Machine Learning Project Report

CIS4035-N-FJ1-2021 Machine Learning

Name: Ajay Kuruvilla Johnson

Student ID: W9521662

School of Computing, MSc Data Science

Module Leader: Dr. Alessandro Di Stefano

Project Title:

Electricity demand forecast for France on an hourly level up to 48 hours ahead using time series and regression analysis (EnAppSys Dataset)

Word count: 2423 (Excluding References)

Submission Date: 27/05/2022

Abstract:

Electricity demand is a vital parameter in today's day and age. The electricity demand in France is highly dependent on the weather. In Europe, France is the biggest country and hence has many neighboring countries. Hence, in times of high electricity demand, they import electricity, leading to higher prices. In times of low demand, France exports electricity to neighboring countries thereby we can see a reduction in prices. Time series analysis is an interesting and extremely useful technique used in Machine Learning. Forecasting electricity demand into the future can help a great deal for companies in the energy business to get a picture of the trend and fluctuation of the demand yearly, monthly even a weekly time frame.

In this paper, the use of various machine learning techniques with Regression and Time-Series analysis has been done on the dataset given by the company EnAppSys Ltd. The data consists of hourly electricity demand [MW] starting from Jan 01, 2017, till March 8th, 2022 with various other features.

Using Univariate and Multivariate analysis different parameters of the dataset and the relationship it has on the demand are analyzed and significant conclusions have been made.

Regression Algorithms and a Time Series algorithm (SARIMAX) have been applied to the dataset. There are 6 Regression Algorithms used. Among the 6 used, the XGBoost Regressor has been seen to have the best accuracy score of 97.53%.

For Time-Series Analysis, the Seasonal Auto-Regressive Integrated Moving Average (SARIMA) model has been used to successfully forecast 48 hours into the future.

1. Introduction:

The purpose of this project is to successfully create a demand forecast for France on an hourly level up to 48 hours ahead.

Ideally, the dataset would be trained on many features, especially temperature. I have done a univariate time series analysis with SARIMA so that my model is lean to accurately predict demand based on the historical dataset given.

Nevertheless, a simple multivariate analysis with a grouped bar chart portraying demand for Weekdays vs weekends for a certain temperature has been done to get some insights.

This research can help in forecasting future data and help in allocating resources and can help business leaders make better decisions.

The motivation behind taking this project is to learn more about time series analysis and techniques as it is a very interesting and useful skill to understand and master.

1.1 Dataset Description:

The dataset is given by the Head of Analytics at EnAppSys BV.

It has 45432 rows and 17 columns.

With the 'demand (MW)' column taken as the target variable, several features of temperature, wind speed, cloud cover, etc. are present.

The column denoting the time stamp is given on an hourly basis for 5+ years. i.e Starting from '2017-01-01 00:00:00' to '2022-03-08 23:00:00'.

2. Methods

2.1 Data Exploration and Pre-processing:

- Basic data pre-processing in MS Excel to add an extra column to denote 'Weekday' or 'Weekend' has been created to visualize and analyze data better.
- Some of the columns had null values. They were filled in by taking the mean values for those respective columns.
- Setting the 'Time' column as an index and converting the DateTime format.
- Created copies of main data so that we can revert to the original copy when in need of other analysis.

```
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Time                          45432 non-null  object
 1   demand                        45432 non-null  float64
 2   solar_actual(MW)              45432 non-null  float64
 3   solar_forecast(MW)            45432 non-null  float64
 4   solar_inferred_capacity(MW)   45432 non-null  float64
 5   wind_actual(MW)               45432 non-null  float64
 6   wind_inferred_capacity(MW)    45432 non-null  float64
 7   albedo(%)                     45432 non-null  float64
 8   cloud_cover(%)                45432 non-null  float64
 9   frozen_precipitation(%)       45432 non-null  float64
 10  pressure(Pa)                  45432 non-null  float64
 11  radiation(W/m2)               45432 non-null  float64
 12  air_tmp(Kelvin)               45432 non-null  float64
 13  ground_tmp(Kelvin)            45432 non-null  float64
 14  apparent_tmp(Kelvin)          45432 non-null  float64
 15  wind_direction(angle)         45432 non-null  float64
 16  wind_speed(m/s)               45432 non-null  float64
 17  Day_of_the_week               45432 non-null  object
dtypes: float64(16), object(2)
memory usage: 6.2+ MB
```

2.2 Univariate and Multivariate analysis

A visualization of how each column has been done with the use of seaborn and matplotlib plots.

