**OPTIMIZING ENERGY GENERATION IN RENEWABLE ENERGY SYSTEMS THROUGH DATA-DRIVEN SIMULATION**

TABLE OF CONTENTS

[Data Analysis and Result 3](#_Toc134140999)

[Data analysis 3](#_Toc134141000)

[Result 7](#_Toc134141001)

[Conclusion 11](#_Toc134141002)

[References 13](#_Toc134141003)

**Data Analysis and Result**

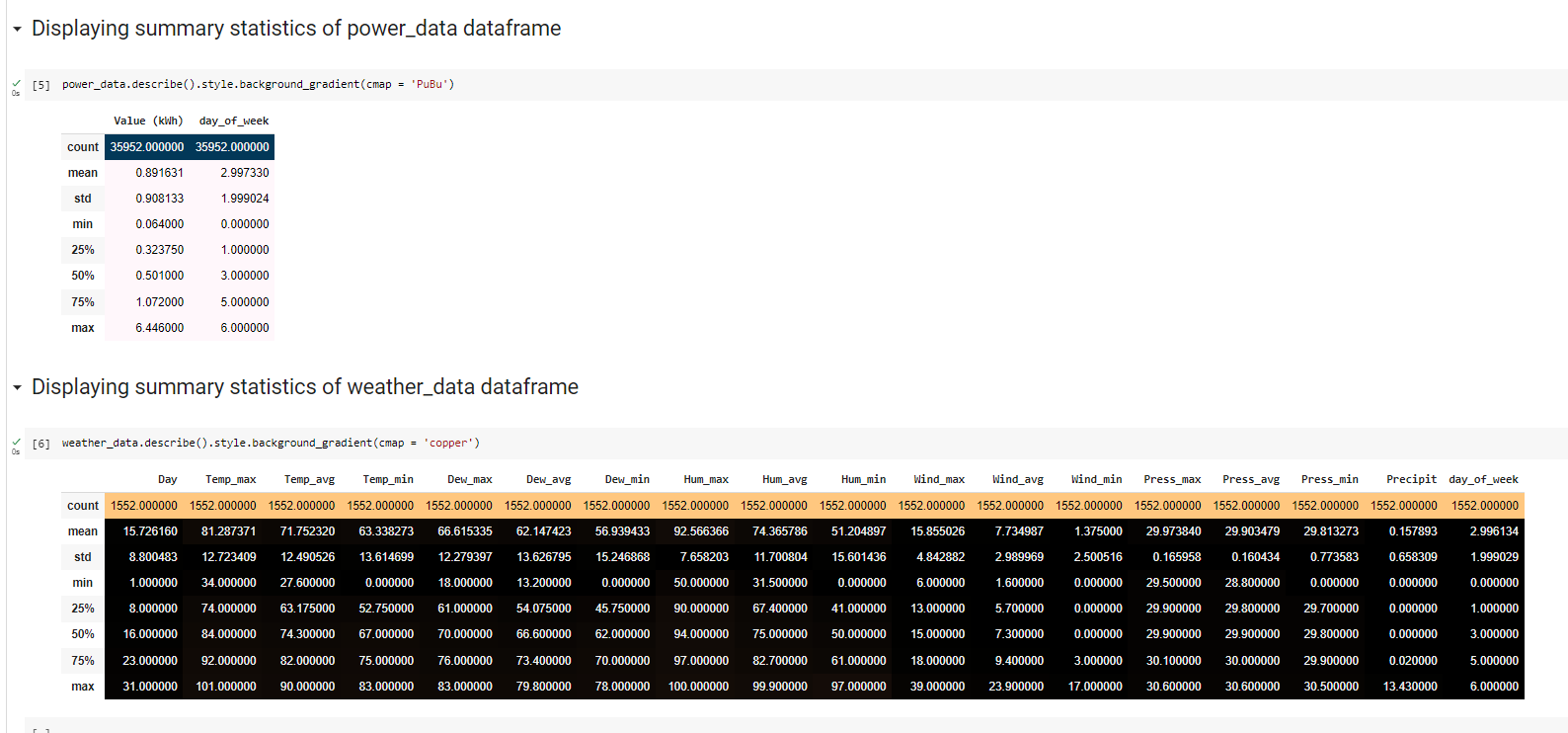
**Data analysis**



**Figure 1: Importing necessary libraries**

(Source: Obtained from Google Collab)

The Python script presented imports essential libraries and two datasets, namely 'power\_usage\_2016\_to\_2020.csv' and 'weather\_2016\_2020\_daily.csv', which contain data pertaining to power consumption and weather conditions, respectively.

