Energy Consumption Forecasting in Smart Homes

A project report submitted to ICT Academy of Kerala

in partial fulfillment of the requirements

for the certification of

CERTIFIED SPECIALIST

IN

DATA SCIENCE & ANALYTICS

submitted by

Team Members:

Anandhu R S
Gazal Vahab
Lijith T
Shaima A
Stephy Mary Thomas

ICT ACADEMY OF KERALA

THIRUVANANTHAPURAM, KERALA, INDIA

Nov 2022

List of Figures

List of Abbreviations

SL No.	Abbreviation	Definition
1	ARIMA	Auto Regressive Integrated Moving Average
2	IOT	Internet of Things
3	kW	Kilo Watt
4	ADF	Augmented Dickey Fuller
5	AIC	Akaike's Information Criterion
6	AR	Auto Regression
7	RMSE	Root Mean Squared Error

Table of Contents

Abstract

Electricity is being consumed in every home at present time. We are using electricity for lots of purposes like watching television, charging cell phones, Electric bulbs and many more things. Analyzing this electricity data will give us a better understanding of how consumers consume electricity in their daily life. In this project we will be analyzing the appliance usage in the house gathered via home sensors. All readings are taken at 1 min. intervals for 350 days.

The goal is to predict future energy consumption with the current weather data.Analysis of this data collected from the IoT devices can help in monitoring the energy consumption patterns and in turn control the energy consumption more efficiently.In this Case study we are going to concentrate on Models Like ARIMA and Isolation Forest that will help us to get some more insight from the data and also build a model to predict the future needs. So that we can manage our day to day usage of appliances at home.

1. Problem Definition

1.1 Overview

Over the last few years, activity recognition in the smart home has become an active research area due to the wide range of human centric-applications. IoT brings together everything at home under one umbrella which has the potential to monitor and remote control such as air conditioning, alarm system, lighting, heating, ventilation, telephone system, tv, etc. To enhance our comfort and security with low energy consumption and energy management is one of the IoT use cases with which energy being sent out or consumed can be monitored. In this case study we are going to focus on predicting the future energy consumption with the past data so that we can manage our day to day usage of appliances at home

1.2 Problem Statement

The energy generation and consumption varies with weather attributes like temperature, precipitation etc. For example, during the winter, people use heaters and the use of air conditioning drops drastically and vice versa. This Case Study aims at predicting this change in energy consumption according to the weather based on the readings taken by a smart meter with a time span of 1 minute of 350 days of house appliances in kW from a smart meter and weather conditions of that particular region. Analysis of the data collected from the IoT devices can help in monitoring the energy consumption patterns and in turn control the energy consumption more efficiently.

In descriptive statistics, a time series is defined as a set of random variables ordered with respect to time. Time series are studied both to interpret a phenomenon, identifying the components of a trend, cyclicity, seasonality and to predict its future values. In this Case study we are going to focus on models to get some more insight from the data and also build a model to predict the future needs.

2. Introduction

IoT brings together everything at home under one umbrella which has the potential to monitor and remote control such as air conditioning, alarm system, lighting, heating, ventilation, telephone system, tv, etc. to enhance our comfort and security with low energy consumption. The home is a specific environment, and energy management is one of the IoT use cases with which energy being sent out or consumed can be monitored. One can monitor each of the IoT appliances and how much power each of the devices is consuming, and easily switch between energy-efficient appliances across the day. The energy generation and consumption varies with weather attributes like temperature, precipitation etc. For example, during the winter, people use heaters and the use of air conditioning drops drastically and vice versa.

The project aims at predicting this change in energy consumption according to the weather based on the readings taken by a smart meter with a time span of 1 minute of 350 days of house appliances in kW from a smart meter and weather conditions of that particular region. Analysis of the data collected from the IoT devices can help in monitoring the energy consumption patterns and, in turn, control the energy consumption more efficiently.

2.1 Data Used

- The dataset we have used for this project is titled “Smart Home Dataset with Weather Information”, which has been downloaded from https://www.kaggle.com/taranvee/smart-home-dataset-with-weather-information.
- The dataset contains 503911 rows and 32 columns. Readings taken within the time-span of 1 minute of 350 days the energy used (in kW) by the appliances of a smart home in the year 2016, and the weather conditions of the area at that time.

3. METHODOLOGY

3.1 Exploratory data analysis :

The dataset we have used for this project is titled “Smart Home Dataset with Weather Information”, which has been downloaded from Kaggle. This dataset has 32 columns and more than 500,000 readings with the time-span of 1 minute of the energy used (in kW) by the appliances of a smart home, and the weather conditions of that area at that time.

We have visualized the time series data to find the pattern of energy consumption by the devices in a smart home. The below plot shows the highest energy consumption and duplicates.
We can do some analysis on the dataset to find important features or some patterns to make better understanding of the data to build model that will predict the future.
As we already modified our data into two groups for energy and weather now we can visualize how these split helps us to understand the data well.
From the plot there is a clear sign that Furnace uses highest amount of energy in a day throughout the year.
Home office used the Maximum energy in the calendar year and Kitchen uses the lowest energy.

