstrates the following attributes that bear as a result the difficulty of making predictions about its value:

1. **Volatility**: The SMP can experience significant and rapid changes, especially in markets with high renewable energy integration or during periods of high demand. Predicting these fluctuations accurately is difficult due to the volatility of influencing factors.
2. **Nonlinearity**: The relationship between the SMP and its influencing factors is often nonlinear. Small changes in these factors can result in disproportionate effects on the SMP, making accurate predictions more complex.
3. **Uncertainty**: Predicting the SMP involves dealing with inherent uncertainties related to future market conditions, policy changes, and unforeseen events. These uncertainties make it challenging to provide accurate forecasts.
4. **Interdependencies**: The SMP is influenced by multiple interconnected factors such as fuel prices, weather patterns, generation mix, and market participant behaviors. These complex interdependencies introduce challenges in accurately modeling and predicting the SMP.
5. **Market Manipulation**: The electricity market can be subject to market manipulation and strategic behaviors by participants, which can distort the SMP and add complexity to its prediction.
6. **Regulatory and Policy Influences**: Changes in regulations, market rules, or government policies can significantly impact the SMP. Predicting the SMP requires understanding and anticipating these influences, which can be challenging.

Addressing these characteristics requires advanced analytical techniques, sophisticated modeling approaches, and robust data analysis methods to improve the accuracy of SMP predictions.

4 Research objectives

The primary goal of this study is to compare machine learning models in forecasting SMP in Spain utilizing multiple input data. The following are the specific research objectives:

- To assess the performance of machine learning methods.
- To assess the performance of machine learning models in comparison to traditional forecasting approaches.
- To inform energy market participants and policymakers about the elements that influence SMP in Spain.

5 Previous studies on SMP prediction

Several studies have been undertaken in Spain utilizing various approaches and strategies to anticipate the System Marginal Price (SMP). Álvaro Romero, et al. (2018) compared several machine learning models, in order to decide that randoms forests produced the least mean absolute error and thus being the most accurate machine learning model in terms of predicting the SMP. In a more recent study, Garcia-Martos et al. (2021) employed machine learning algorithms to anticipate the SMP in the Spanish day-ahead power market, considering both market and weather data. The authors utilized a combination of gradient boosting and random forest models and discovered that their method accurately predicted the SMP.

6 Methodology

There are multiple commonly used models for predicting the system marginal price (SMP) in electricity markets.

1. **Statistical Time Series Models**: These models utilize historical SMP data to forecast future values. They employ techniques such as autoregressive integrated moving average (ARIMA), seasonal ARIMA (SARIMA), and exponential smoothing models (e.g., Holt-Winters). By identifying patterns and trends in the SMP data, these models make predictions.
2. **Regression Models**: Regression models incorporate historical SMP data and other relevant variables like weather conditions, demand forecasts, fuel prices, and generation capacity to predict the SMP. Multiple linear regression, nonlinear regression, or machine learning algorithms can be used to construct regression models.
3. **Artificial Neural Networks (ANNs)**: ANNs are machine learning models that can capture intricate relationships between input variables and SMP. They consist of interconnected nodes (neurons) organized in layers. ANNs are trained using historical SMP data and other relevant variables to make predictions.
4. **Ensemble Methods**: Ensemble methods combine predictions from multiple models to enhance accuracy. Techniques such as ensemble averaging, bagging, and boosting are employed to aggregate predictions from different

models, which may include regression models, time series models, or machine learning models.

5. **Support Vector Regression (SVR)**: SVR is a machine learning technique that employs support vector machines for regression analysis. It can capture nonlinear relationships between input variables and SMP. SVR models seek to find a hyperplane that optimally fits the data while controlling the error margin.
6. **Long Short-Term Memory (LSTM) Networks**: LSTM networks are a type of recurrent neural network (RNN) that effectively model sequential data. They have been successfully applied to time series forecasting, including SMP prediction. LSTM networks can capture long-term dependencies and patterns in SMP data.

It's important to note that the choice of prediction model depends on various factors, such as the characteristics of the SMP data, available features and variables, desired forecast horizon, and required level of accuracy. Different models may be more suitable for specific situations and datasets, and practitioners often explore and compare multiple models to determine the most appropriate one for SMP prediction.

7 Time series

7.1 Definition and Characteristics of Time Series Data

Time series data refers to a collection of observations recorded over time at regular intervals. It is characterized by its sequential nature, where each observation is associated with a specific time stamp. Time series data can be univariate, involving a single variable measured over time, or multivariate, involving multiple variables measured simultaneously. Examples of time series data include stock prices, weather measurements, economic indicators, and patient vital signs.

7.2 Importance of Time Series Analysis

Time series analysis plays a crucial role in various fields due to its ability to uncover underlying patterns, detect trends, and make predictions. In finance, for instance, it helps investors analyze stock market data to make informed decisions. In economics, it assists in understanding and forecasting economic indicators, such as GDP growth and unemployment rates. In meteorology, time series analysis enables weather forecasting and climate modeling. Moreover, time series analysis aids in analyzing and predicting phenomena in social sciences, engineering, healthcare, and many other domains.

7.3 Time series analysis

The first produced graph depicts the mean hourly SPM value, for each day of the week, for the time period of the studied timeframe. (1/1/2020 – 31/12/2022).

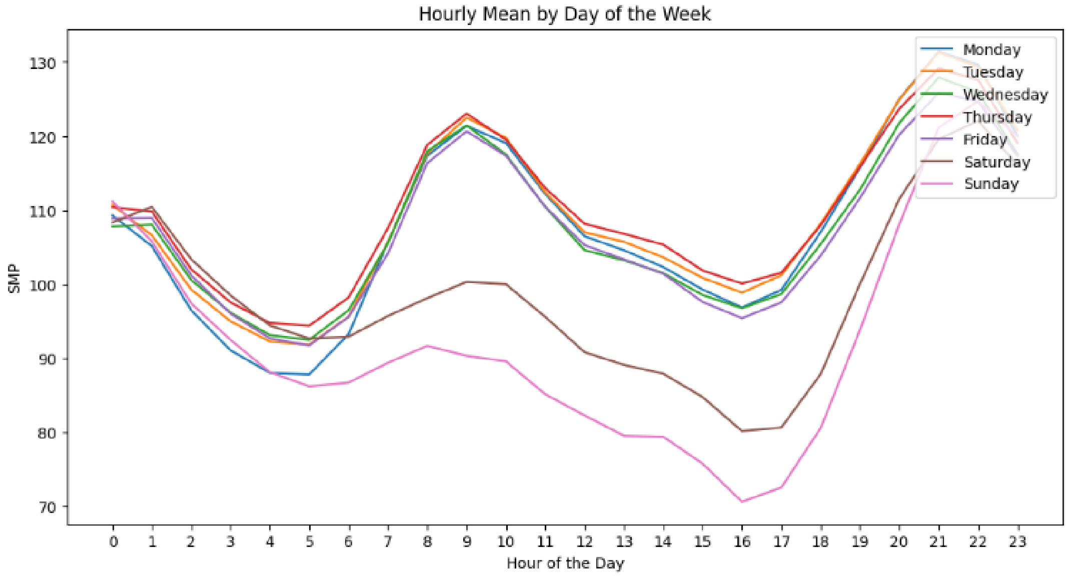


Figure 7-1 Mean hourly SPM value

The examination of price time series data reveals a significant recurring pattern, primarily driven by demand-related factors. Firstly, it is easily observed that Sundays and Saturdays produce lower SMP prices that the other weekdays. Moreover, the SMP price seems to reach two peaks during each day, the first one at around 9 am and the second, half a day later at around 9 pm. Between these peaks, SMP reaches its lowest price at around 5 am in weekdays and 4 pm during the weekend. These values can be justified by the fact that the prices follow the trend of the electricity demand.

Below, histograms for each day of the week for the studied time period, are depicted.

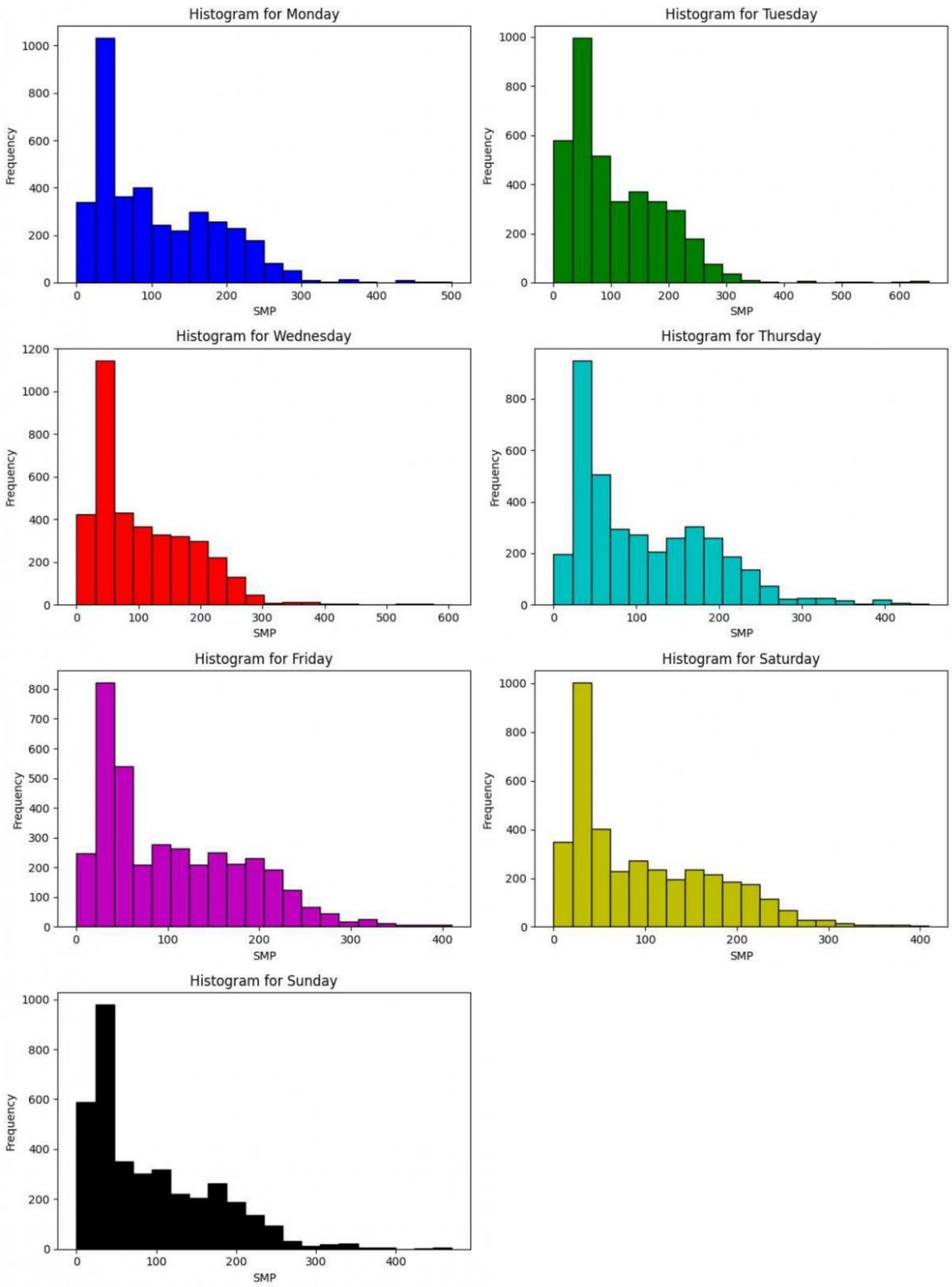


Figure 7-2 SMP frequency for each day

A distinct common pattern is observed for every day of the week about the SMP price contribution. Every day has most of the prices around €40. However, on weekdays, there are numerous instances where prices exceed €100, whereas on Saturdays and

Sundays, such instances are significantly fewer. Additionally, Sundays exhibit a higher proportion of hours with prices below €40.