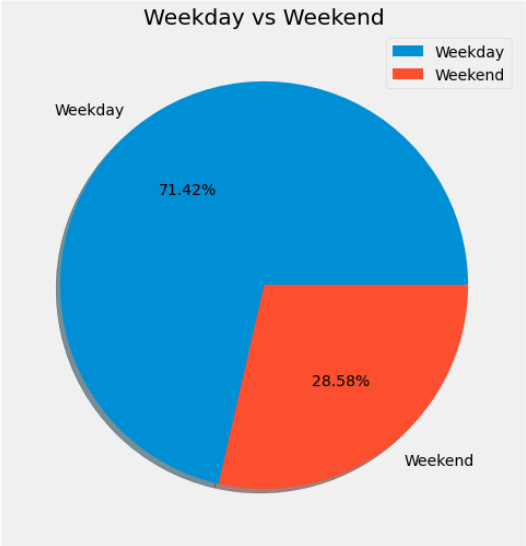

Fig 1: A univariate analysis of the number of Weekdays vs Weekends in the dataset using a piechart

From fig 1, we see that the number of Weekdays is 71.42% compared to 28.58%. The data for both are significant enough and trends are analyzed on the electricity demand for these days.

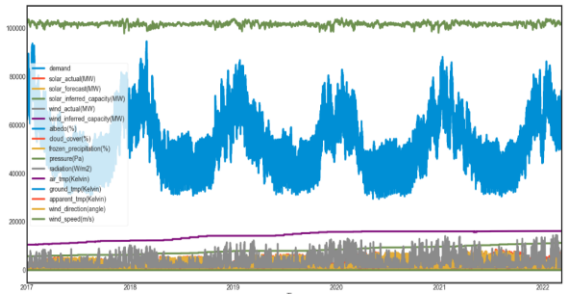

Fig 2: A multi-variate analysis of the entire dataset visualized by a multi-line chart.

From Fig 2, we see a plot of all the features given in the dataset. The insight drawn from here is how demand varies over the years. We can safely say from Fig 3 that the demand is higher during the winter seasons and during the summer months of June, July, and August the electricity demand is low.

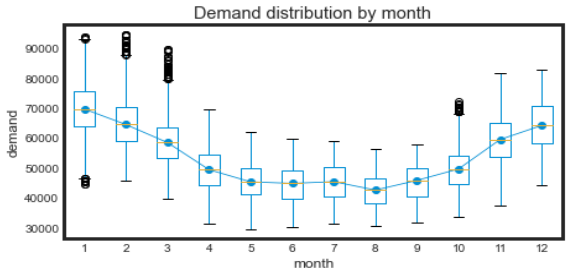

Fig 3: A view of the demand distribution month-wise using a boxplot.

This is because solar and wind energy generation increases during the summers with better sunlight and wind patterns in producing clean renewable energy which supplements conventional electricity production. We observe that in August the demand is the least.

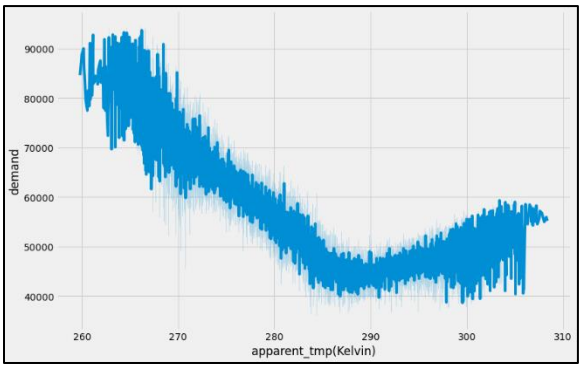

Fig 4: A line plot between demand (MW) vs apparent temperature (Kelvin)

In Figure 4, we can see that the cooler the temperature, the electricity demand is which coincides with our insights on electricity demand being higher in winters. We see a sharp decline in the demand as the temperature rises from 270 Kelvin (-3.25 °Celsius) to 285 Kelvin (6.85 °Celsius).

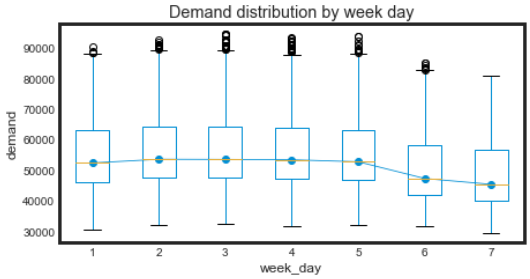

Fig 5: A box plot between demand (MW) vs ground temperature (Kelvin)

Figure 5 is a representation of the electricity demand over a week. We can see that 6 and 7 which represent 'Saturday' and 'Sunday' i.e Weekends have less electricity demand compared to the weekdays.