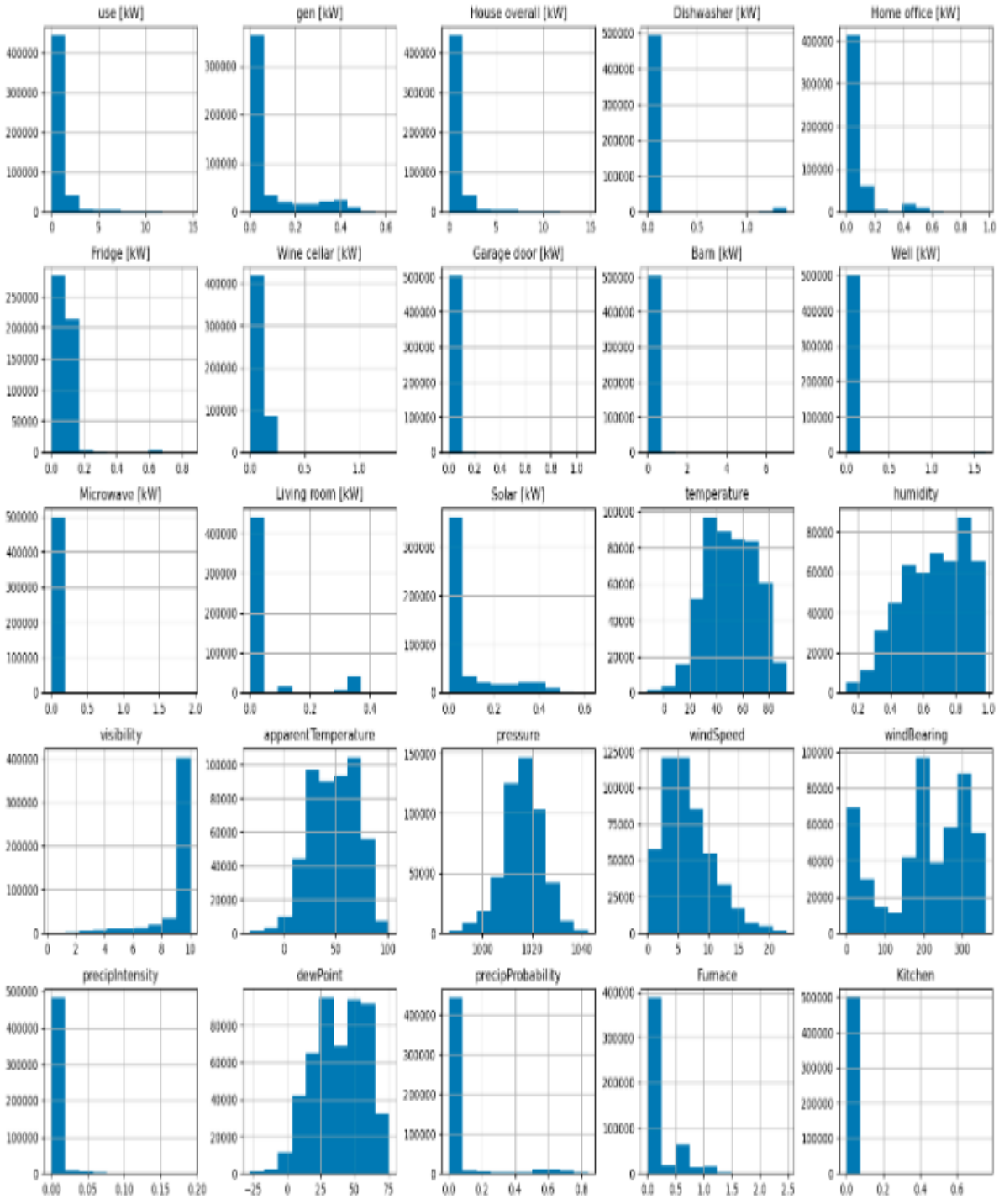


Fig 3.1.1 Histogram of Energy data

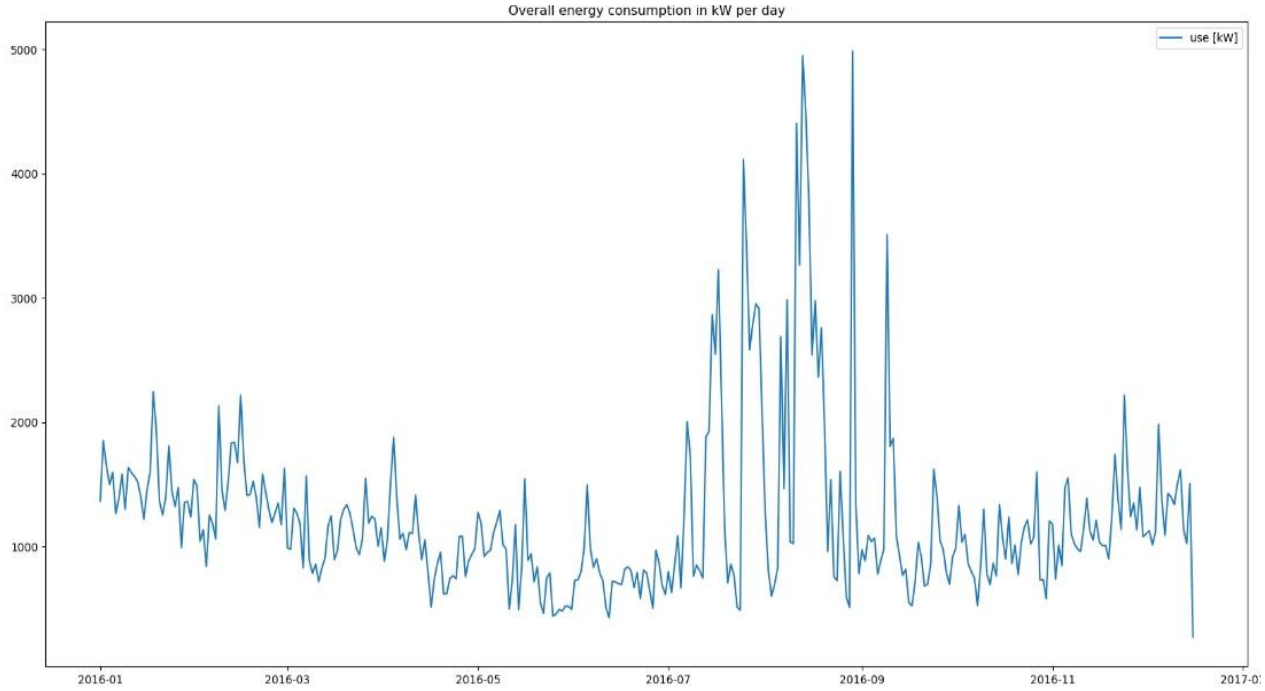