8 Data Manipulation

The data set used consists of the hourly rates of the Spanish System Marginal Price from 1/1/2020-31/12/2022. In total, the data size is 27400 entries.

Firstly, the cumulative time series is presented below.

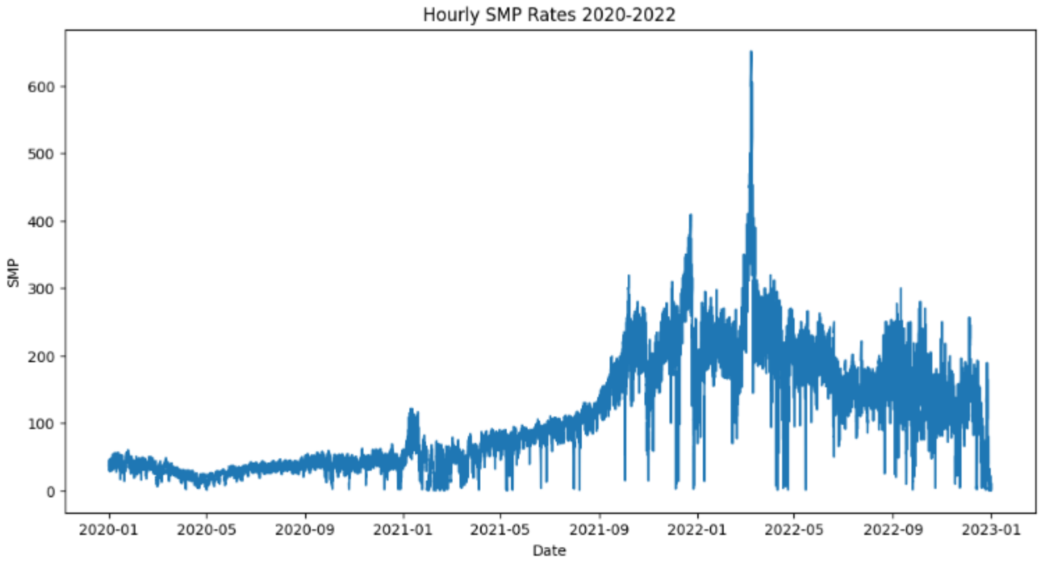


Figure 8-1 Hourly SMP Rates

By a mere observation of the time series, the impact of the Ukrainian war becomes evident in the energy market prices across the entirety of Europe as well as the Iberian Peninsula.

8.1 Time Series Components

A time series can be decomposed into several components, each contributing to its overall behavior:

8.1.1 Trend

The trend component embodies the prevailing trajectory of the data across an extended timeframe. It encapsulates the sustained and systematic movement over the long term, signifying whether the series demonstrates an ascending, descending, or inconsequential trend. Trends may manifest as linear patterns characterized by consistent upward or downward shifts, or they may assume nonlinear forms, characterized by intricate patterns. Comprehending the trend facilitates the recognition of the fundamental behavior and furnishes practical insights for prognosticating forthcoming values.

8.1.2 Seasonality

Seasonality pertains to the systematic repetition of patterns within a designated timeframe. These recurring patterns are subject to influence by temporal factors such as specific times of day, days of the week, months of the year, or other fixed intervals. Seasonality may stem from natural occurrences, cultural events, holidays, or other cyclic phenomena. The identification and comprehension of seasonality hold paramount importance in the accurate forecasting and prediction of anticipated cyclical patterns. To discern seasonality, various established methodologies are employed, including techniques such as seasonal subseries plots, autocorrelation analysis, or Fourier analysis. Once seasonality is detected, suitable approaches such as seasonal decomposition of time series (e.g., STL decomposition) or seasonal adjustment can be implemented to capture and incorporate the seasonal patterns appropriately.

8.1.3 Cyclical Variations

Cyclical variations encompass prolonged fluctuations that lack fixed temporal intervals. These patterns can persist over multiple years and are subject to the influences of economic, political, or business cycles. Diverging from seasonality, cyclical variations lack a predetermined duration and may display irregular oscillations. Scrutinizing cyclical patterns facilitates comprehension of broader economic or societal forces exerting an impact on the time series. Detecting cyclical patterns poses challenges and frequently necessitates advanced statistical techniques such as spectral analysis, wavelet analysis, or filtering methods like the Hodrick-Prescott filter or Baxter-King filter.

8.1.4 Random or Residual Fluctuations

Random or residual fluctuations denote the unpredictable and irregular constituent of a time series that eludes explanation by other components such as trend, seasonality, and cyclical variations. These fluctuations stem from diverse sources, including measurement errors, unforeseen occurrences, or inherent randomness inherent in the observed system. Incorporating the random component holds crucial significance in precise modeling and forecasting endeavors, as it encapsulates the intrinsic uncertainty within the data. Analyzing the random component frequently entails employing techniques like residual analysis, autocorrelation and partial autocorrelation plots, or statistical tests to assess randomness.

By comprehensively grasping and effectively modeling each component within a time series, analysts can gain invaluable insights, achieve accurate predictions, and derive meaningful interpretations from the data.

8.2 Stationarity

Stationarity is a crucial assumption when analyzing time series data. It refers to consistent statistical properties that remain unchanged over time, including a constant mean, constant variance, and an autocovariance structure dependent only on the time lag. On the other hand, non-stationary time series display varying statistical properties and often exhibit trends, seasonal patterns, or changing variances. Analyzing and modeling non-stationary data can be challenging as it introduces complexities and hinders pattern identification.

For a time series to be considered stationary, it must meet the following criteria:

Constant Mean: The average value of the time series remains consistent over time. This can be represented mathematically as:

$$E(X_t) = \mu$$

where $E(X_t)$ denotes the mean of the time series at time point t, and μ is a constant.

Constant Variance: The spread or variability of the time series remains constant over time. Mathematically, it can be expressed as:

$$Var(X_t) = \sigma^2$$

where $Var(X_t)$ represents the variance of the time series at time point t, and σ^2 is a constant.

Constant Autocovariance (Autocorrelation): The relationship between any two observations in the time series depends solely on the time difference between them (lag), rather than their specific time points. In simpler terms, the covariance between $X_t$and X_s(where t and s are time points) is equivalent to the covariance between X_{t+h} and X_{s+h} where h represents the time difference or lag. Mathematically, it can be written as:

$$Cov(X_t, X_s) = Cov(X_{t+h}, X_{s+h})$$

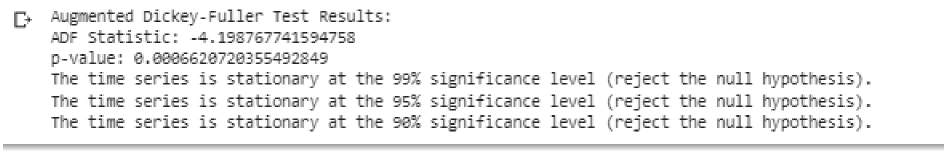
with Cov(.) denoting the covariance.

To make a non-stationary time series suitable for conventional modeling techniques, various transformations can be applied. Differencing, for instance, involves computing the differences between consecutive observations to remove trends and stabilize the mean. Logarithmic transformations are also useful in addressing exponential growth or shrinkage observed in non-stationary series. Achieving stationarity through these transformations allows analysts to effectively use traditional time series models like ARIMA to capture underlying patterns and dependencies. Stationarity is vital for accurate modeling, forecasting, and drawing meaningful conclusions from time series analyses.

Several statistical tests are commonly used to assess stationarity. Three widely recognized methods are:

1. **Augmented Dickey-Fuller (ADF) Test**: This test examines whether the autoregressive model of the time series has a unit root equal to 1. The null hypothesis assumes non-stationarity, and the resulting p-value is compared to a significance level (typically 0.05) to determine stationarity.
2. **Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test**: This test evaluates trend stationarity in a time series. The null hypothesis assumes stationarity, and if the computed p-value exceeds the chosen significance level, stationarity is indicated.
3. **Phillips-Perron (PP) Test**: This test, similar to ADF, considers serial correlation and heteroscedasticity. It computes a test statistic and p-value to determine stationarity.

To employ the Augmented Dickey-Fuller (ADF) test in Python for the examination of time series data, the adfuller() function from the statsmodels library can be employed. The result of the whole time series tested can be shown below:



```
Augmented Dickey-Fuller Test Results:
ADF Statistic: -4.198767741594758
p-value: 0.0006620720355492849
The time series is stationary at the 99% significance level (reject the null hypothesis).
The time series is stationary at the 95% significance level (reject the null hypothesis).
The time series is stationary at the 90% significance level (reject the null hypothesis).
```

Consequently, our time series can be considered as stationary, which will enhance the forecasting ability of the methods used below in this thesis.

8.3 Autocorrelation and Partial Autocorrelation

Autocorrelation pertains to the correlation between a time series and its lagged versions, indicating the extent to which an observation depends on its past values. Autocorrelation plots, such as the autocorrelation function (ACF) plot, are widely employed for discerning patterns in the correlation structure of a time series. On the other hand, Partial Autocorrelation measures the direct relationship between two observations while omitting the influence of the intervening observations. It aids in identifying the lag at which the correlation is most pronounced, facilitating the selection of appropriate autoregressive (AR) models.

Below, the graphs of autocorrelation and partial autocorrelation are produced and depicted, using the *plot_acf* and *plot_pacf* functions from the *statsmodels* library. Note that, only the first 100 data points were used for clarity and coherence reasons.