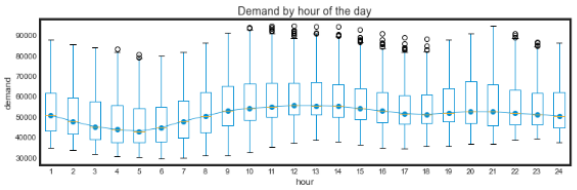

Fig 6: A box plot between demand (MW) vs hour of the day

Figure 6 is a representation of the electricity demand and how it varies in a span of 24 hours.

We see that the early hours of the day such as 4 am – 5 am have the lowest electricity demand compared to the other hours. The demand is generally higher in the waking hours when most people are active and need electricity to carry out their daily activities.

2.3 Heatmap and Correlation

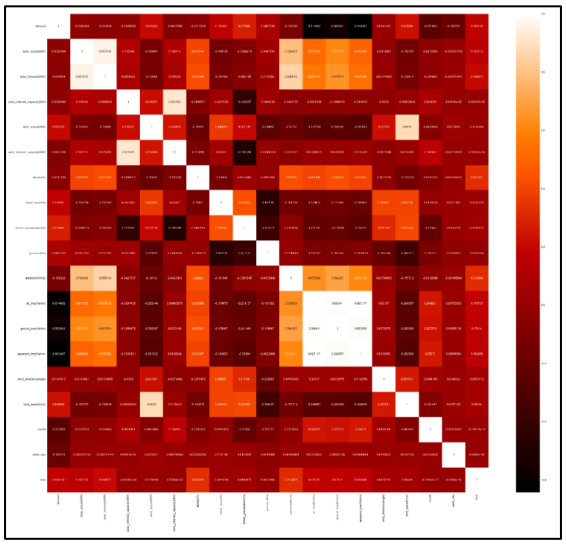

Fig 7: A heatmap is plotted to showcase correlation.

demand	1.000000
apparent_tmp(Kelvin)	0.643467
air_tmp(Kelvin)	0.614909
ground_tmp(Kelvin)	0.563841
frozen_precipitation(%)	0.273865
month	0.251063
wind_speed(m/s)	0.225889
wind_actual(MW)	0.200233
week_day	0.185751
hour	0.182119
cloud_cover(%)	0.178167
radiation(W/m2)	0.129322
pressure(Pa)	0.086730
wind_inferred_capacity(MW)	0.080777
albedo(%)	0.051724
solar_forecast(MW)	0.044934
solar_inferred_capacity(MW)	0.032910
solar_actual(MW)	0.032035
wind_direction(angle)	0.014151
Name: demand, dtype: float64	

Fig 8 Correlation arranged in ascending order.

From Fig 7 and Fig 8 we can conclude how important the temperature features are in determining electricity demand.

2.4 Temperature as a factor:

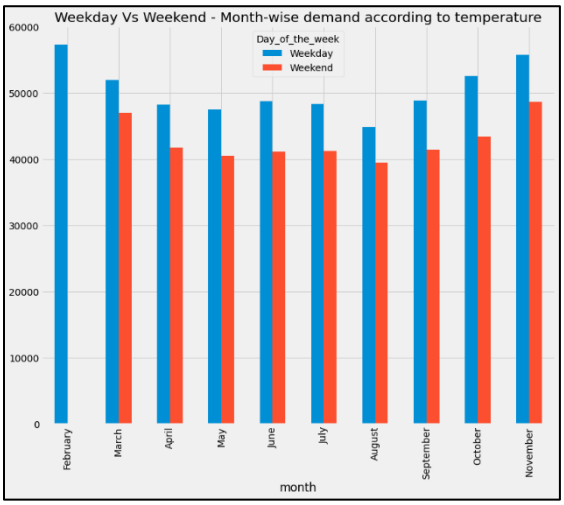

Fig 9: Analyzing apparent temperature (Celsius) >= 15 °C with demand on weekdays and weekends.

From Figure 9, the graph is plotted with an apparent temperature >= 15 °C. It is interesting to note that in February, the demand on weekends is not visible because the temperatures are seemingly lower on the weekends in February. We can see that January and December are not even in the picture due to cooler temperatures.

Algorithm	Accuracy	MAE	MSE	RMSE	R2_Score
XGBoost Regressor	97.53%	0.11	0.03	0.158121	0.975295
Random Forest Regressor	96.62%	0.13	0.03	0.18488	0.966226
KNN Regressor	95.06%	0.15	0.05	0.22363	0.950585
Decision Tree Regressor	92.90%	0.18	0.07	0.268058	0.929
Adaboost Regressor	82.90%	0.34	0.17	0.416024	0.828983
Linear Regression	72.20%	0.42	0.28	0.530375	0.722048

3 Analysis

Both regression and time series analysis has been done to best understand and train the dataset

3.1 Regression Analysis:

For this, we split the data into 80% training data and 20% testing data. In total, 6 regression algorithms are being used.