Fig 3.1.2 Overall Energy Consumption in kW per Day

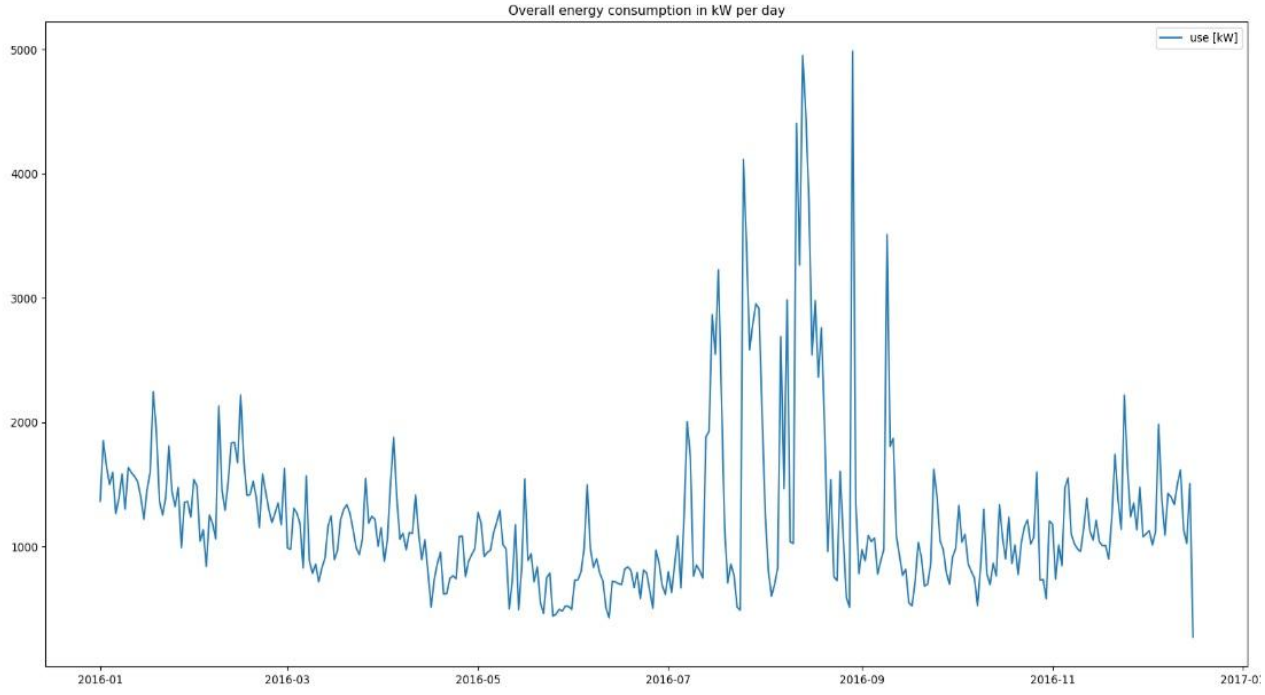


Fig 3.1.3 Overall Energy Consumption in kW per Week