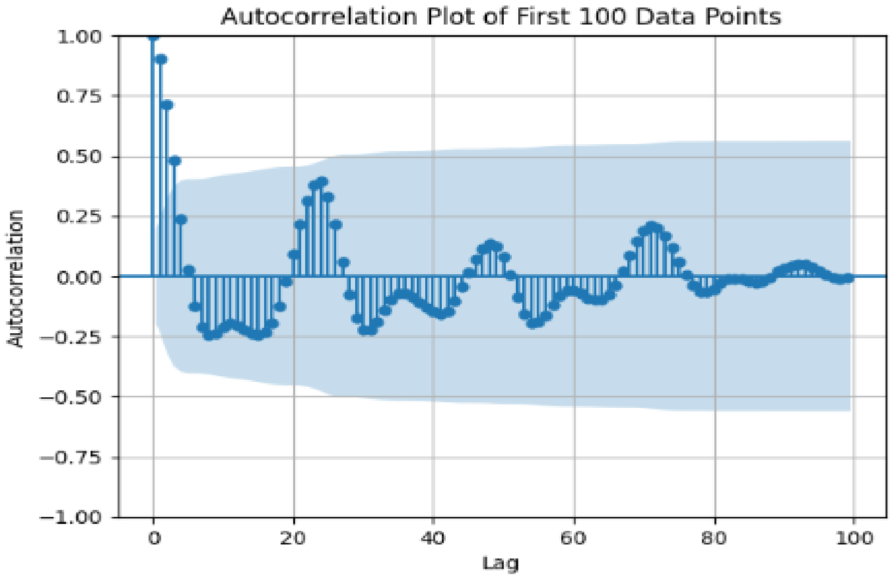


Figure 8-2 Autocorrelation Plot of First 100 Data Points

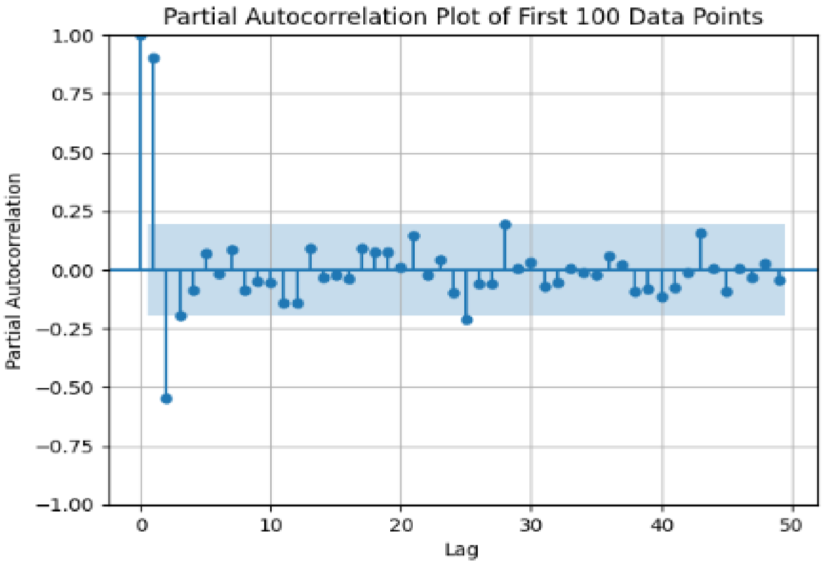


Figure 8-3 Partial Autocorrelation Plot of First 100 Data Points

In the realm of time series analysis, a lag pertains to the temporal interval or the quantity of observations that separate the current data point from a preceding data point. It represents the temporal delay or displacement between two consecutive values in a time series. When performing autocorrelation or partial autocorrelation computations, we examine the association between a data point and its lagged versions. The lag is specified as the number of time steps or observations considered during the correlation calculation. To illustrate, suppose we possess an hourly time series dataset and wish to examine the autocorrelation at lag 1. This implies scrutinizing the correlation between a data point and the value recorded one hour prior to it. Alternatively, if we set the lag to 2, we would be assessing the correlation between a data point and the value that occurred two hours earlier, and so forth. The concept of lag holds great significance in comprehending the temporal dependencies and patterns within a time series. It aids in identifying the degree to which past observations influence the current observation and assists in selecting appropriate models for time series analysis.

The inclusion of the blue cone, also referred to as the confidence interval cone, is a common practice in autocorrelation and partial autocorrelation plots. Its purpose is to illustrate the level of statistical uncertainty associated with the estimated autocorrelation and partial autocorrelation coefficients. As the lag increases, the cone expands, indicating an increasing degree of uncertainty in the estimated coefficients as the time lag grows. The cone's width is indicative of the variability in the estimated coefficients across various lags. Generally, the construction of the cone assumes that the autocorrelation or partial autocorrelation coefficients conform to a normal distribution. The width of the cone corresponds to a specific level of confidence, such as a 95% confidence interval. When a coefficient falls outside the cone, it implies that the coefficient at that lag is statistically significant, suggesting a stronger correlation compared to what might be expected by chance alone. Conversely, coefficients falling within the cone are generally considered statistically insignificant. In summary, the blue cone observed in autocorrelation and partial autocorrelation plots visually represents the uncertainty surrounding the estimated coefficients at different lags, assisting in the identification of statistically significant correlations.

Based on the above, we can observe on the produced graphs, that the first point has a value of 1, which is expected because the current value should always correlate with the current value. The second point almost scores 0.9 in the y axis, which means that it is described by the previous point at about 90%. The third point strikes almost 0.6 in the y axis, meaning that current values will influence the values in three hours by 60%. We can also observe that the correlation can be both positive and negative, as illustrated in both graphs. Lastly, besides the first four, the rest of the points remain inside the blue cone which can be interpreted on the following ways:

No autocorrelation: The current observation in the time series is not correlated with its lagged versions at any lag. The data points are essentially independent of each other, indicating a lack of any predictable patterns or dependencies.

Random or white noise: The time series may exhibit random fluctuations or noise, where each data point is unrelated to its previous observations. This can occur when the data is purely stochastic and lacks any underlying trend or pattern.

The partial autocorrelation plot specifically quantifies the extent of the direct association between two consecutive observations in a time series while effectively accounting for the influence of the intermediate observations. It assists in pinpointing the lag value where this direct correlation is most pronounced and informative.

8.4 Decomposition

Time series decomposition is a fundamental technique in the field of statistics and data analysis, used to understand the underlying patterns and trends within a time series data. The process involves breaking down a time series into its individual components, typically including trend, seasonality, and noise. The forecasting aspect of analyzing time series is closely tied to decomposition. Once a time series is broken down into its distinct elements (trend, seasonality, and residual), each component can be separately modeled to predict future values. Trend forecasting involves using various modeling techniques, such as linear regression, exponential smoothing, or autoregressive integrated moving average (ARIMA) models, to forecast the long-term direction or tendency of the time series. These models capture the underlying growth or decline pattern and project it into the future. Seasonal forecasting entails modeling and predicting the recurring patterns or cycles captured in the seasonal component derived from decomposition. Techniques like seasonal decomposition of time series (STL), seasonal ARIMA (SARIMA), or seasonal exponential smoothing can be employed. By forecasting the seasonal component, we can anticipate future seasonal patterns and adjust our predictions accordingly. Residual forecasting focuses on capturing any remaining irregular or random fluctuations in the time series that are not explained by the trend or seasonality. Methods like autoregressive integrated moving average (ARIMA) modeling or other advanced techniques are used to forecast the residual component and account for any patterns or anomalies that may influence future values. Once each component has been forecasted individually, the predicted values can be combined by summing the trend forecast, seasonal forecast, and forecasted residual. This generates an overall forecast for the future values of the time series. By decomposing the time series and employing suitable forecasting techniques for each component, we can achieve more accurate and dependable predictions. Decomposition enables us to capture and model the different patterns and characteristics of the time series, leading to improved forecasting accuracy. It allows us to consider the long-term trend, recurring seasonal patterns, and any remaining irregularities in the data when making forecasts.

8.4.1 Additive Decomposition

The prevalent method for decomposing a time series is known as additive decomposition. This method assumes that the observed series can be represented as the combination of three elements: trend, seasonality, and residual and is illustrated by the below formula.

$$Y_t = T_t + S_t + \varepsilon_t$$

where:

- Y_t is the observed value at time t.
- T_t represents the trend component, which captures the long-term behavior or overall direction of the time series.
- S_t is the seasonal component, representing the regular, repeating patterns within the data.
- ε_t is the residual component, accounting for random noise or unexplained variation.

Statsmodels library along with the seasonal decomposition function will be used for the decomposition. The output is depicted below:

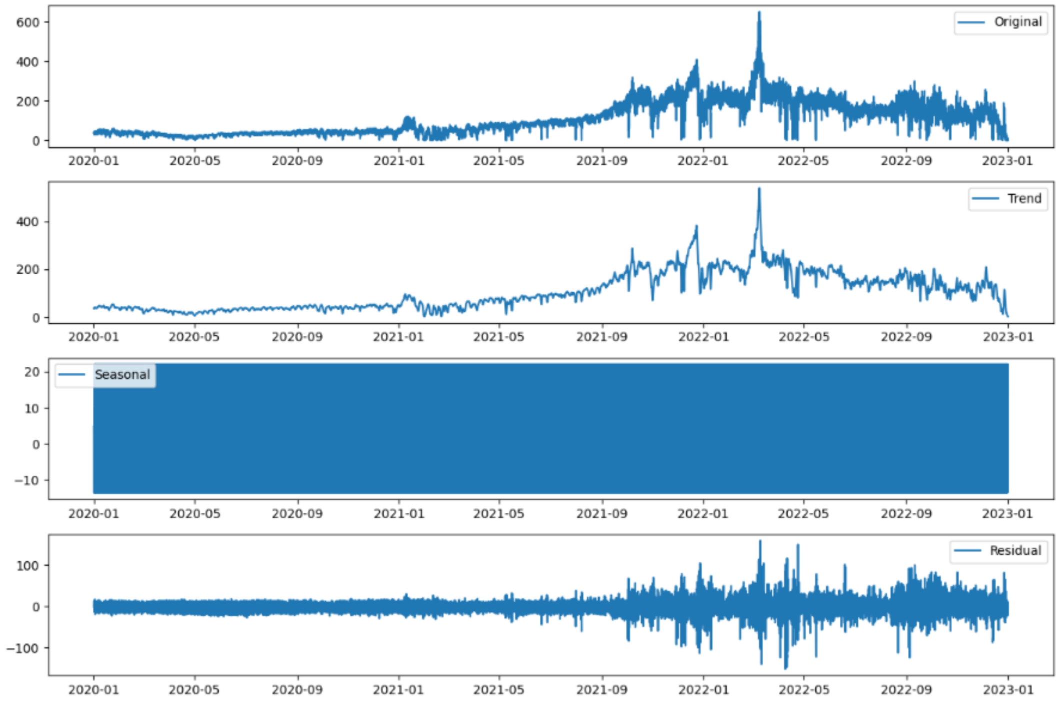


Figure 8-4 Additive Decomposition of the time series

As it becomes obvious, not many conclusions can be drawn from the seasonality graph.