1. Adaboost Regressor
2. Decision Tree Regressor
3. Linear Regression
4. Random Forest Regressor
5. XGBoost Regressor
6. KNN Regressor (K Nearest Neighbors)

Metrics Used to Evaluate a Regression Model:

1. Mean Absolute Error (MAE)
2. Mean Squared Error (MSE)
3. Root Mean Square Error (RMSE)
4. R2_Score (Coefficient of Determination)

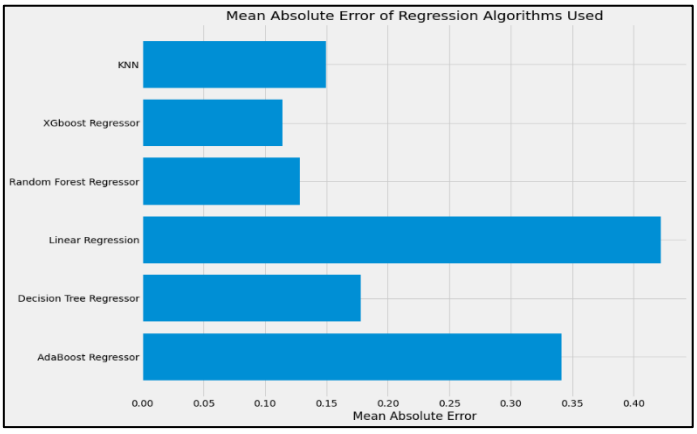

Fig 10(a): Mean Absolute Error of Regression Algorithms Used

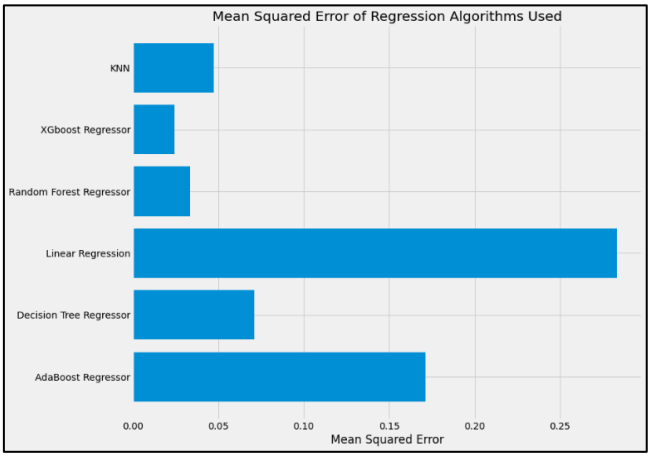

Fig 10(b): Mean Squared Error of Regression Algorithms Used

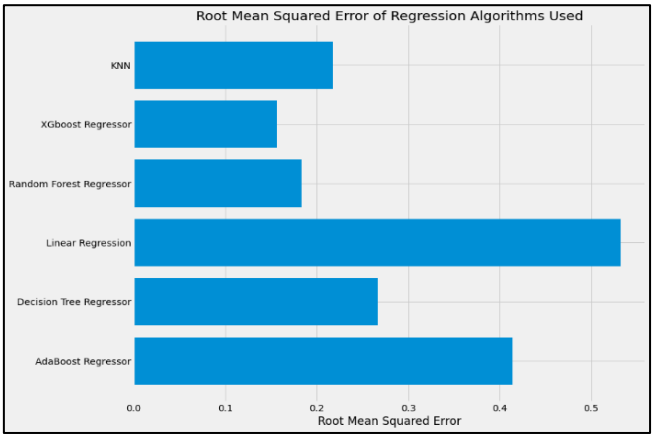

Fig 10(c): Root Mean Squared Error of Regression Algorithms Used

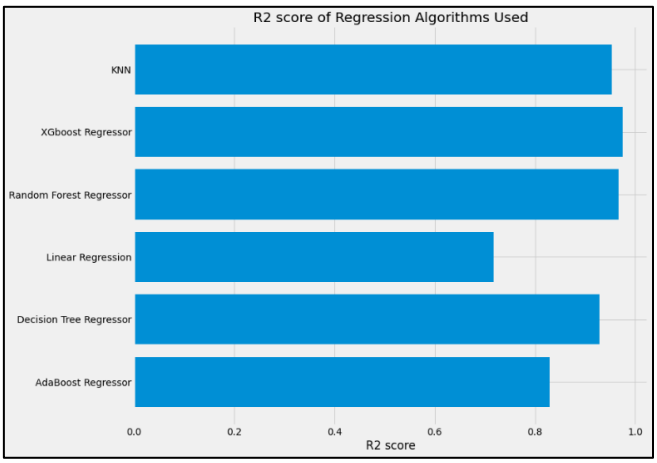

Fig 10(d): R2 Score of Regression Algorithms Used

3.2 Time Series Analysis:

For this analysis, I have chosen the Autoregressive Integrated Moving Average (ARIMA). ARIMA is one of the most widely used and effective machine learning algorithms to perform univariate time series forecasting.