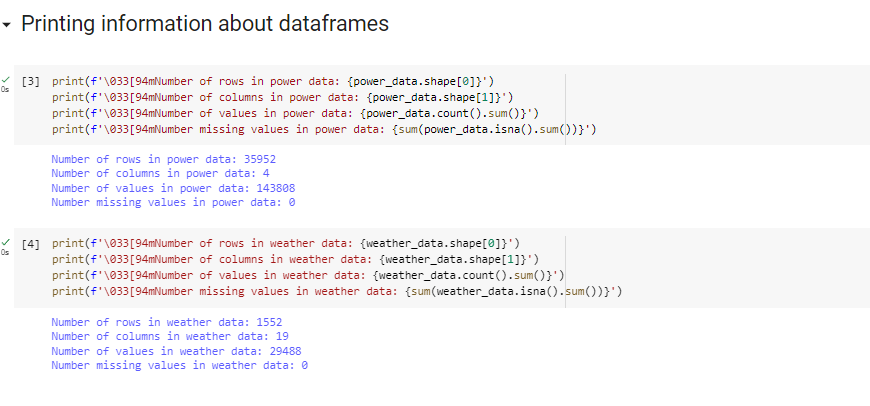


**Figure 2: Displaying summary statistics of power\_data data frame**

(Source: Obtained from Google Collab)

The programme performs calculations of descriptive statistics on a set of data, wherein the data pertains to two distinct columns that are identified as "Value (kWh)" and "day\_of\_week". The method is known as ".describe()" and provides a summary of statistical measures, encompassing the count, mean, standard deviation, minimum and maximum values, as well as percentiles. Based on the tabulated data, it can be observed that there is a combined total of 35,952 occurrences for all of the variables mentioned above. The variable "Value (kWh)" has an arithmetic mean of 0.891631, indicating an approximate typical energy usage level of 0.89 kilowatt-hours. The computed standard deviation of the variable "Value (kWh)" is 0.908133, indicating a significant level of variability in the data pertaining to energy consumption. The energy consumption exhibits a broad spectrum, as indicated by the minimum and maximum values of "Value (kWh)" which are 0.064 and 6.446, correspondingly. The variable "day\_of\_week" displays a numeric range of values spanning from 0 to 6. The calculated mean value of 2.997330 implies that the dataset includes observations for all seven days of the week.

The descriptive summary of the weather data provides statistical data on a number of factors, including temperature, humidity, wind, pressure, precipitation, and the day of the week. There are 1552 total observations in the dataset. The summary shows that the average maximum temperature (Temp\_max), average temperature (Temp\_avg), and average lowest temperature (Temp\_min) are all 81.29°F, 71.75°F, and 63.34°F, respectively. The observed temperature variables have a large standard deviation, which indicates that there is a significant amount of temperature variation within the dataset. Maximum humidity (Hum\_max) is often 92.57%, while the lowest humidity (Hum\_min) is typically 51.20%. The average amount of precipitation is 0.16 inches. The summary report includes statistical data for other factors including dew point, wind, and pressure, such as the mean, standard deviation, minimum, maximum, and quartiles. The summary includes the total number of observations for each variable as well as the typical day of the week for each observation.