8.4.2 Multiplicative Decomposition

The multiplicative decomposition model assumes that the components interact multiplicatively:

$$Y_t = T_t \times S_t \times \varepsilon_t$$

Since the studied time series has zero values, we cannot use multiplicative decomposition, as it will not produce any fruitful results.

8.4.3 Seasonal Decomposition of Time Series (STL)

STL (Seasonal and Trend decomposition using Loess) decomposition method combines locally weighted regression (Loess) with seasonal decomposition to overcome limitations of traditional decomposition methods. The algorithm estimates the trend component by applying Loess smoothing, captures the seasonal component by removing the trend, and calculates the residual component. STL is flexible, robust to outliers, and adaptable to different types of seasonality. The trend component represents the long-term direction, the seasonal component reflects recurring patterns, and the residual captures random fluctuations. STL decomposition is used for seasonal adjustment, forecasting, and anomaly detection in time series analysis. Below, the STL decomposition is depicted using the STL function form the *statmodels* library.

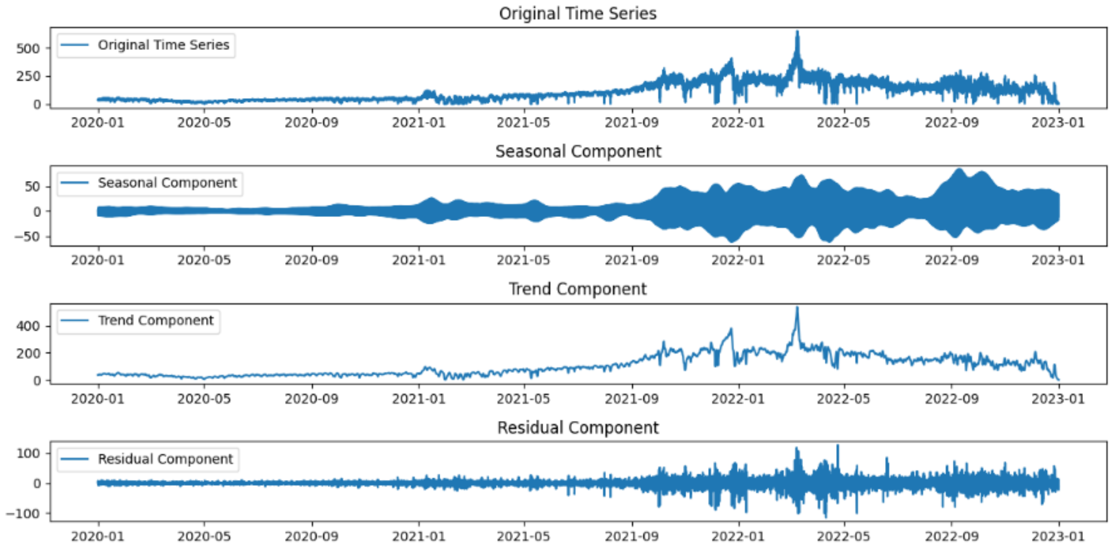


Figure 8-5 STL Decomposition of time series

8.5 Fourier Transform

Another way of trying to analyze the seasonality of a time series is through Fourier Transform. The Fourier Transform converts a time-domain signal into its frequency-domain representation. By applying the Fourier Transform to a time series, you can identify the dominant frequencies or periodic components, which correspond to the seasonality patterns in the data. In order to implement the Fourier Transform in a time series, the latter must be stationary. By employing Fourier Transform, we can gain insights into the seasonality of a time series by revealing its underlying frequency components. It allows the identification and analysis of dominant periods or cycles that contribute significantly to the observed seasonality. Below, the graph of the whole time series' frequencies is presented.

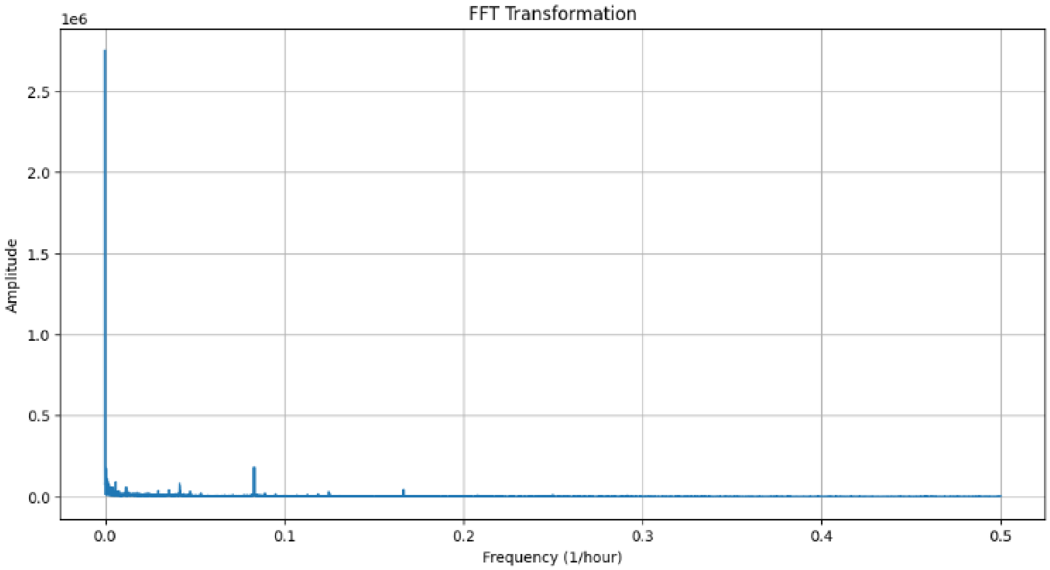
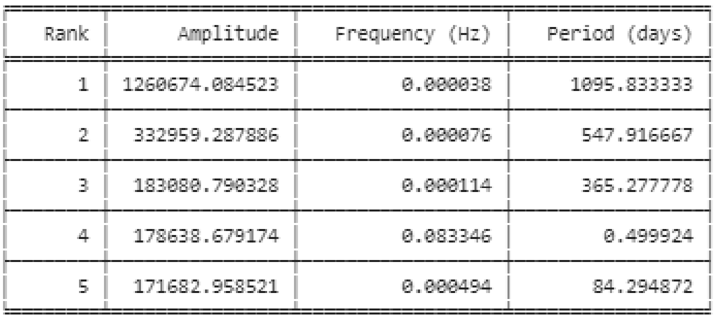


Figure 8-6 FFT Transformation implemented on time series

The prevailing frequencies, also known as dominant frequencies or prominent frequencies, are the frequencies with the highest amplitudes or power in each signal or time series. They play an essential role in understanding the underlying patterns, behaviors, and characteristics of the signal. The prevailing frequencies produced from the Fourier fast transfer (in the form of days) are the following:



Rank	Amplitude	Frequency (Hz)	Period (days)
1	1260674.084523	0.000038	1095.833333
2	332959.287886	0.000076	547.916667
3	183080.790328	0.000114	365.277778
4	178638.679174	0.083346	0.499924
5	171682.958521	0.000494	84.294872

Figure 8-7 Prevailing frequencies along with the respective amplitudes and periods (days)

The prevailing frequencies in a time series can provide insights into the presence of seasonality. Seasonality refers to a recurring pattern or variation that repeats over regular intervals of time, such as daily, weekly, monthly, or yearly cycles. The prevailing frequencies directly correspond to the periods or lengths of the seasonal cycles in the time series. The amplitudes or magnitudes associated with the prevailing frequencies indicate the strength or intensity of the seasonal components. Higher amplitudes imply more pronounced and significant seasonality, while lower amplitudes suggest weaker or less influential seasonal patterns or even residuals and white noise. Consequently, in our case, the prevailing period at almost 1095 days, which can be translated to 3 years, does not provide an insight in our time series, since it is 3 years long. However, 3 significant frequencies are detected in the tested dataset. Two long term seasonalities, 1,5 year and 1 year respectively, and a short term, half a day.

Lastly, using the *scipy* library, we produced the spectrogram of our time series. A spectrogram is a graphical representation that illustrates the distribution of frequency components within a signal as they vary over time. It serves as a fundamental tool in signal processing and analysis for the examination of the frequency characteristics of dynamic signals.

The core elements of a spectrogram encompass:

- **Temporal Axis**: The horizontal axis on a spectrogram is dedicated to time, with the signal segmented into discrete time intervals, either overlapping or non-overlapping. Each segment corresponds to a specific temporal window.
- **Frequency Axis**: Representing frequency, the vertical axis spans a range of frequencies from low to high. The precise frequency range depends on the application and the characteristics of the signal under examination.
- **Intensity or Coloration**: The intensity or color at each point within the spectrogram denotes the magnitude or power of the corresponding frequency components at a particular moment in time. Common representations include grayscale shading or color maps, reflecting power levels.

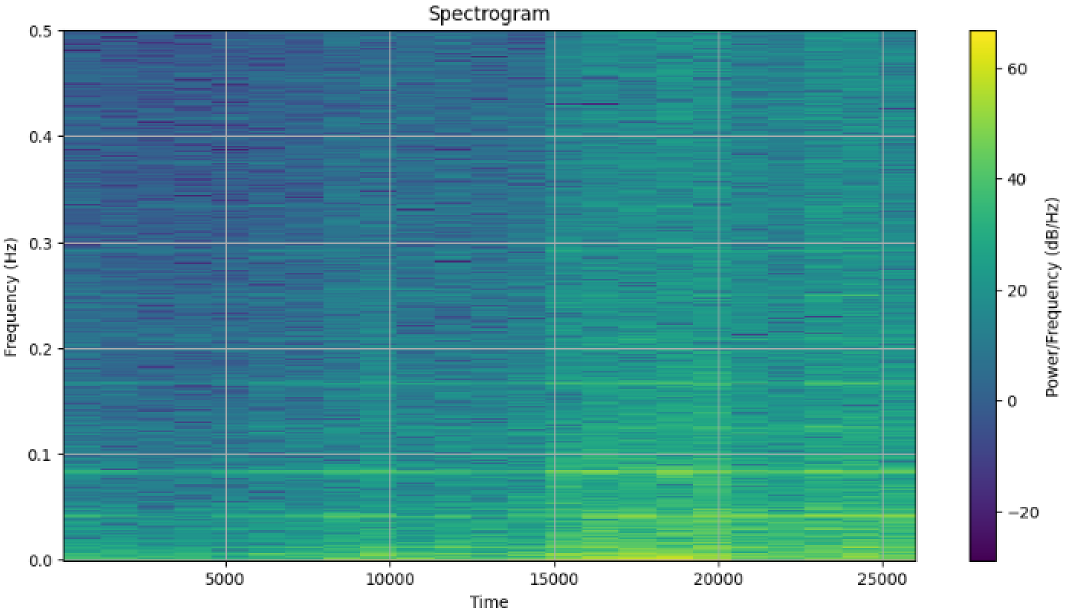


Figure 8-8 Frequency over time

The findings reveal some straightforward insights. The presence of lightered colored bands in the spectrogram corresponds to very low frequencies, indicating that significant fluctuations in this time series are infrequent. On the other hand, the appearance of darker blue bands at relatively higher frequencies suggests that these fluctuations are more frequent but possess less significant impact. In essence, while the underlying causes for these more frequent fluctuations may surface more often, their influence on the overall picture remains relatively modest.