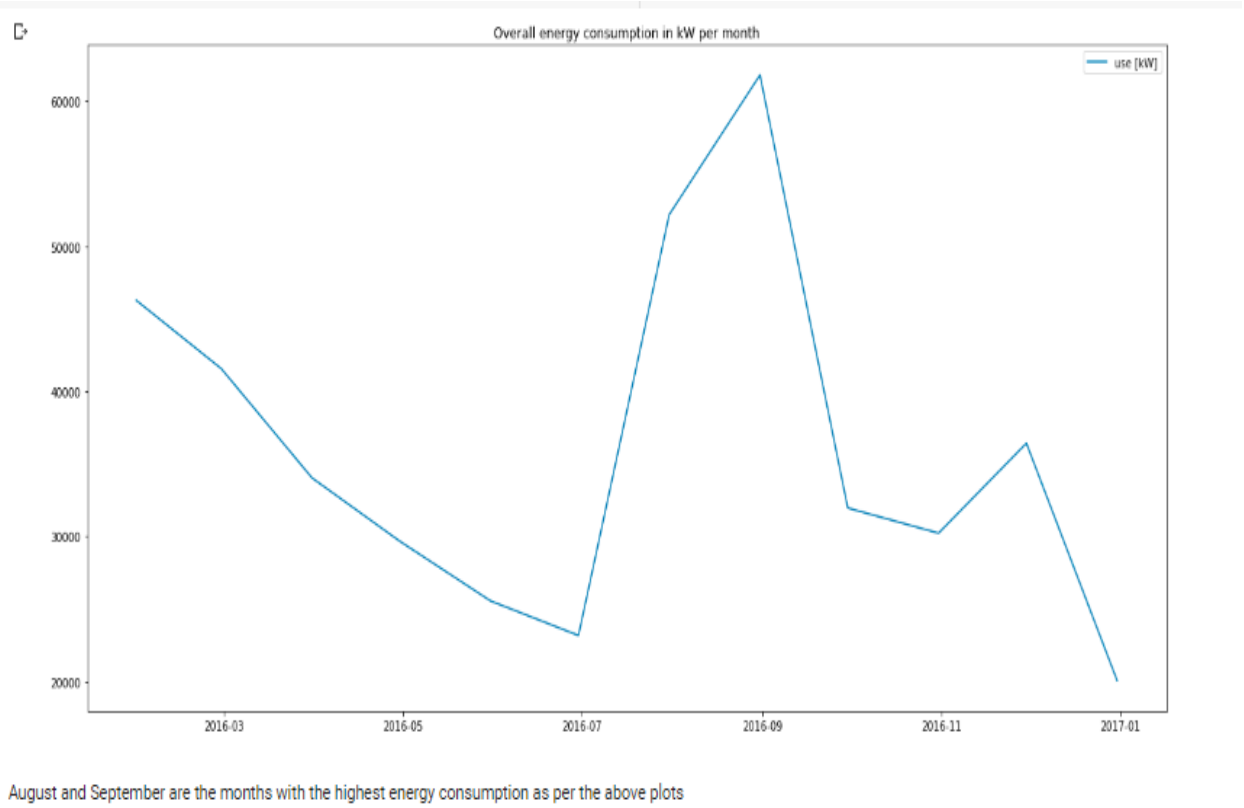
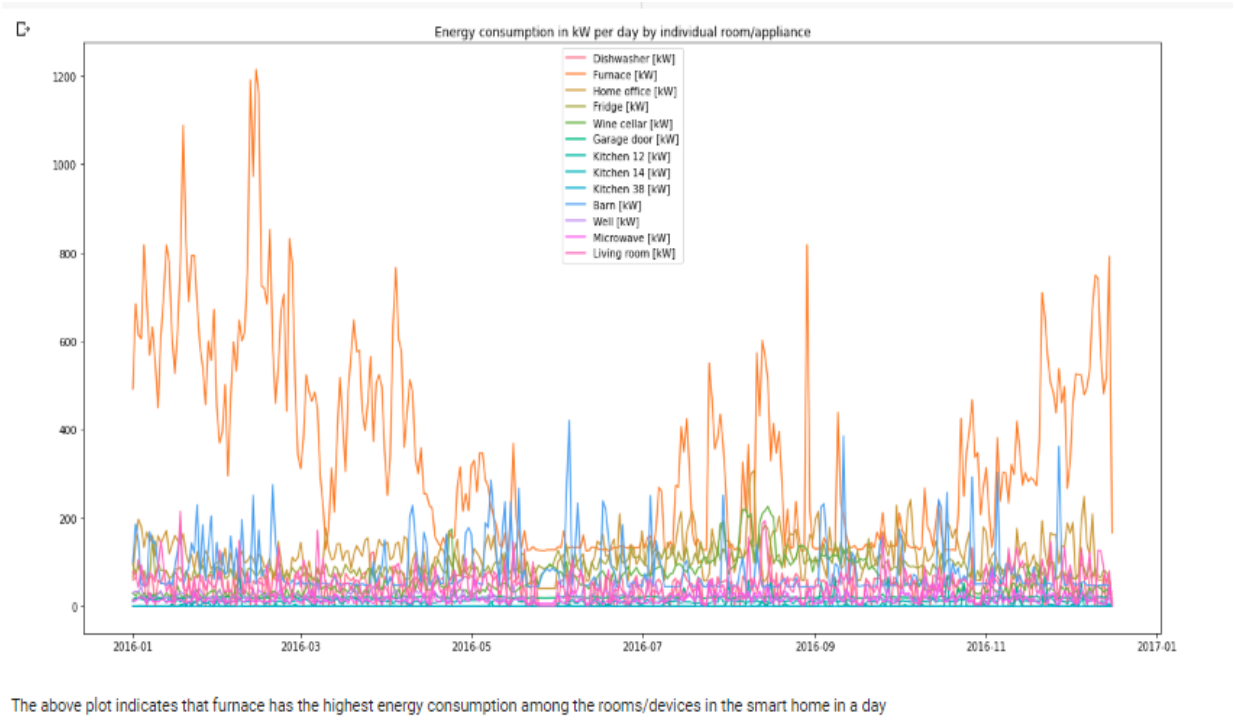
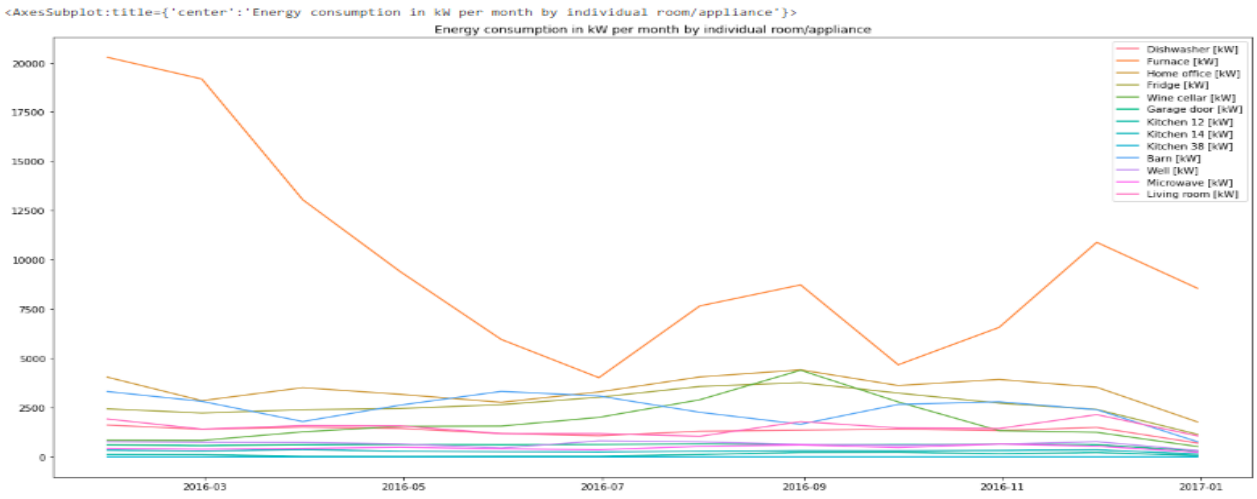