**Figure 3: Printing information about data frames**

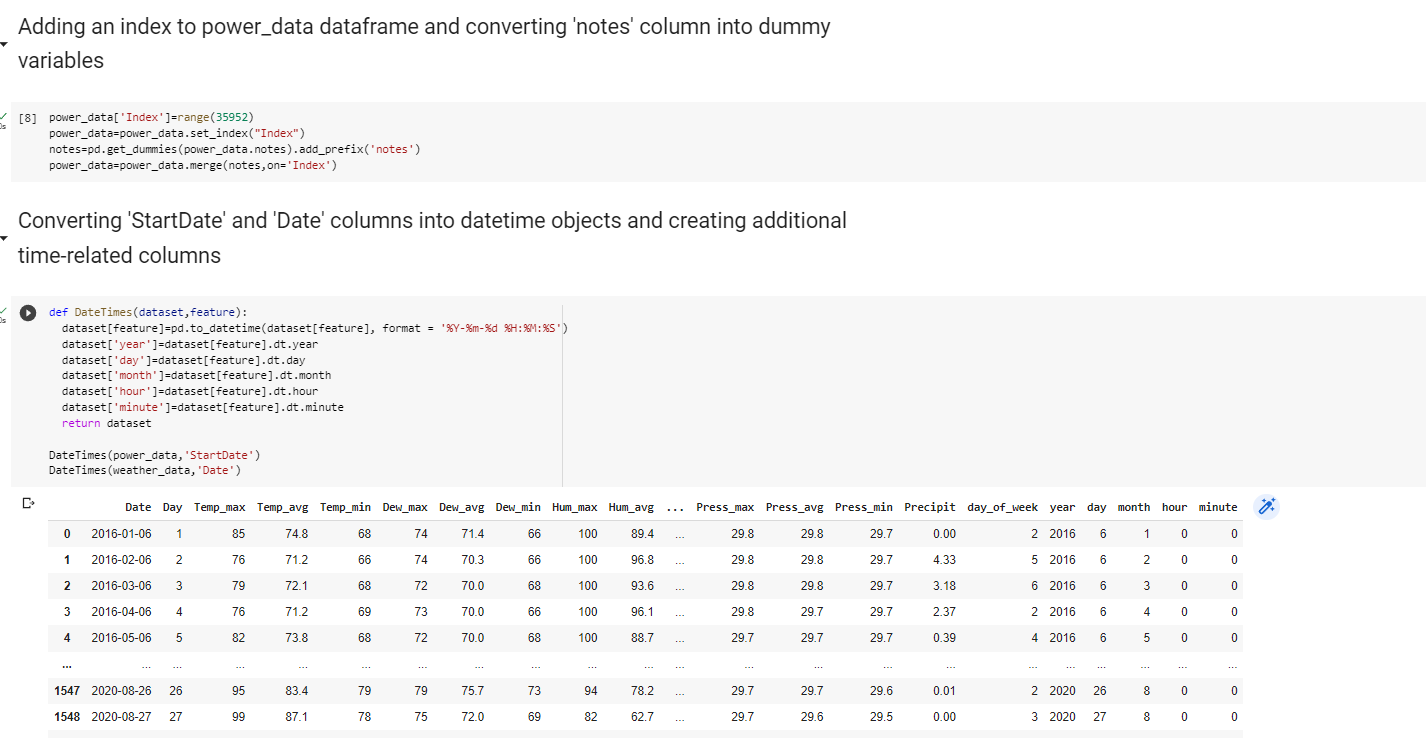
(Source: Obtained from Google Collab)

The first section of the code makes use of the technique known as shape to calculate the number of rows and columns contained inside the power\_data dataset. After that, it uses the count technique to tally the total number of non-missing values in the dataset, and it uses the sum method to add up the total number of missing values in the dataset by utilising the isna method. Both of these methods are described below.

The second section of the code applies the exact same procedures in order to ascertain the number of rows, columns, the total number of non-missing values, and the total number of missing values included inside the weather\_data dataset. The results of these computations are shown in the output. There are a total of 143,808 values and 0 missing values in the power\_data dataset, which has 35,952 rows and 4 columns. There are no missing values. There are 1,552 rows and 19 columns in the weather\_data dataset, for a total of 29,488 values, and there are 0 values that are missing from the dataset.

The code makes use of the escape character '033[94m' to alter the colour of the written output to blue. This makes it simpler to differentiate the output from any other output that may be present in the console.

The script outputs details pertaining to the datasets, including the count of rows, columns, values, and absent values. Subsequently, the describe() function is employed to present summary statistics for both datasets. The summary statistics of the dataset's measures of central tendency, variability, and distribution are crucial in detecting outliers and facilitating informed data cleaning and preprocessing decisions.



**Figure 4: Displaying summary statistics of power data and whether data**

(Source: Obtained from Google Collab)

The index is generated for the dataset pertaining to power usage, and the 'notes' column is transformed into binary variables. The process involves the conversion of the 'StartDate' and 'Date' columns from the power usage and weather datasets, respectively, into datetime objects. Subsequently, the creation of supplementary time-related columns, including year, day, month, hour, and minute, is executed.