9 Forecasting Models

9.1 Autoregression Algorithms

9.1.1 SARIMA

Autoregression is a time series technique that leverages historical observations as input in a regression equation to predict future values. This straightforward concept has demonstrated its effectiveness in generating accurate forecasts across various time series problems. In a typical regression model, such as linear regression, the output value is modeled as a linear combination of independent input variables, represented as:

$$Y = C0 + (C1 \cdot X1) + \cdots + (Cn \cdot Xn)(1)$$

In this equation, Y signifies the target variable, $C0$ and Cn denote coefficients determined through model fitting on training data, and ($X1$, Xn) represent input variables. This same concept can be applied to time series data, where previous observations of the target variable, referred to as lag variables, are utilized as input variables:

$$Y(t,p) = e + b0 + b1 \cdot X(t-1) + \cdots + bp \cdot X(t-p)(2)$$

Here, $Y(t,p)$ denotes the target variable at time t, e represents white noise, $b0$, $b1$, …., bp are coefficients obtained through model training, and $X(t-1)$, $X(t-p)$ denote the lag variables. This regression model, owing to its utilization of data from the same input variable at previous time steps, is termed an autoregression.

Furthermore, the discrepancies between forecasted and actual/expected values, known as residual errors in a time series dataset, constitute another source of valuable information. These residual errors form their own time series, which may possess temporal patterns. A basic autoregression model can be employed to predict forecast errors, which can subsequently be used for forecasting corrections. This modeling of residual errors is also considered an autoregression and is expressed as follows:

$$Y'(t,q) = e1 + z0 + z1 \cdot e(t-1) + \cdots + zq \cdot e(t-q)(3)$$

Here, $Y'(t,q)$ signifies the target variable at time t, $e1$ represents white noise, $z0$, $z1$,….., zq are coefficients determined through model fitting, and $e(t-1)$, $e(t-q)$ are lag values of the residual errors.

Combining these concepts leads us to the Autoregressive Moving Average (ARMA) model, which is a statistical approach for analyzing and forecasting time series data. It can be expressed as:

$$ARMA(p,q) = Y(t,p) + Y'(t,q)(2)\ and\ (3)$$

Here, p,q denote the number of lags for $Y(t,p)$ and $Y'(t,q)$, respectively. Notably, the ARMA model is applicable only to stationary time series data. In cases where the time series is not stationary, the Autoregressive Integrated Moving Average (ARIMA) model must be applied. ARIMA, an extension of ARMA, incorporates differencing to render time series data stationary.

- AR: A model that utilizes the relationship between an observation and lagged observations.
- I: The application of differencing to raw observations over time, involving the subtraction of a current observation from a prior one to achieve stationarity.
- MA: A model that employs the dependency between an observation and residual errors derived from a moving average model applied to lagged observations.

ARIMA (p, d, q) accepts the same parameters as the ARMA model, along with d, representing the number of differencing operations applied to lag observations. However, ARIMA still does not account for seasonality in the data. To address seasonality, the Seasonal Autoregressive Moving Average (SARIMA) model becomes necessary. SARIMA, an extension of ARIMA, introduces a seasonal component that accounts for and offsets seasonality. SARIMA entails the selection of hyperparameters for both the trend and seasonal aspects of the time series, denoted as SARIMA(p, d, q)(P, D, Q)S.

$$\text{ARIMA} \quad \underbrace{(p,d,q)}_{\substack{\uparrow \\ \text{Non-seasonal part} \\ \text{of the model}}} \quad \underbrace{(P,D,Q)_m}_{\substack{\uparrow \\ \text{Seasonal part of} \\ \text{of the model}}}$$

Trend elements:

- p: Trend AR order.
- d: Trend difference order.
- q: Trend MA order.

Seasonal elements:

- P: Seasonal AR order.
- D: Seasonal difference order.
- Q: Seasonal MA order.
- S: The number of time steps constituting a single seasonal period.

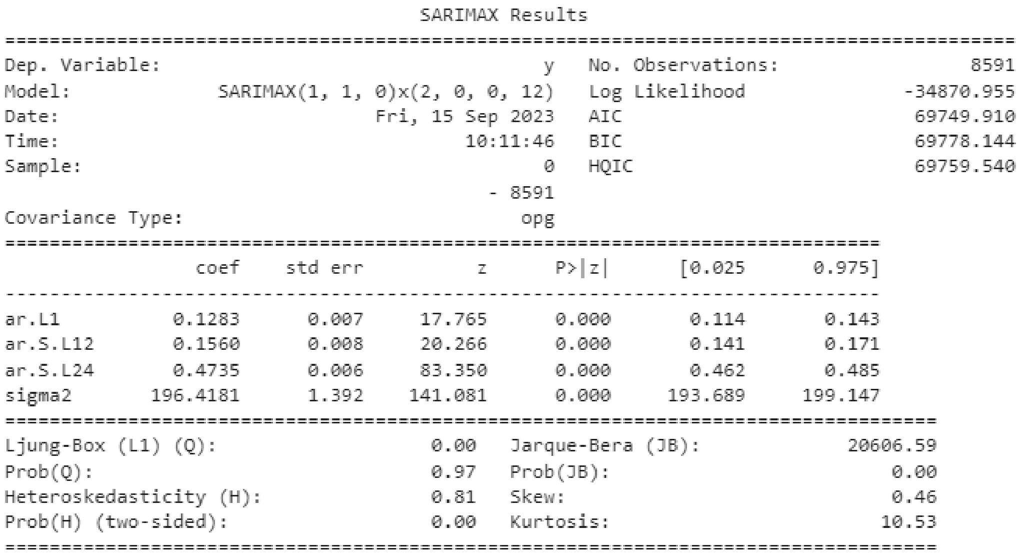
Based on the above, a SARIMA model will be implemented on our data, since our data has a strong seasonal component. Firstly, the parameters of SARIMA model need to be specified. In order to do that, the auto_arima() function from the pmdarima package will be used, so that we can perform a parameter search for the optimal values of the model. Two alternative approaches to determine the ideal SARIMA parameters are through the ACF and PACF plots (shown in the previous chapter), and a grid search.

The Akaike Information Criterion (AIC) plays a pivotal role in the process of model selection within SARIMA modeling. AIC, being a statistical metric, skillfully strikes an equilibrium between the model's goodness of fit and its inherent complexity, with the primary aim of identifying the model that optimally represents the inherent data patterns while meticulously avoiding overfitting. In the realm of SARIMA, where the task entails judiciously determining the values for pivotal parameters such as autoregressive order (p), differencing order (d), and moving average order (q), the AIC assumes the role of a guiding principle. A lower AIC value is indicative of a superior trade-off between model fitness and simplicity. Consequently, during the comparative evaluation of distinct SARIMA models, practitioners often favor the model associated with the lowest AIC value, as it embodies the highest degree of parsimony while effectively encapsulating the underlying intricacies of the time series dynamics. The AIC endorses an objective and quantifiable approach to the assessment of model quality, thereby significantly augmenting the reliability of time series forecasting and analysis. AIC is the main formula lying behind auto_arima function.

$$AIC = -2 * \log(L) + 2 * k$$

The initial component, $-2 * log(L),$ assesses how well the model fits the data, measuring the quality of fit. It is calculated by taking the negative of twice the natural logarithm of the likelihood function, which indicates how effectively the model accounts for the observed data. Smaller values of this component signify a stronger fit between the model and the data. On the other hand, the second component, $2 * k$, acts as a penalty factor to account for the model's complexity. It considers the count of estimated parameters within the SARIMA model. Models with more parameters receive a higher penalty, promoting the preference for simpler models that are still capable of adequately representing the data patterns.

The results of the auto_arima function are presented on the table below:



```
                                     SARIMAX Results
==========================================================================================
Dep. Variable:                                  y   No. Observations:                 8591
Model:             SARIMAX(1, 1, 0)x(2, 0, 0, 12)   Log Likelihood              -34870.955
Date:                            Fri, 15 Sep 2023   AIC                          69749.910
Time:                                    10:11:46   BIC                          69778.144
Sample:                                         0   HQIC                         69759.540
                                           - 8591
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.1283      0.007     17.765      0.000       0.114       0.143
ar.S.L12       0.1560      0.008     20.266      0.000       0.141       0.171
ar.S.L24       0.4735      0.006     83.350      0.000       0.462       0.485
sigma2       196.4181      1.392    141.081      0.000     193.689     199.147
===================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):              20606.59
Prob(Q):                              0.97   Prob(JB):                          0.00
Heteroskedasticity (H):               0.81   Skew:                              0.46
Prob(H) (two-sided):                  0.00   Kurtosis:                         10.53
===================================================================================
```

Figure 9-1 SARIMAX results

Through the SARIMAX results table the value of AIC is indicated, meaning that the model depicted scored the lowest AIC value.