When the seasonal component is added to ARIMA it is named SARIMA.

- AR – Autoregressive (p)
- I – Integrated (d) (Differentiation)
- MA – Moving Average (q)

Now, there are 4 seasonal elements as well:

- P – Seasonal Autoregressive order
- D – Seasonal Difference Order
- Q – Seasonal Moving Average
- M – No. for a seasonal period

We would be doing a univariate time series analysis, so only the DateTime column and the demand (MW) would be taken into consideration for maximum accuracy.

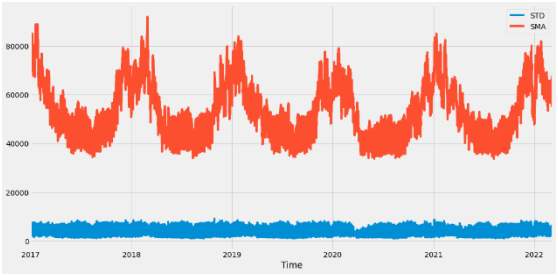

Fig 11: Plotting the (SMA) to understand the trend and also the Standard Deviation (STD) curve

Fig 11: Shows a visualization of plotting (SMA) – Simple Moving Average and Standard Deviation (STF) curve to understand trend and variance.

We can see that the Standard Deviation does not change much and hence there is not much variance.

3.2.1 Decomposition of the time series data to its trend, seasonality, and residual components.

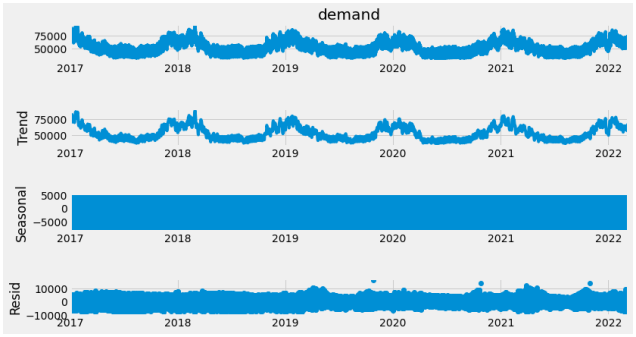

Fig 12: Visualizing the decomposition of the time series data to its trend, seasonality, and residual components.

From Figure 12, we can see that the data is stationary. But we will do a stationarity test to confirm the same.

3.2.2 Augmented Dickey-Fuller (ADF) Test for Stationarity

One of the most commonly used tests for Stationarity.

If the p-value <= 0.05 then we can discard the null hypothesis, the data is stationary.

From analysis we get:
p-value: 8.813621317254258e-11

Hence, we can discard the null hypothesis, so the data is stationary.

"Null Hypothesis: If failed to be rejected, it suggests the time series has a unit root, meaning it is non-stationary. It has some time-dependent structure."

Since data is stationary, we do not have to do a differentiation(d). Hence d = 0.

Using the pmdarima module which helps us to identify the differentiation (d) value for applying ARIMA, I reconfirmed that d = 0

3.2.3 Autocorrelation (ACF) and Partial Autocorrelation (PACF) plots:

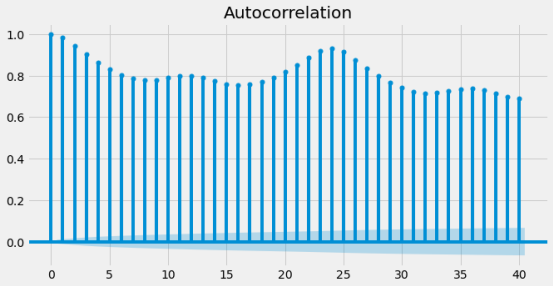

Fig 13: Autocorrelation (ACF) plot

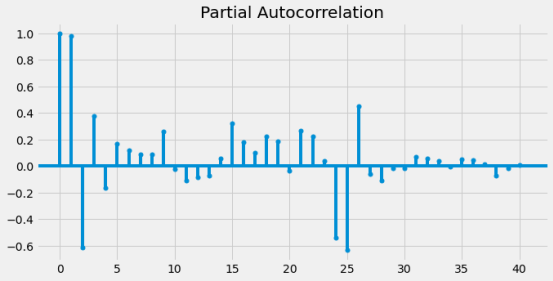

Fig 14: Partial Autocorrelation (PACF) plot

The ACF and PACF plots are extremely crucial to get the q and p values respectively for the ARIMA model.