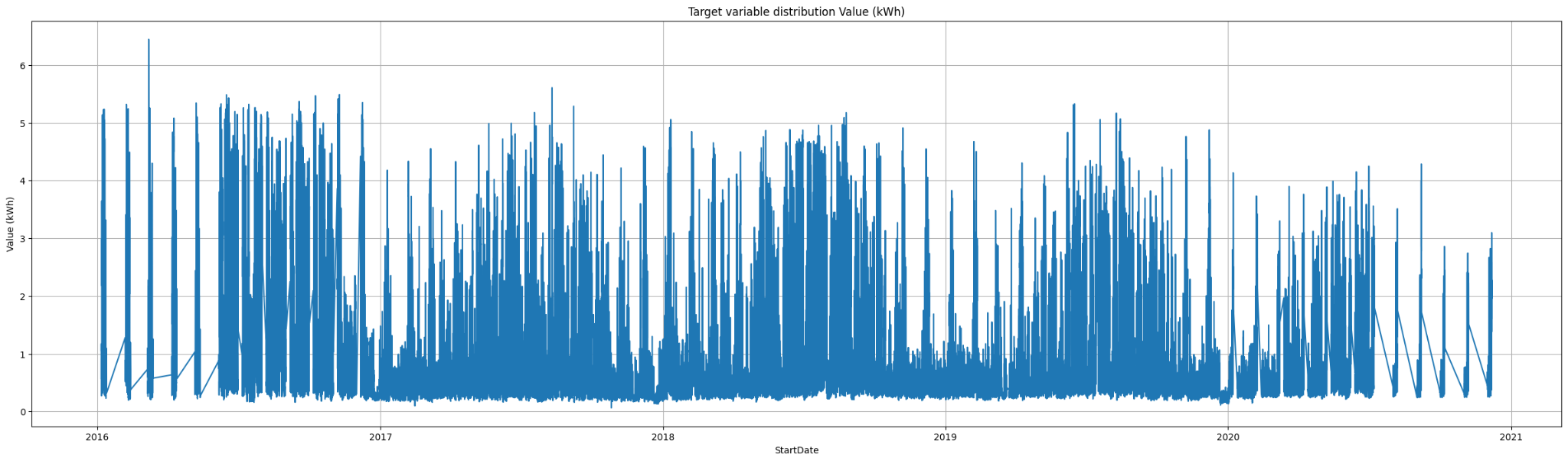


**Figure 5: Function to display missing values in a dataset**

(Source: Obtained from Google Collab)

The script contains a function called missing\_values() that is intended to exhibit any absent values present within a given dataset. The present function receives a dataset as its input and generates a bar plot that visually represents the count and proportion of absent values for every column in the dataset. The functionality offers a convenient approach to detecting columns that contain null values, a crucial task as the presence of such values can potentially impact the precision of statistical evaluations and machine learning algorithms.