The 2 x 2 diagnostics plot of the model can be shown below:

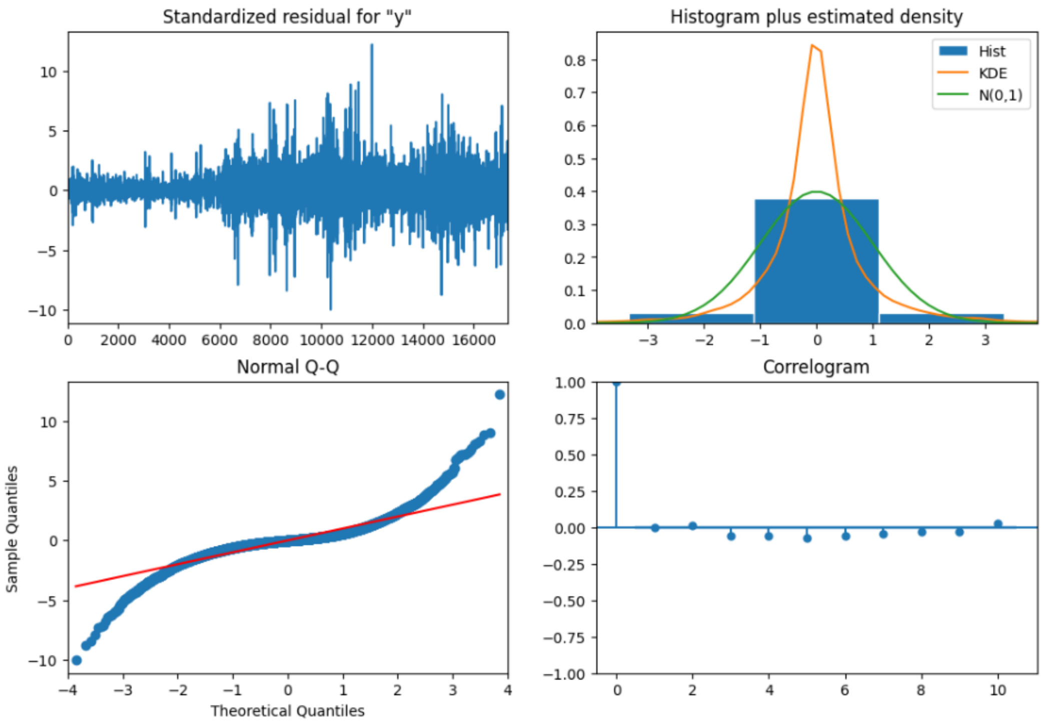


Figure 9-2 2 x 2 diagnostics plot

In the bottom left graph, which depicts the autocorrelation function (ACF) of the residuals, no significant autocorrelation is observed between the residuals, which means that there is no significant information neglected and the residuals are probably white noise.

Therefore the SARIMA model that we will use to train our data and forecast future values is the (1,1,0)x(2,0,0,12). The seasonality was selected based on the second prevailing period discussed in the previous chapter, which was half a day which equals to 12 hours for hourly data like our dataset.

The forecasted values will represent the last week of our dataframe, from 25/12/2022 until 31/12/2022.

The final forecast is illustrated below:

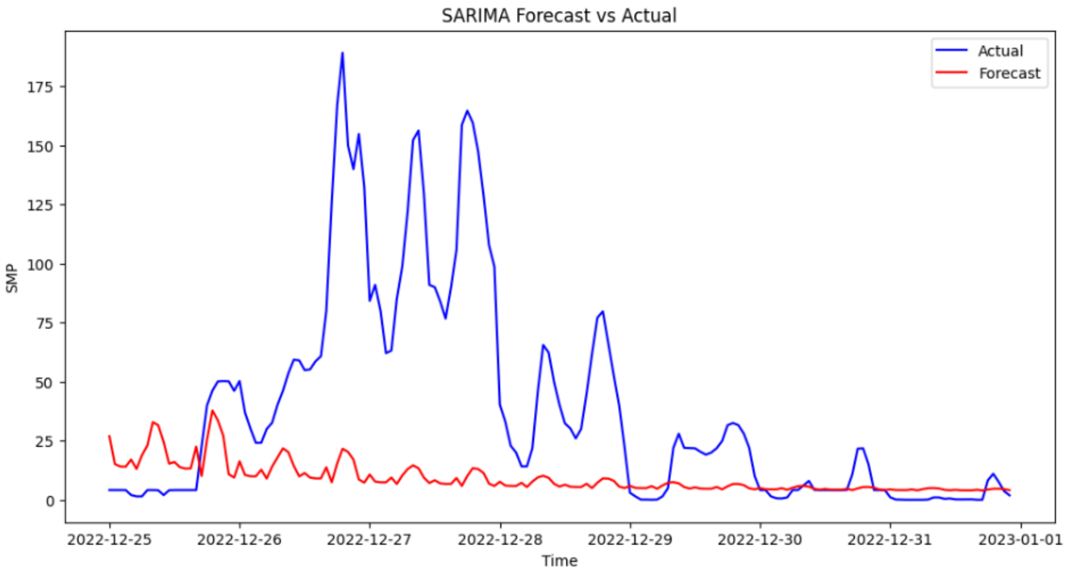


Figure 9-3 SARIMA final forecast

The forecast does not look promising and the fact that the values of the forecast approach zero values may increase the errors which will be presented in the next chapter.

9.1.2 FBProphet

Facebook Prophet, an open-source forecasting tool developed by Facebook's Core Data Science team, it was introduced in 2017 and has rapidly gained popularity as a preferred solution for time series forecasting. Its allure stems from its user-friendly nature, adaptability, and capacity to tackle various forecasting tasks with minimal user intervention.

It is purpose-built to handle time series data, which entails data collected at distinct time intervals. Such data typically comprises historical records of a metric, such as stock prices, website traffic, or daily temperature records. The primary objective of Prophet is to make precise forecasts regarding future values within the time series.

Facebook Prophet offers several essential features for time series forecasting. Firstly, it possesses Automated Seasonality Recognition, which allows Prophet to independently detect and adapt to seasonal patterns in the data, encompassing daily, weekly, and yearly cycles. This is particularly beneficial for datasets marked by repetitive patterns. Additionally, it includes functionality for Accounting for Holidays and Special Occasions, granting users the flexibility to specify relevant holidays and significant events within the time series, enhancing Prophet's ability to handle data irregularities effectively. Furthermore, Prophet excels in Trend Projection, capable of modeling both short-term trends and long-term growth or decline, thus facilitating more robust predictions. Lastly, its Versatility is notable, as it is adept at seamlessly working with datasets that may contain missing data points or outliers, making it highly suitable for real-world data characterized by noise.

The fundamental equation of Facebook Prophet is an additive time series model expressed as:

$$y(t) = g(t) + s(t) + h(t) + \epsilon t$$

Where:

- $y(t)$ signifies the observed value of the time series at a specific time point t.
- $g(t)$ denotes the trend component, which characterizes the long-term growth or decline tendencies inherent in the data.
- $s(t)$ represents the seasonality component, capturing cyclic patterns that recur at regular intervals, such as daily, weekly, or yearly patterns.
- $h(t)$accounts for holiday effects and other exceptional events that may exert an influence on the time series at specific time points.
- ϵt stands for the error term, encompassing any stochastic or random variations within the data.

The Prophet function can work without any given arguments. Therefore, we tried to forecast the last week of our SMP values, and the produced result can be found below.

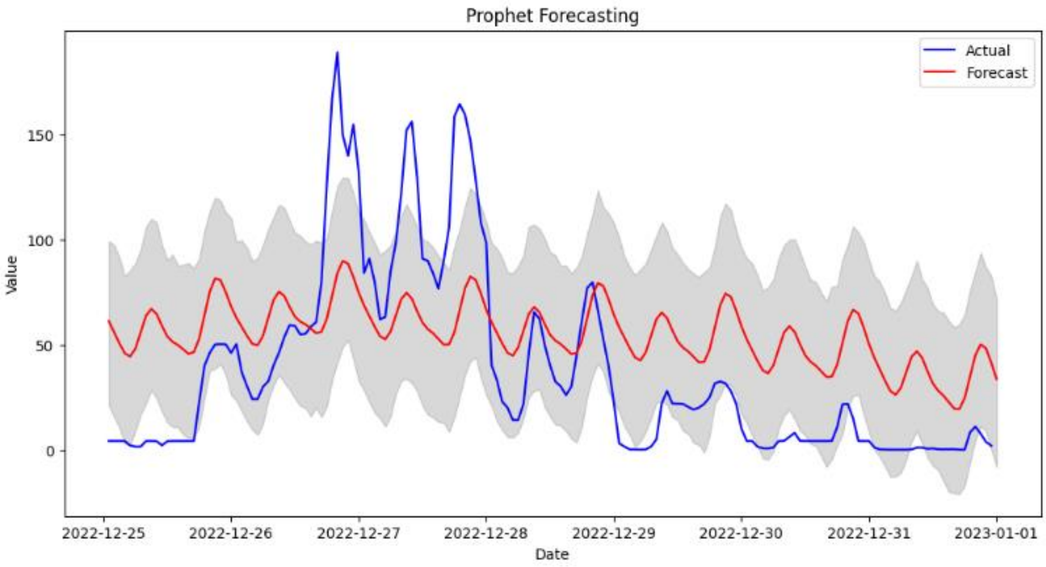


Figure 9-4 Prophet forecasting

The result looks promising provided the fact that there were not exogenous variables involved in the forecast. Note that, the grey area dictates the upper and bottom limits of the model's predictions.

The error values, which will ultimately decide the model's performance, in contrast with the other model's used, will be presented in the next chapter.

Also, an overview of the forecast along with the test and part of the training set is presented:

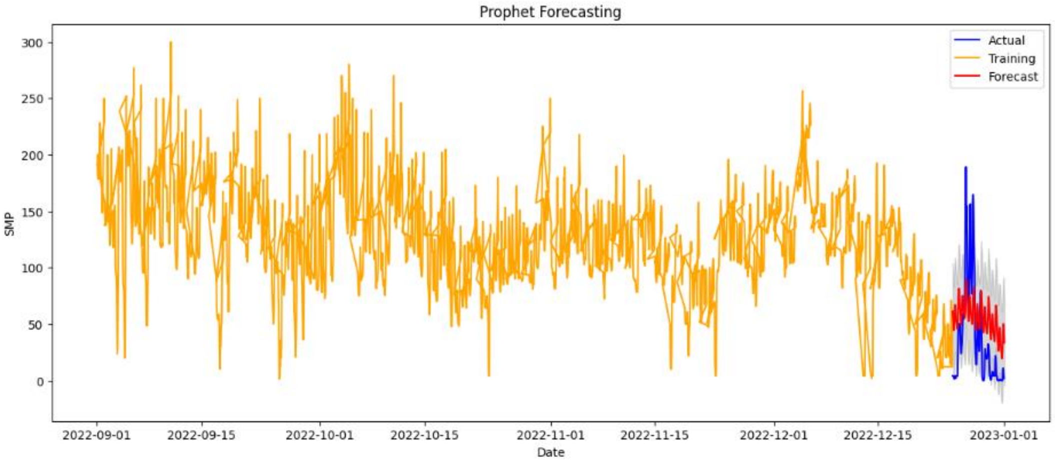


Figure 9-5 Overview of the Prophet forecast

9.2 Gradient Boosting

Gradient boosting stands out as one of the most widely adopted machine learning algorithms, particularly for tabular datasets. Its effectiveness lies in its ability to uncover intricate nonlinear relationships between the target variable and features within a model. Moreover, it offers exceptional versatility, enabling the handling of missing data, outliers, and high cardinality categorical features without requiring special preprocessing. While you can construct basic gradient boosting trees with the help of popular libraries like XGBoost or LightGBM without delving into the algorithm's intricacies, gaining insight into its inner workings becomes essential when you embark on tasks such as hyperparameter tuning and customizing loss functions. This understanding plays a crucial role in enhancing the overall quality of your model.