After carefully analysing the ACF and PACF plots in fig 13 and fig 14 along with some trial and error methods, I have concluded the values for (p,d,q) as (1,0,1) respectively.

Since the data follows a yearly cycle, the m value would be 12.

3.2.4 SARIMA:

SARIMAX Results

Dep. Variable:	demand	No. Observations:	45432
Model:	SARIMAX(1, 0, 1)x(1, 0, 1, 12)	Log Likelihood	-385754.534
Date:	Thu, 26 May 2022	AIC	771519.068
Time:	00:02:24	BIC	771562.688
Sample:	01-01-2017 - 03-08-2022	HQIC	771532.793
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.9972	0.001	745.339	0.000	0.995	1.000
ma.L1	0.6614	0.001	728.222	0.000	0.660	0.663
ar.S.L12	1.0000	7.64e-06	1.31e+05	0.000	1.000	1.000
ma.S.L12	-0.9388	0.002	-564.115	0.000	-0.942	-0.936
sigma2	1.518e+06	1.61e-10	9.45e+15	0.000	1.52e+06	1.52e+06

Ljung-Box (L1) (Q):	164.47	Jarque-Bera (JB):	1344785.54
Prob(Q):	0.00	Prob(JB):	0.00
Heteroskedasticity (H):	1.10	Skew:	-0.43
Prob(H) (two-sided):	0.00	Kurtosis:	29.64

Fig 15: SARIMAX Results

From Figure 15, I was concerned about the high AIC and BIC values (Akaike and Bayesian Information Criterion) and faced a challenge to understand if I was on the right track. Hence, I decided to plot the results and diagnosed the outcome along with finding out the metrics such as MAE, MSE, RMSE, and R2_score.

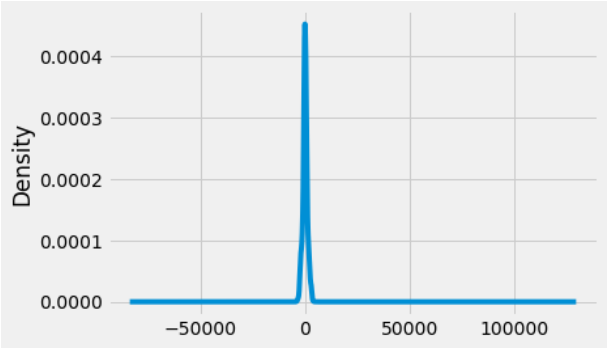

Fig 16: A Kernel Density plot (KDE) of the results

From Figure 16, we see that the KDE plot has a peak at 0, which shows that there is very less bias in the prediction

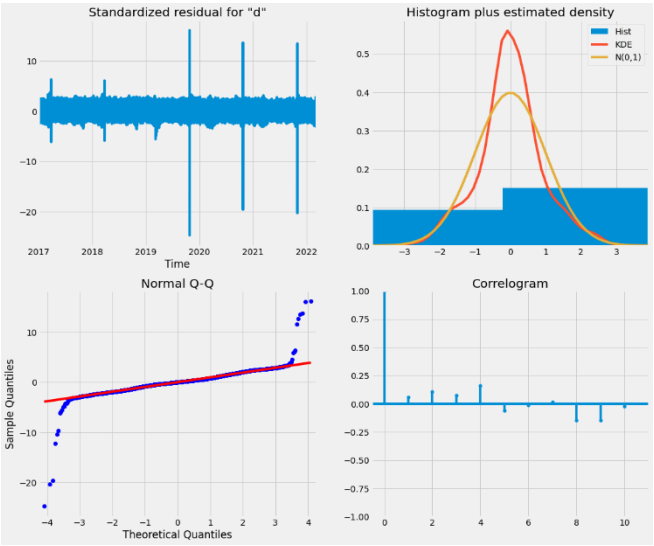

Fig 17: A visualization of the diagnostics of the SARIMA results

In Figure 17, at the top left corner, the standardized residuals move around a mean of zero. The bottom left charts a correlation which shows a normal distribution all along the red line The top-right plot is a histogram plus estimated density plot, it should ideally show a histogram occurring normally at 0, while here in the figure, it does not do so. Lastly, the bottom right graph is a Correlogram which generally depicts the randomness in a dataset. If the values are closer to zero, it means that the data has very less randomness. In this case, it is between the range of (-0.2) to (0.2), so there is room for improvement.