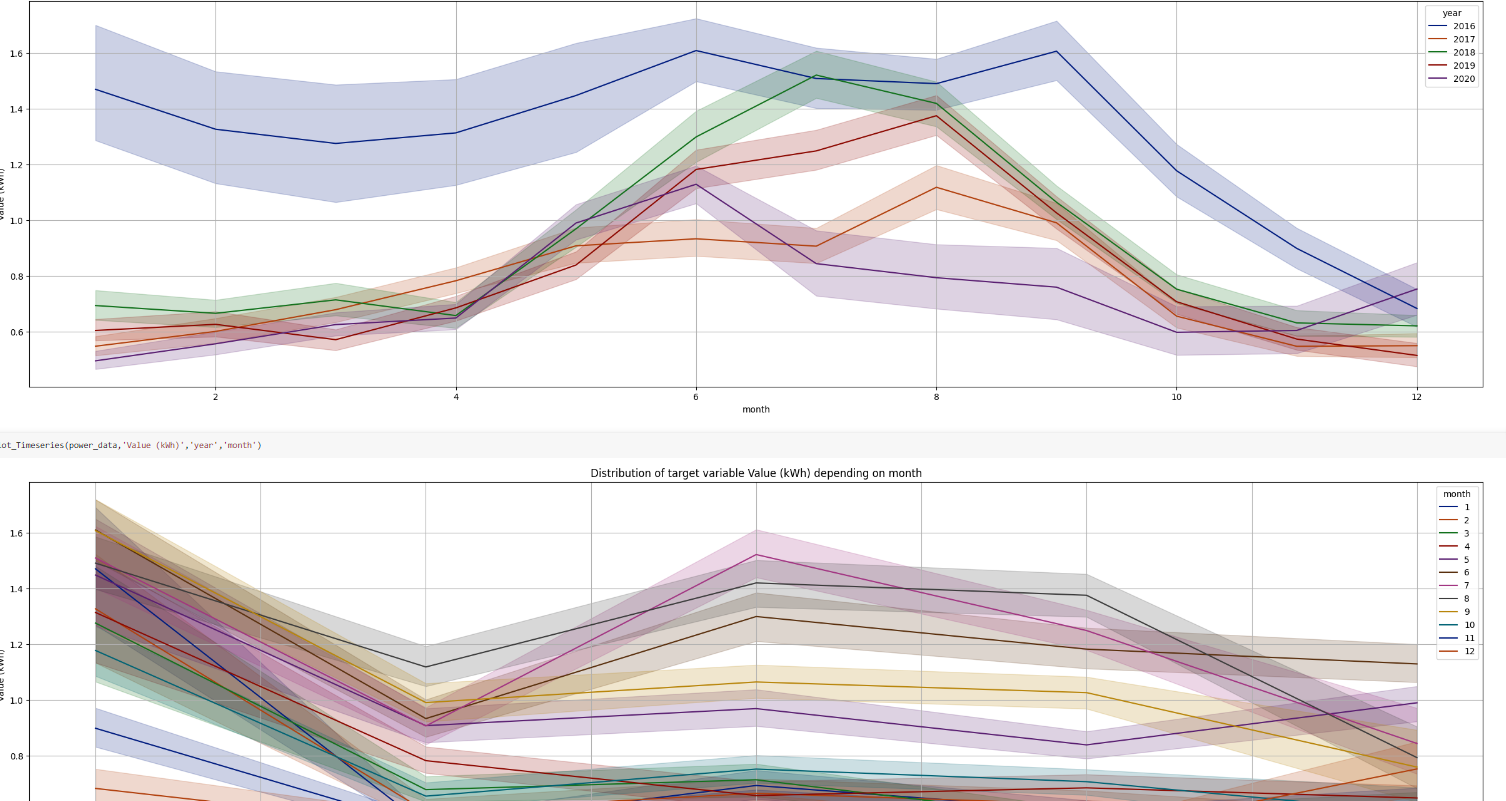
**Result**



**Figure 6: Target variable distribution**

(Source: Obtained from Google Collab)

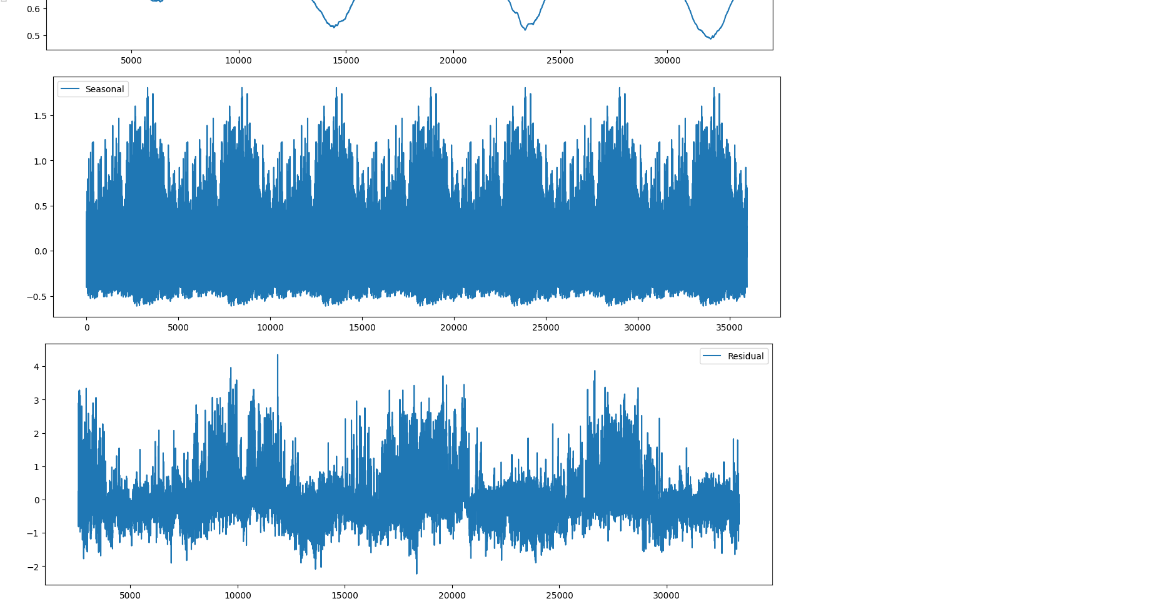
The script incorporates the Plot\_Timeseries() function for the purpose of visualising time series data.



**Figure 7: Distribution of target variable with different values (year, month etc)**

(Source: Obtained from Google Collab)