Light GBM is a gradient boosting framework that employs a tree-based learning algorithm. Its distinction from other tree-based algorithms lies in the way it constructs trees. Unlike other algorithms that build trees horizontally, Light GBM grows trees vertically. In other words, LGBM grows trees leaf by leaf, whereas alternative algorithms grow them level by level. LGBM selects the leaf with the maximum reduction in loss for growth. This leaf-wise approach can result in greater loss reduction compared to the level-wise approach when growing the same leaf. The diagrams below provide a visual representation of how LightGBM differs from other boosting algorithms in terms of implementation.

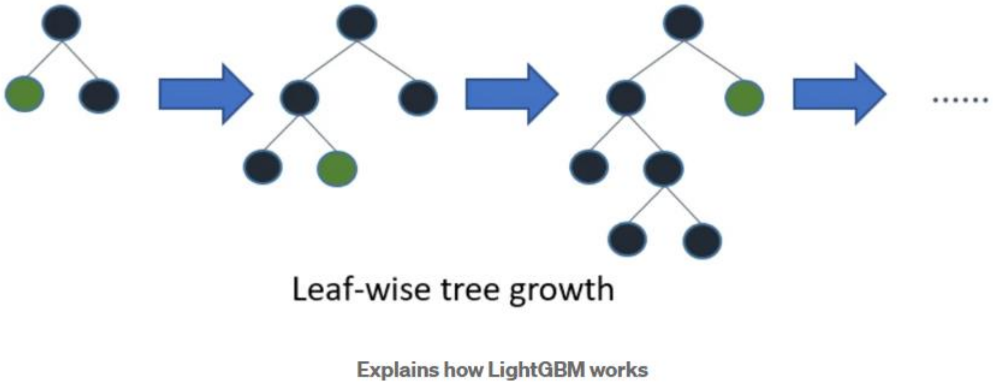


Figure 9-6 Leaf – wise tree growth

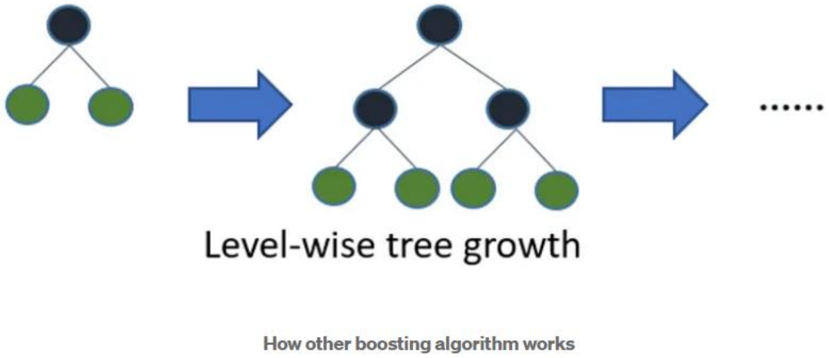


Figure 9-7 Level – wise tree growth

The data volume is constantly on the rise, posing challenges for traditional data science algorithms in terms of speed and efficiency. Light GBM earns its 'Light' designation due to its remarkable speed, making it adept at handling large datasets while consuming minimal memory resources. Its popularity also stems from its emphasis on result accuracy and support for GPU learning, making it a favored tool in data science applications. However, it's important to note that Light GBM may not be suitable for all scenarios. It's not advisable to use LGBM with small datasets, as it is prone to overfitting. While there's no strict row count threshold, practical experience suggests its optimal use for datasets exceeding 10,000 rows.

In order to construct and utilize the LGBM model Darts library was used. Darts, short for "Data Analysis and Regression Tools with Smoothing," stands as a potent open-source library primarily crafted with the purpose of enhancing time series forecasting and analysis. Its inception aimed to furnish a robust arsenal of tools and models, facilitating data scientists, researchers, and analysts in effectively addressing a diverse array of challenges within the realm of time series forecasting. Darts has been implemented in Python and is constructed atop well-established libraries like NumPy, pandas, and scikit-learn, rendering it seamlessly integrable into prevailing data science workflows.

After converting our dataframe into a time series and using MissingValuesFiller() function to fill the NaN values the LightGBM Model on Darts has the following parameters:

***class* darts.models.forecasting.lgbm.LightGBMModel(*lags=None*, *lags_past_covariates=None*, *lags_future_covariates=None*, *output_chunk_length=1*, *add_encoders=None*, *likelihood=None*, *quantiles=None*, *random_state=None*, *multi_models=True*, *use_static_covariates=True*, *categorical_past_covariates=None*, *categorical_future_covariates=None*, *categorical_static_covariates=None*, ***kwargs*)**

For the results presented below, only the lags argument was given a value. The value chosen, was based on the most prevailing frequency of the dataset ~ 346. The produced result is shown below:

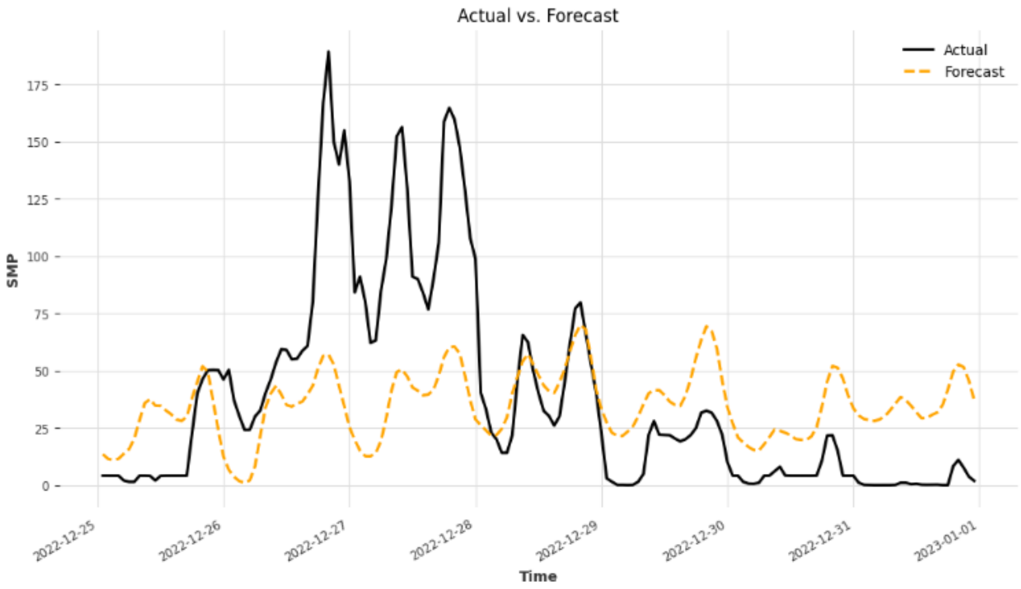


Figure 9-8 Actual vs Forecast

The results dictate that LGBM might be the most accurate forecasting model of all the ones in test. Errors metrics, which will illustrate the result will be discussed in the next chapter.

Moreover, as shown in the previous cases an overview of the training, test and forecast set:

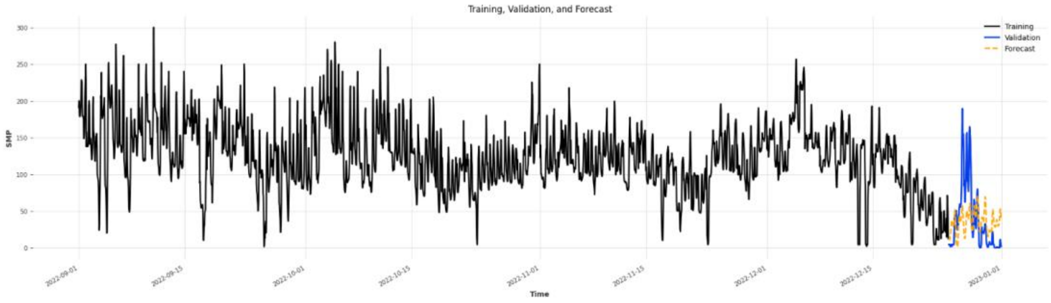


Figure 9-9 Training, validation and forecast

10 Results

Three categories of machine learning models were developed, assessed, and analyzed. These included two autoregressive models and a gradient boosting model. To conduct a comprehensive evaluation of forecast performance, a training dataset encompassing records from January 1, 2020, to December 24, 2022, was utilized. Additionally, a test dataset spanning from December 25 ,2022, to December 31, 2022, was employed to generate Day-Ahead forecasts extending up to one week into the future. The assessment of forecast accuracy was conducted by computing the respective Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE) values. Both metrics were calculated by employing the respective functions, *mean_squared_error*, *mean_absolute_error* from the sklearn.metrics library.

Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) are widely employed metrics for the assessment of predictive model performance, particularly in the realms of regression analysis and forecasting. These metrics serve the purpose of gauging how accurately a model's predictions align with actual observed values. Let us delve into the theoretical underpinnings and mathematical expressions for both RMSE and MAPE:

Root Mean Square Error (RMSE):

RMSE serves as a means to quantify the average magnitude of discrepancies between predicted values (ŷ) and their corresponding actual values (y). It emerges as a frequently utilized metric, especially in regression scenarios, and it places greater emphasis on larger discrepancies compared to smaller ones.

The RMSE formula is articulated as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum(y_i - \hat{y}_i)2}$$

In this equation:

- n represents the total count of observations.
- y_i stands for the actual (observed) value for observation i.
- $\hat{y}_i$ signifies the predicted value for observation i.

Mean Absolute Error (MAE):

MAE is a fundamental metric used for evaluating predictive models, especially in the context of regression analysis. It quantifies the average magnitude of errors between predicted values (ŷ) and their corresponding actual values (y). Unlike RMSE, MAE treats all errors equally without giving preference to larger discrepancies.

The formula for MAE is defined as follows:

$$MAE = \frac{1}{n}\sum | y_{i-\hat{y}_i} |$$

In this equation:

- n represents the total count of observations.
- y_i stands for the actual (observed) value for observation i.
- $\hat{y}_i$ signifies the predicted value for observation i.

MAE provides a straightforward measure of the average absolute error between predicted and actual values. It is robust to outliers and offers an easily interpretable assessment of a model's accuracy.

When considering MAE alongside RMSE and MAPE, it is essential to recognize that while RMSE emphasizes larger errors and MAPE expresses errors as percentages, MAE treats all errors equally and provides a clear and balanced evaluation of predictive model performance. The choice of metric depends on the specific objectives and characteristics of the problem being analyzed.