Fig 3.1.4 Overall Energy Consumption in kW per Month



The above plot indicates that furnace has the highest energy consumption among the rooms/devices in the smart home in a day

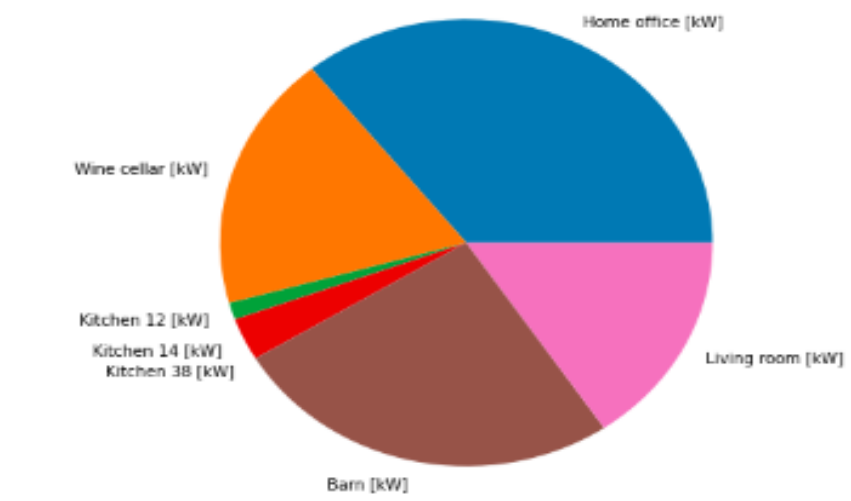
Fig 3.1.5 Energy Consumption in kW per day by individual room/appliance



The above plot indicates that furnace has the highest energy consumption among the rooms/devices and Kitchen has the lowest in the smart home in a month.

Fig 3.1.6 Energy Consumption in kW per month by individual room/appliance

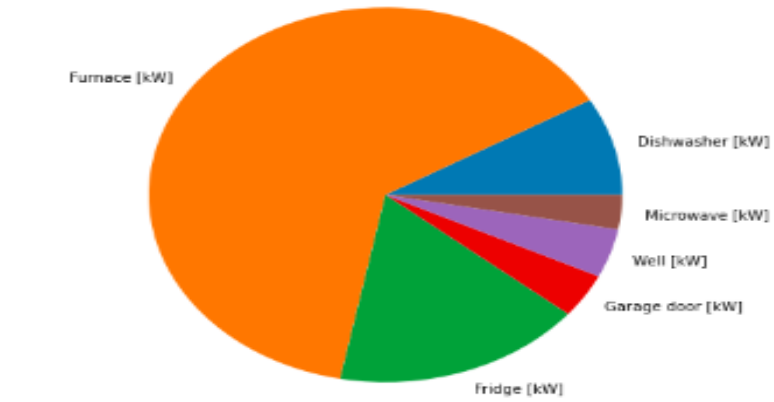
```
Text(0.5, 1.0, 'Energy consumption in kW by each room in the house')
```



Home office has the highest energy consumption and kitchen the lowest among all the rooms in the house.