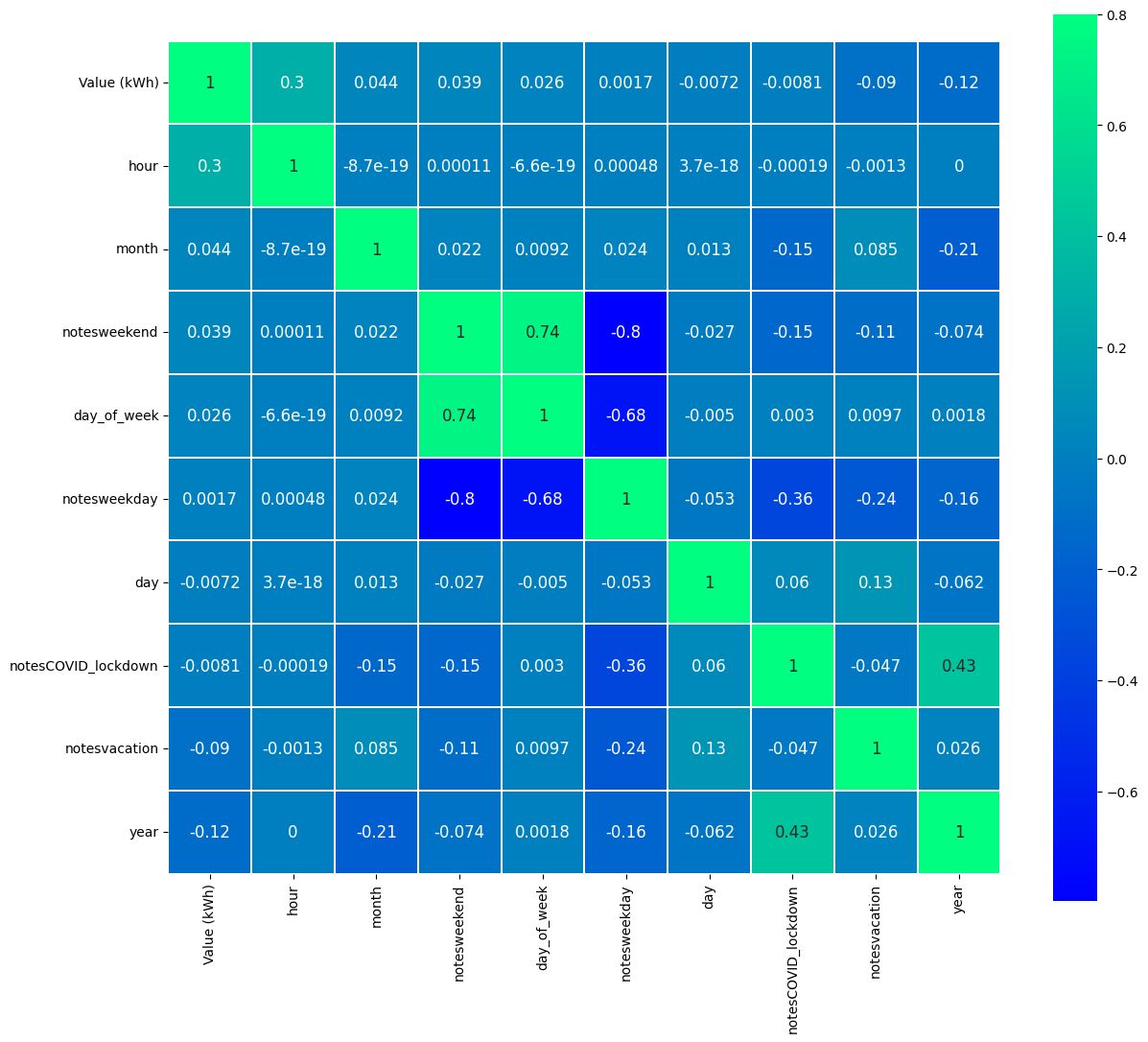
The present function accepts a dataset, a target variable, and a temporal feature as arguments and generates a line plot that visualises the temporal distribution of the target variable.



**Figure 8: Decomposition plot of the time series**

(Source: Obtained from Google Collab)