The results table is illustrated below:

7 DAYS FORECAST	*MAE (EUR/MWH)*	*RMSE (EUR/MWH)*
SARIMA	*32.98*	*51.24*
PROPHET	*34.06*	*38.75*
LGBM	*29.97*	*39.70*

Table 1 Results

It becomes clear by the above array that there is not a clear answer to the most effective model among the three.

When it comes to the RMSE metric, Fb Prophet algorithm strikes the best result and as described above, RMSE is sensitive to outliers and gives more weight to the larger errors, so FB Prophet’s forecasts does not produce large deviations from the actual values. Furthermore, it can be deduced that the FB Prophet is a good fit for the data as it captures underlying patterns and trends effectively. Low RMSE also means that, the model’s performance is consistent across the dataset as it does not exhibit significant bias or variability in the predictions.

As for the MAE metric is concerned, LGBM algorithm strikes the lowest rate. A low MAE rate suggests robustness in the model, as it is less sensitive to outliers compared to RMSE and we can conclude that the model is not affected easily by extreme values in the dataset, which is important especially in our dataset as couple of months before the forecast the Russian-Ukrainian war had skyrocketed the SMP values. Also, MAE provides a straightforward and interpretable measure of forecasting error. The average absolute difference between predicted and actual values is easy to understand. Lastly, low MAE values indicate that the model's forecasts are reliable and can be used for decision-making. This is especially important in applications where accurate forecasting is crucial.

Overall, considering both metrics as well as the graphs presented in the previous chapter, it is safe to come to the conclusion that the LGBM algorithm was the best fit for the dataset used and provided the most accurate results.

11 Conclusion and future work

This MSc thesis engaged with the creation and comparison of different machine learning models, in order to accurately forecast the Spanish Day- Ahead System Marginal Price. Two main categories were investigated, autoregression and gradient boosting models. Specifically, three approaches were tested, a Sarima, a FBPhropet and a LGBM model. Even though, a small number of studies have been carried out regarding the forecast of the Spanish SMP, it becomes clear that the produced results, even though promising, could improve. System Marginal Price depends on a lot of factors. The main ones are the expenses related to the fuels employed in producing electricity, such as natural gas, coal, and oil, have a direct influence on the operational expenditures of electricity-generating facilities. Changes in fuel costs can notably affect the prices of energy produced, particularly for power plants running on natural gas. Moreover, the production of renewable energy, including wind and solar power, can influence pricing within the energy system. Abundant renewable energy output during sunny or windy periods can result in reduced electricity prices due to their minimal or non-existent production expenses. Weather conditions also have a pivotal impact on both the demand and supply of electricity. Extreme weather occurrences, such as heatwaves or cold spells, can spur greater demand for heating or cooling, which can in turn influence pricing. Severe weather events also have the potential to disrupt the infrastructure related to energy generation and transmission. Lastly, electricity imports and exports have the capacity to influence prices by harmonizing supply and demand across broader geographic regions. The models used in this thesis have the ability to include exogenous parameters in the forecast (e.g. SARIMAX) and thus a first though could be to integrate one or more of the variables mentioned above to the forecasting procedure. However, very few data about the aforementioned variables could be found for the discussed period (1/1/2020 – 31/1/2022) and therefore it was considered not using any of them as they were inadequate. Another suggestion could be the creation of an ensemble model, combining ARIMA, Facebook Prophet, and LightGBM to leverage their respective strengths and improve overall forecasting accuracy. Ensemble methods such as stacking and weighted averaging could be also put to test so that the metrics of the tested models would improve and therefore the forecast would be more accurate. Lastly, it is worth mentioning that our dataset contained a variation that could not be quantified, yet it affected the SMP prices and the forecast, the Russian-Ukrainian war. An unfortunate situation that skyrocketed the energy prices all over the European continent and disorientated our tested models. It can be clearly seen, that none of the models implemented can predict the prices that at that time were surging due to the war.

12 Bibliography

Rob J Hyndman, R., Athanasopoulos, G., 2018, Forecasting: Principles and Practice, 2nd ed., Monash University, Australia

Bontempi, G., Taieb, S., Le Borgne, Y., 2013, Machine Learning Strategies for Time Series Forecasting, DOI: 10.1007/978-3-642-36318-4_3

García-Martos, C., Rodríguez, J., Sánchez, M., 2007, Mixed Models for Short-Run Forecasting of Electricity Prices: Application for the Spanish Market, pp 544 - 552, DOI: 10.1109/TPWRS.2007.894857

IEA, 2023, Electricity Market Report 2023

Pino, R., et al, 1999, Short term forecasting of the electricity market of Spain using neural networks

Popovska, E., Georgieva-Tsaneva, G., 2022, ARIMA Model for Day-Ahead Electricity Market Price Forecasting, pp 149-161, DOI: 10.55630/STEM.2022.0418

Romero, A. et al., 2018, Day-Ahead Price Forecasting for the Spanish Electricity Market, DOI: 10.9781/ijimai.2018.04.008

Wang, D., et al., 2022, Electricity Price Instability over Time: Time Series Analysis and Forecasting, https://doi.org/10.3390/su14159081

https://facebook.github.io/prophet/

https://towardsdatascience.com/machine-learning-part-19-time-series-and-autoregressive-integrated-moving-average-model-arima-c1005347b0d7

https://www.endesa.com/en/blogs/endesa-s-blog/light/why-electricity-expensive-in-spain

https://www.omie.es/en/mercado-de-electricidad

https://www.researchgate.net/figure/Determination-of-System-Marginal-Price-SMP-where-the-aggregate-Supply-and-Demand_fig1_340013197

https://www.spglobal.com/commodityinsights/en

https://www.statista.com/statistics/1267552/spain-monthly-wholesale-electricity-price/

A. APPENDIX 1 – PYTHON CODES

Augmented Dickey-Fuller Test

```
result = adfuller(df['Value'])
print('ADF Statistic:', result[0])
print('p-value:', result[1])
if result[1] <= 0.05:
    print("Reject the null hypothesis - Data is stationary")
else:
    print("Fail to reject the null hypothesis - Data is non-stationary")
```

Additive Decomposition

```
decomposition = sm.tsa.seasonal_decompose(time_series, freq=frequency, model="additive")
trend = decomposition.trend
seasonal = decomposition.seasonal
residual = decomposition.resid
return trend, seasonal, residual
trend, seasonal, residual = simple_additive_decomposition(df[‘y’], frequency=12)
```

Seasonal Decomposition of Time Series (STL)

```
stl_decomposition = sm.tsa.STL(df['y'], seasonal=12)
result = stl_decomposition.fit()
trend = result.trend
seasonal = result.seasonal
residual = result.resid
```

Autocorrelation and Partial Autocorrelation

```
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
first_100_data_points = df['y'][:100]
acf_result = sm.tsa.acf(first_100_data_points['y'], nlags=100, fft=False)
plt.figure(figsize=(16, 6))
plot_acf(first_100_data_points, lags=99)
plt.title("Autocorrelation Plot of First 100 Data Points")
plt.xlabel("Lag")
plt.ylabel("Autocorrelation")
plt.grid(True)
plt.show()
plt.figure(figsize=(16,6))
plot_pacf(first_100_data_points, lags=49)
plt.title("Partial Autocorrelation Plot of First 100 Data Points")
plt.xlabel("Lag")
plt.ylabel("Partial Autocorrelation")
plt.grid(True)
plt.show()
```

Fourier Transform & Spectrogram

```
fft_result = np.fft.fft(df['y'])
N = len(df['y'])
Fs = 1.0
frequencies = np.fft.fftfreq(N, 1.0 / Fs)
amplitude = np.abs(fft_result)
positive_freq_indices = np.where(frequencies >= 0)

from scipy.signal import spectrogram
window_size = 1256
overlap = 128
requencies, times, Sxx = spectrogram(df['y'], fs=Fs, nperseg=window_size, noverlap=overlap)
plt.figure(figsize=(12, 6))
plt.pcolormesh(times, frequencies, 10 * np.log10(Sxx))
plt.colorbar(label='Power/Frequency (dB/Hz)')
plt.xlabel('Time')
plt.ylabel('Frequency (Hz)')
plt.title('Spectrogram')
plt.grid(True)
plt.show()
```

Sarima

```
from statsmodels.tsa.statespace.sarimax import SARIMAX
from pmdarima import auto_arima
from sklearn.metrics import mean_squared_error, mean_absolute_error
from math import sqrt

model = auto_arima(train_data['y'], seasonal=True, m=12, stepwise=True, trace=True)
```

```
print(model.summary())
print(model.order)
sarima_model = SARIMAX(train_data['y'], order=(1, 1, 0), seasonal_order=(2, 0, 0, 12))
sarima_fit = sarima_model.fit()
forecast_period = 168
forecast = sarima_fit.forecast(steps=forecast_period)
rmse = sqrt(mean_squared_error(test_data['y'], forecast))
mae = mean_absolute_error(test_data['y'], forecast)
print(f"RMSE: {rmse}")
print(f"MAE: {mae}")
sarima_fit.plot_diagnostics(figsize=(12, 8))
plt.show()
```

FBProphet

```
df['ds'] = pd.to_datetime(df['ds'])
df = df.set_index('ds')

split_date = pd.to_datetime('2022-12-25')
train_data = data[data['ds'] < split_date]
test_data = data[data['ds'] >= split_date]
train_data.head()

model = Prophet()
model.fit(train_data)

future = pd.DataFrame({'ds': pd.date_range(start=test_data['ds'].min(), periods=7*24, freq='H')})
forecast = model.predict(future)
forecasted_data = forecast[['ds', 'yhat']]
forecasted_data.head()
actual_data = test_data[['ds', 'y']]
actual= actual_data.sort_values(by='ds', ascending=True) rmse = np.sqrt(mean_squared_error(test_data['y'], forecasted_data.iloc[-len(test_data):]['yhat']))
mae = mean_absolute_error(test_data['y'], forecasted_data.iloc[-len(test_data):]['yhat'])
```

LGBM

```
pip install u8darts[all] lightgbm

import pandas as pd
import numpy as np
from darts import TimeSeries
from darts.models import LightGBMModel
from darts.dataprocessing.transformers import Scaler , MissingValuesFiller
from darts.utils.timeseries_generation import datetime_attribute_timeseries
```

```
data = TimeSeries.from_dataframe(data, time_col='ds',
value_cols='y',fill_missing_dates=True, freq='H')


from darts.models import LightGBMModel
model = LightGBMModel(lags =395, lags_past_covariates=None,
lags_future_covariates=None)


mae = mean_absolute_error(test.pd_series(), forecast.pd_series())
rmse = sqrt(mean_squared_error(test.pd_series(), forecast.pd_series())
```