Fig 3.1.7 Energy Consumption in kW by each room in the house

```
Text(0.5, 1.0, 'Energy consumption in kW by each device in the house')
```



Furnace has the highest energy consumption and Microwave has the lowest among all the devices in the house.

Fig 3.1.8 Energy Consumption in kW by each device in the house

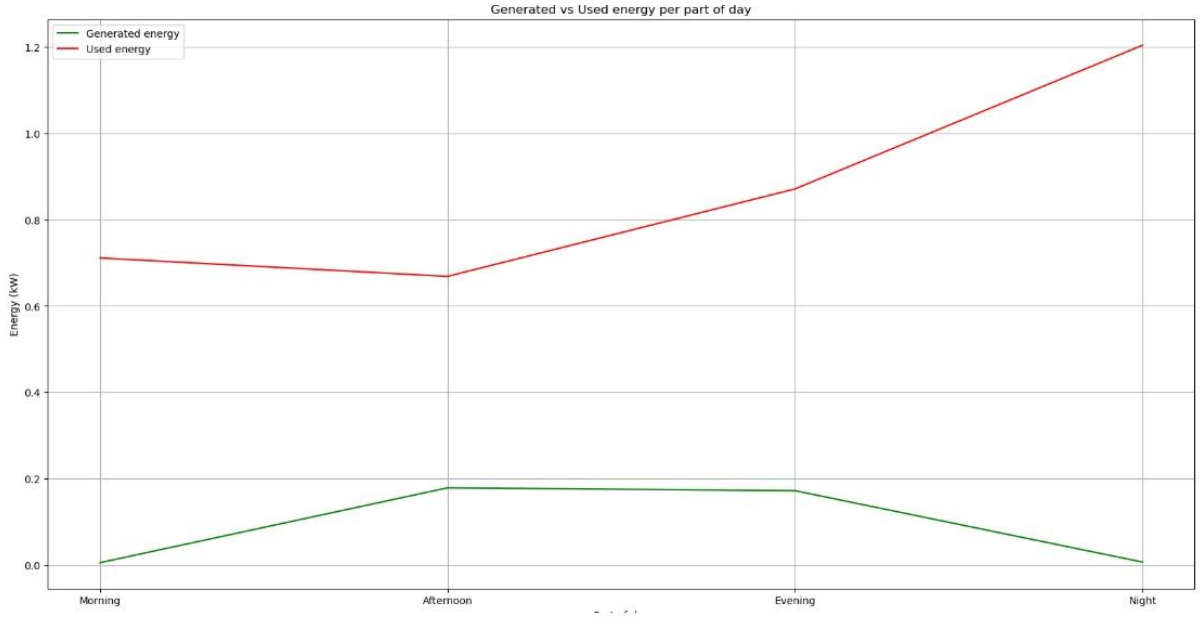


Fig 3.1.9 Generated vs Used energy per part of the day

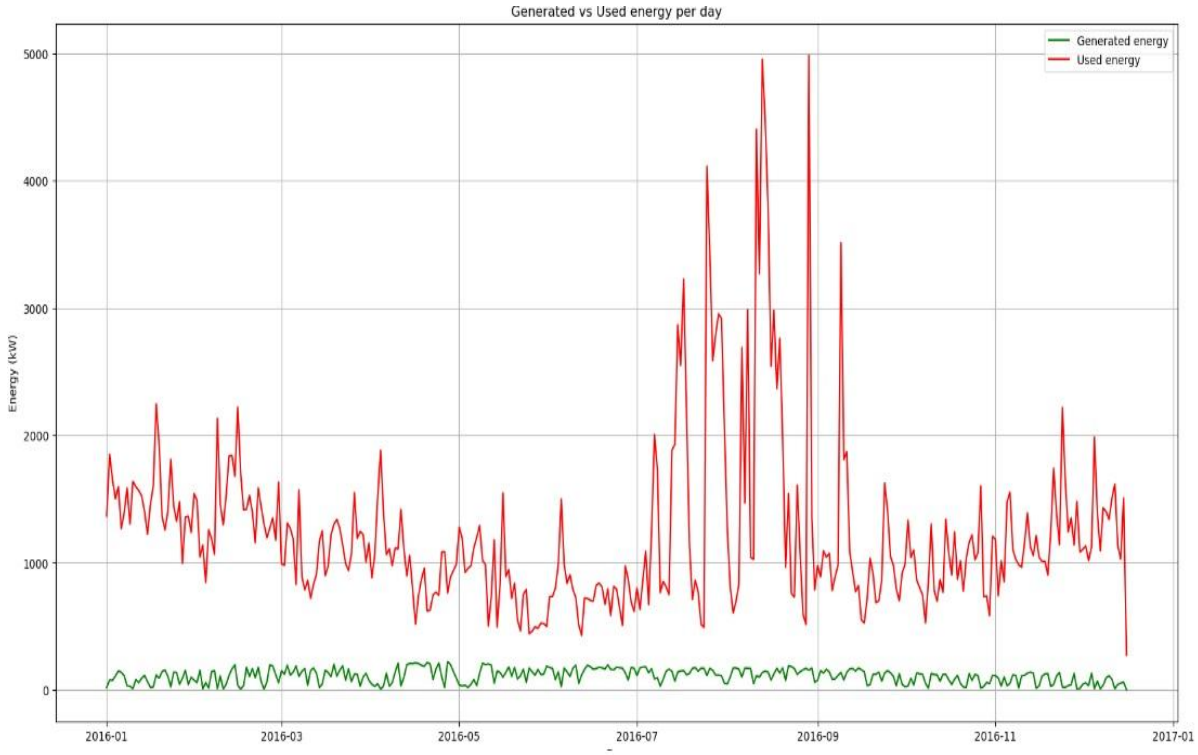


Fig 3.1.10 Generated vs Used energy per Day

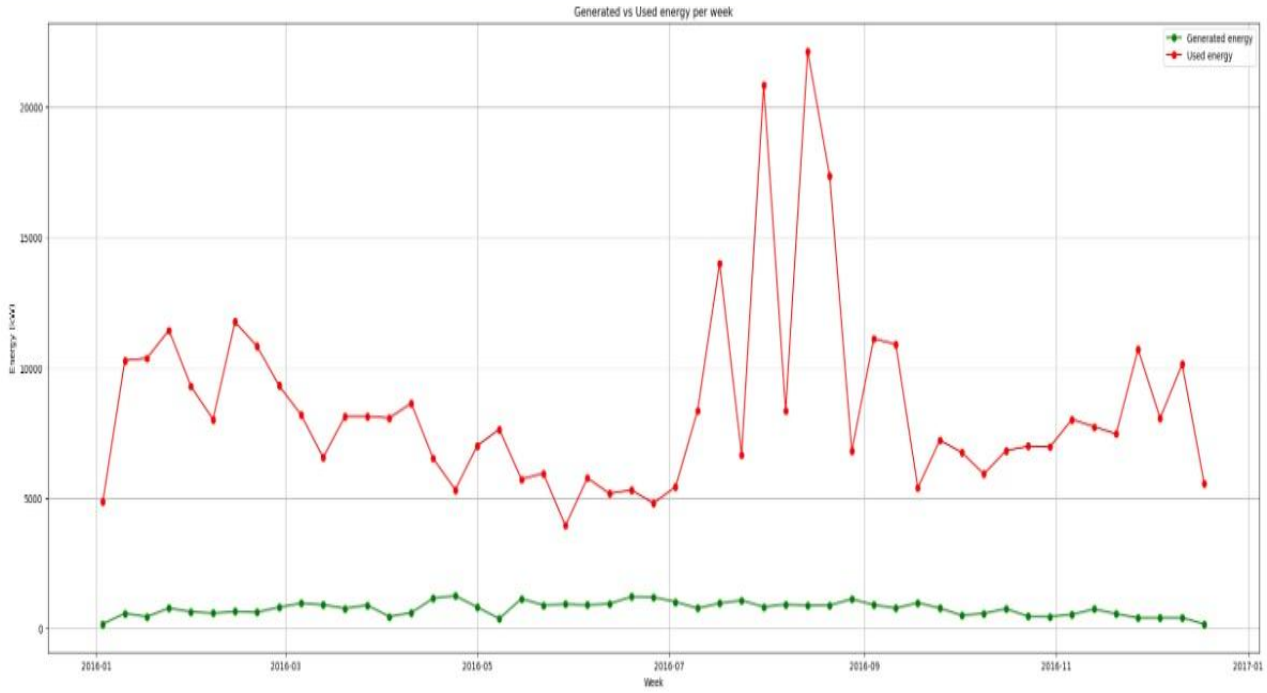


Fig 3.1.11 Generated vs Used energy per Week