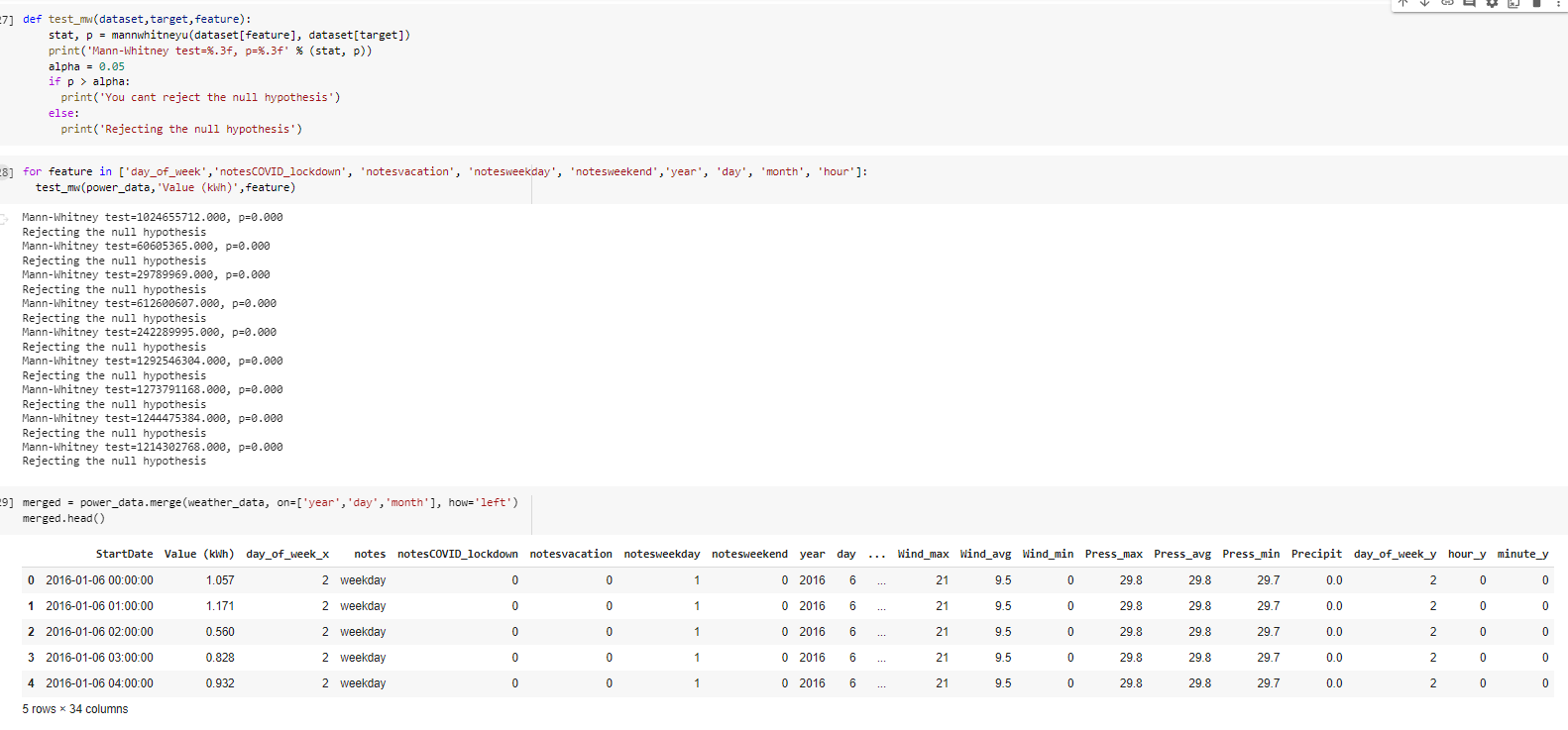
The function has the capability to accept an extra hue parameter, which exhibits the dispersion of the target variable based on another categorical variable.



**Figure 9: Correlation plot of power data**

(Source: Obtained from Google Collab)

Upon function definition, the script employs the Plot\_Timeseries() function to graph the power consumption data as a function of time, revealing a conspicuous positive trend in energy usage. Subsequently, the script employs the aforementioned function to generate a graph that displays the power consumption data as a function of months and years. The resulting plot illustrates a recurring trend in power usage, with the highest levels observed during winter months and the lowest levels during summer months. The plot indicates a noticeable upward trend in power consumption over time.



**Figure 10: Merging dataset**

(Source: Obtained from Google Collab)

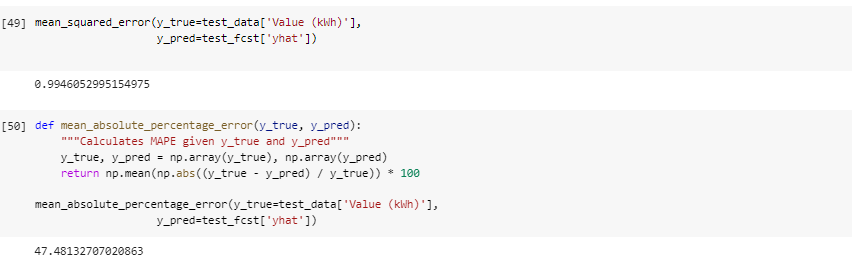
The script employs the Plot\_Timeseries() function to graph the power consumption data in relation to the days of the week. The results indicate that power usage is at its peak during weekdays, while it is at its lowest during weekends. The plot indicates that power consumption remains relatively consistent from Monday through Friday, but experiences a notable decline during the weekends.