Algorithm	ARIMA (1,0,1) (1,0,1,12)
Accuracy	98.92%
MAE	853.68
MSE	1510325.48
RMSE	1228.953
R2_Score	0.98917

3.2.5 Validating forecasts

First, we try to forecast accurately with the given data to see if the forecast and real data match.

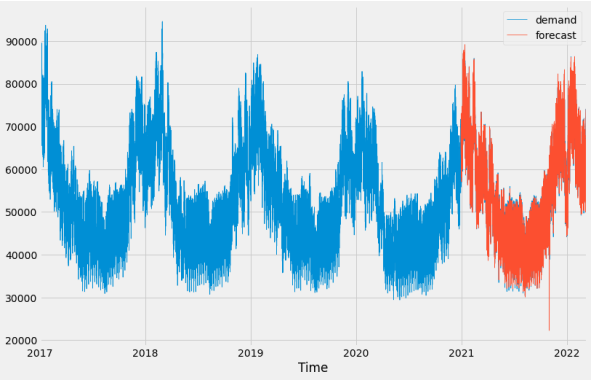

Fig 18: Forecast with available data

From figure 19, we see that electricity demand has been forecasted for 48 hours into the future. i.e Forecasted electricity demand for **'2022-03-09 00:00:00' to '2022-03-10 23:00:00'.**

4 Results

Out of all the algorithms used, we see that the seasonal ARIMA algorithm has the highest accuracy of 98.92%

In the Regression algorithms, the XGBoost Regressor has the highest accuracy with 97.53%.

Total Algorithms Used	Accuracy
ARIMA (1,0,1) (1,0,1,12)	98.92%
XGBoost Regressor	97.53%
Random Forest Regressor	96.62%
K Nearest Neighbors Regressor	95.06%
Decision Tree Regressor	92.90%
AdaBoost Regressor	82.90%
Linear Regression	72.20%

Fig 20: Accuracy of all algorithms arranged in descending order

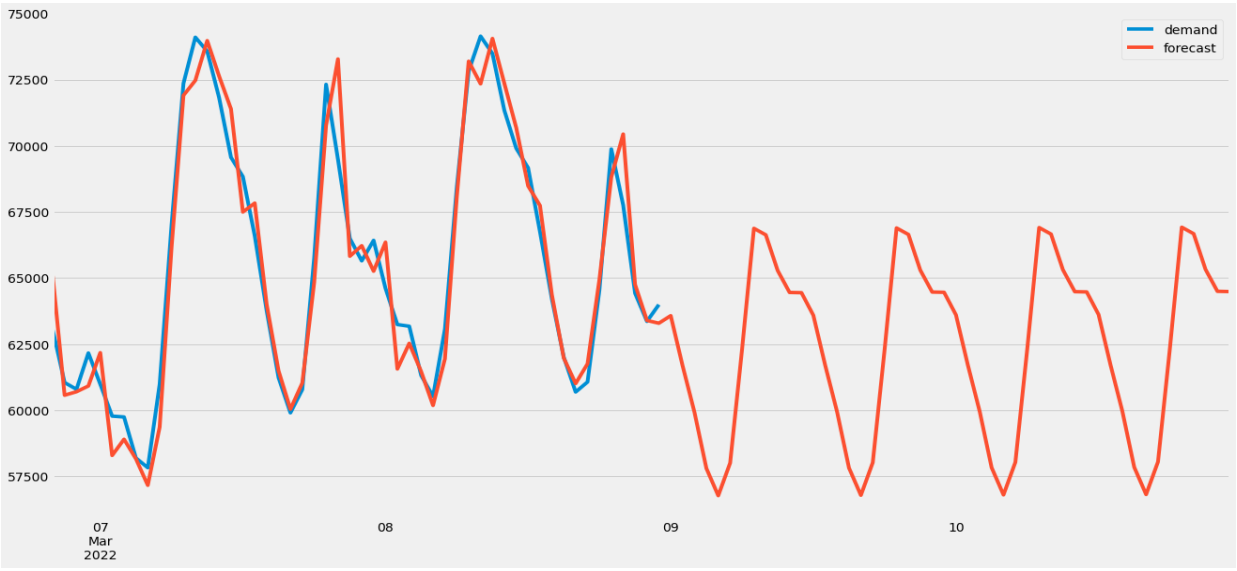

Fig 19: Forecasting future data for 48 hours

Algorithm	Accuracy	MAE	MSE	RMSE	R2_Score
ARIMA (1,0,1) (1,0,1,12)	98.92%	853.68	1510325.48	1228.953	0.98917
XGBoost Regressor	97.53%	0.11	0.03	0.158121	0.975295
Random Forest Regressor	96.62%	0.13	0.03	0.18488	0.966226
KNN Regressor	95.06%	0.15	0.05	0.22363	0.950585
Decision Tree Regressor	92.90%	0.18	0.07	0.268058	0.929
Adaboost Regressor	82.90%	0.34	0.17	0.416024	0.828983
Linear Regression	72.20%	0.42	0.28	0.530375	0.722048

Fig 21: All metrics of all algorithms arranged in descending order of accuracy

5 Discussion

For discussion, I will briefly explain the top 3 performing algorithms based on the results.