**Figure 11: Fitting into the Prophet model**

(Source: Obtained from Google Collab)

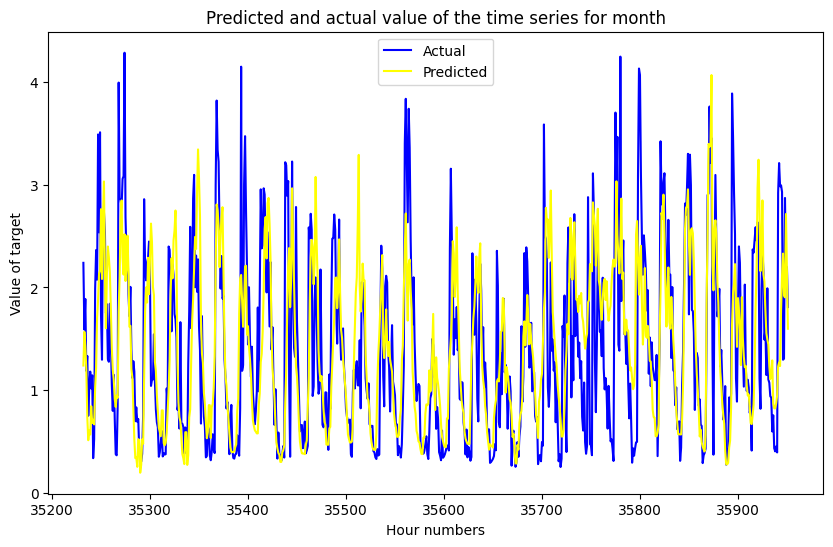
The above figure shows the procedure for Fitting the merged dataset into the Prophet model.



**Figure 11: percentage error and mse value**

(Source: Obtained from Google Collab)

The above figure shows the percentage error and mse value of the merged time series data.



**Figure 12: Predicted and actual value of the time series for the month**

(Source: Obtained from Google Collab)

The above figure shows the Predicted and actual value of the time series for the month on the basis of the dataset.

# Conclusion

In conclusion, optimising energy generation in renewable energy systems through data-driven simulation utilizing Python is a complex and multifarious process requiring a rigorous and exhaustive research strategy. Pragmatism-based research would emphasise the significance of practical outcomes and solutions and involve collaboration among researchers, stakeholders, and practitioners. The research design would be adaptable and iterative, prioritising experimentation, observation, and experience to develop a predictive model that accurately reflects the complex relationships between weather conditions, energy production, and maintenance requirements in renewable energy systems. Quantitative and qualitative data collection techniques would be utilised, with purposive sampling ensuring that participants with pertinent expertise and experiences are selected. To ensure that the research is conducted in an ethical and responsible manner, ethical considerations such as data privacy, consent, and transparency would also be crucial.

The resultant predictive model would be used to optimise energy generation in renewable energy systems and enhance their efficiency, resilience, and sustainability. It would also contribute to the development of more advanced and sophisticated renewable energy systems that are better equipped to satisfy future energy needs. Optimisation of energy generation in renewable energy systems through data-driven simulation utilising Python represents a crucial move towards a more sustainable and equitable energy future. It necessitates a collaborative, interdisciplinary, and ethical strategy that prioritises practical outcomes and solutions and aims to bridge the divide between theory and practise. We can realise the full potential of renewable energy for a more sustainable future if we develop a predictive model that accurately reflects the complex relationships between weather conditions, energy production, and maintenance needs for renewable energy systems.

# References

Ahmad, T., Madonski, R., Zhang, D., Huang, C. and Mujeeb, A., 2022. Data-driven probabilistic machine learning in sustainable smart energy/smart energy systems: Key developments, challenges, and future research opportunities in the context of smart grid paradigm. Renewable and Sustainable Energy Reviews, 160, p.112128.

Shen, F., Zhao, L., Du, W., Zhong, W. and Qian, F., 2020. Large-scale industrial energy systems optimization under uncertainty: A data-driven robust optimization approach. Applied Energy, 259, p.114199.

Sun, L. and You, F., 2021. Machine learning and data-driven techniques for the control of smart power generation systems: An uncertainty handling perspective. Engineering, 7(9), pp.1239-1247.

Xu, C., Ke, Y., Li, Y., Chu, H. and Wu, Y., 2020. Data-driven configuration optimization of an off-grid wind/PV/hydrogen system based on modified NSGA-II and CRITIC-TOPSIS. Energy Conversion and Management, 215, p.112892.

Xu, X., Yan, Z., Shahidehpour, M., Li, Z., Yan, M. and Kong, X., 2019. Data-driven risk-averse two-stage optimal stochastic scheduling of energy and reserve with correlated wind power. IEEE Transactions on Sustainable Energy, 11(1), pp.436-447.

Zhang, G., Hu, W., Cao, D., Liu, W., Huang, R., Huang, Q., Chen, Z. and Blaabjerg, F., 2021. Data-driven optimal energy management for a wind-solar-diesel-battery-reverse osmosis hybrid energy system using a deep reinforcement learning approach. Energy conversion and management, 227, p.113608.

Zhao, N. and You, F., 2021. New York State's 100% renewable electricity transition planning under uncertainty using a data-driven multistage adaptive robust optimization approach with machine-learning. Advances in Applied Energy, 2, p.100019.