- **Seasonal ARIMA (SARIMA):**

Out of many available time series algorithms, I chose SARIMA as it was effective and easier to comprehend and apply to my dataset. I was considering other algorithms like Prophet and LSTM but both were slightly more complex and ARIMA seemed comparatively easier to understand and apply.

The limitations of the ARIMA model are that we need to understand the data well to get the right (p,d,q) values which are not necessarily needed for other time series algorithms. It took a lot of trial and testing to get the values right.

Since ARIMA performs best with univariate time series, I could not add temperature features to train the model.

- **XGBoost (Extreme Gradient Boosting) Regressor:**

This is one of the powerful algorithms which can be applied to regression models. It uses advanced regularization techniques which helps in improving model generalization. It's also generally known to perform well with large datasets.

The limitation of this algorithm is that gradient boosting is sensitive to outliers and unstructured data.

- **Random Forest Regressor:**

One of the best algorithms in machine learning as it does not matter if you have a regression or classification problem. It's good at handling outliers.
Limitations are that it tends to overfit if the necessary standardization steps are not taken.

6 Conclusion

1. We can see that Seasonal ARIMA has the highest accuracy (98.92%) and is best suited for this dataset compared to the other algorithms used.

2. Temperature data is also available in the given dataset and it would be really interesting to add temperature as a predictor. However, to do so I believe would not be correct as we do not know the future temperature. Nevertheless, if needed, we can do so by training the model on the forecasted temperature rather than the actual temperature. Multivariate Time series analysis can be done using Vector Auto Regression (VAR).

7 Future Work:

I would like to learn more about LSTM time series models and Vector Auto Regression (VAR) so that I can include temperature parameters in the time series model. It would be interesting to see how an increase/decrease in temperature values can result in a change in electricity demand.

8 References:

1. Joaquín A R, Javier E O (March 2022) "Forecasting electricity demand with Python." Available at: https://www.cienciadedatos.net/documentos/py29-forecasting-electricity-power-demand-python.html (Accessed: 20 April, 2022)

2. Editorial Team (February 18, 2021) "Electricity production forecasting using ARIMA model in Python." Available at: https://towardsai.net/p/data-visualization/electricity-production-forecasting-using-arima-model-in-python (Accessed: 02 May, 2022)

3. Nagesh S C (January 2, 2020) "Predict Electricity Consumption Using Time Series Analysis" Available at: https://www.kdnuggets.com/2020/01/predict-electricity-consumption-time-series-analysis.html (Accessed: 12 May, 2022)

4. Khoa L (February 27, 2020) "Time Series Analysis and Weather Forecast in Python" Available at: https://medium.com/@llmkhoa511/time-series-analysis-and-weather-forecast-in-python-e80b664c7f71 (Accessed: 12 May, 2022)

5. Guest (July 29, 2020) "Univariate Data Visualizations With Illustrations in Python" Available at:https://www.analyticsvidhya.com/blog/2020/07/univariate-analysis-visualization-with-illustrations-in-python/#:~:text=HISTOGRAMS%20%3A,spread%20of%20continuous%20sample%20data (Accessed: 12 May, 2022)

6. Seema S (May 21, 2018) "Understanding the Bias-Variance Tradeoff" Available at: https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229 (Accessed: 12 May, 2022)

7. Datascience George (April 29, 2020) "A Brief Introduction to ARIMA and SARIMAX Modeling in Python" Available at: https://medium.com/swlh/a-brief-introduction-to-arima-and-sarima-modeling-in-python-87a58d375def (Accessed: 10 May, 2022).

8. Julia K (October 19, 2018) "Why Random Forest is My Favorite Machine Learning Model" Available at: https://towardsdatascience.com/why-random-forest-is-my-favorite-machine-learning-model-b97651fa3706 (Accessed: 18 May, 2022).

9. Jason B (August 17, 2018) "A Gentle Introduction to SARIMA for Time Series Forecasting in Python" Available at: https://machinelearningmastery.com/sarima-for-time-series-forecasting-in-python/ (Accessed: 05 May, 2022).

10. Jason B (December 30, 2016) "How to Check if Time Series Data is Stationary with Python" Available at: https://machinelearningmastery.com/time-series-data-stationary-python/ (Accessed: 11 May, 2022)