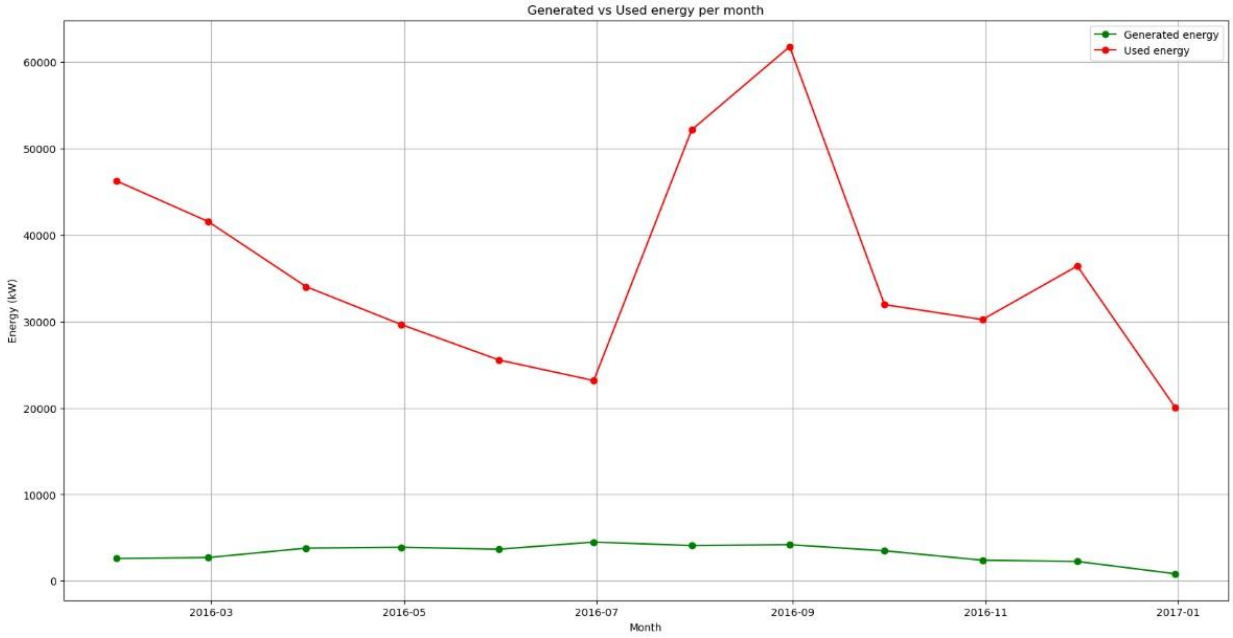


Fig 3.1.12 Generated vs Used energy per Month

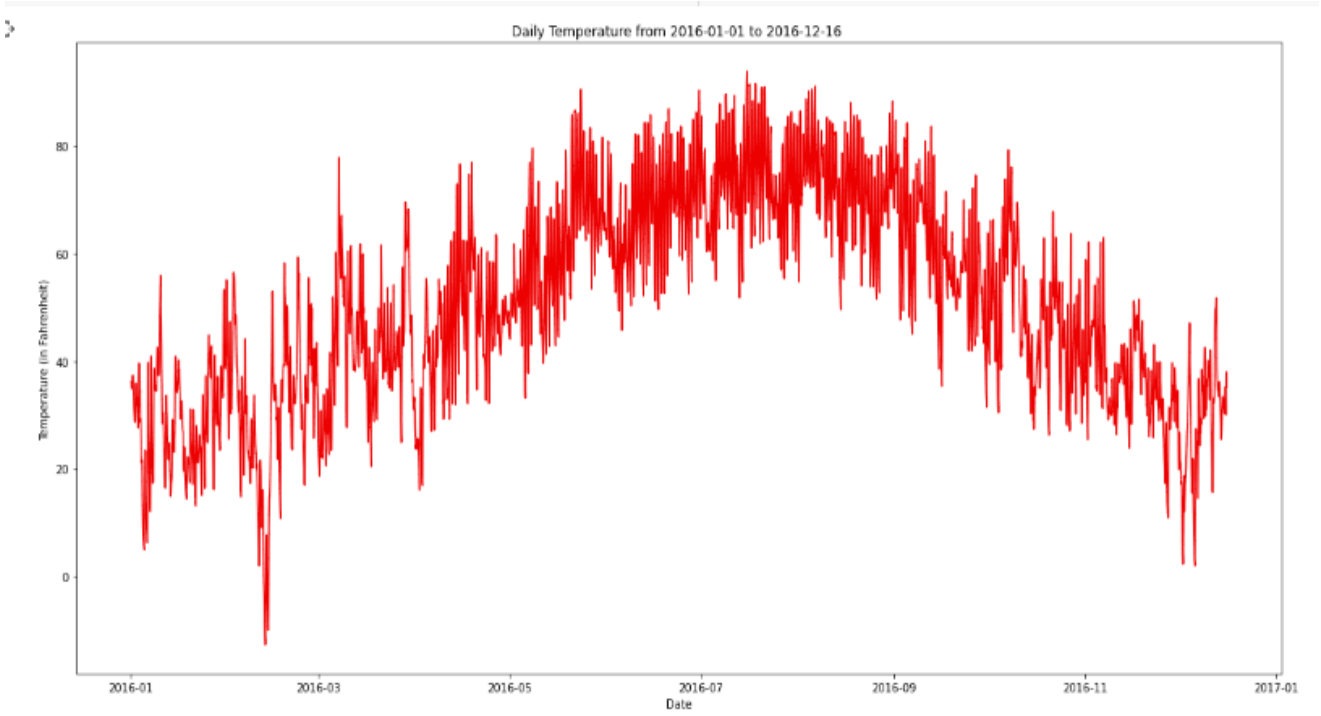


Fig 3.1.13 Daily Temperature from 2016-01-01 to 2016-12-16

From the above plots we observed that high energy consumption is in the months of August and September similar to temperature plots.

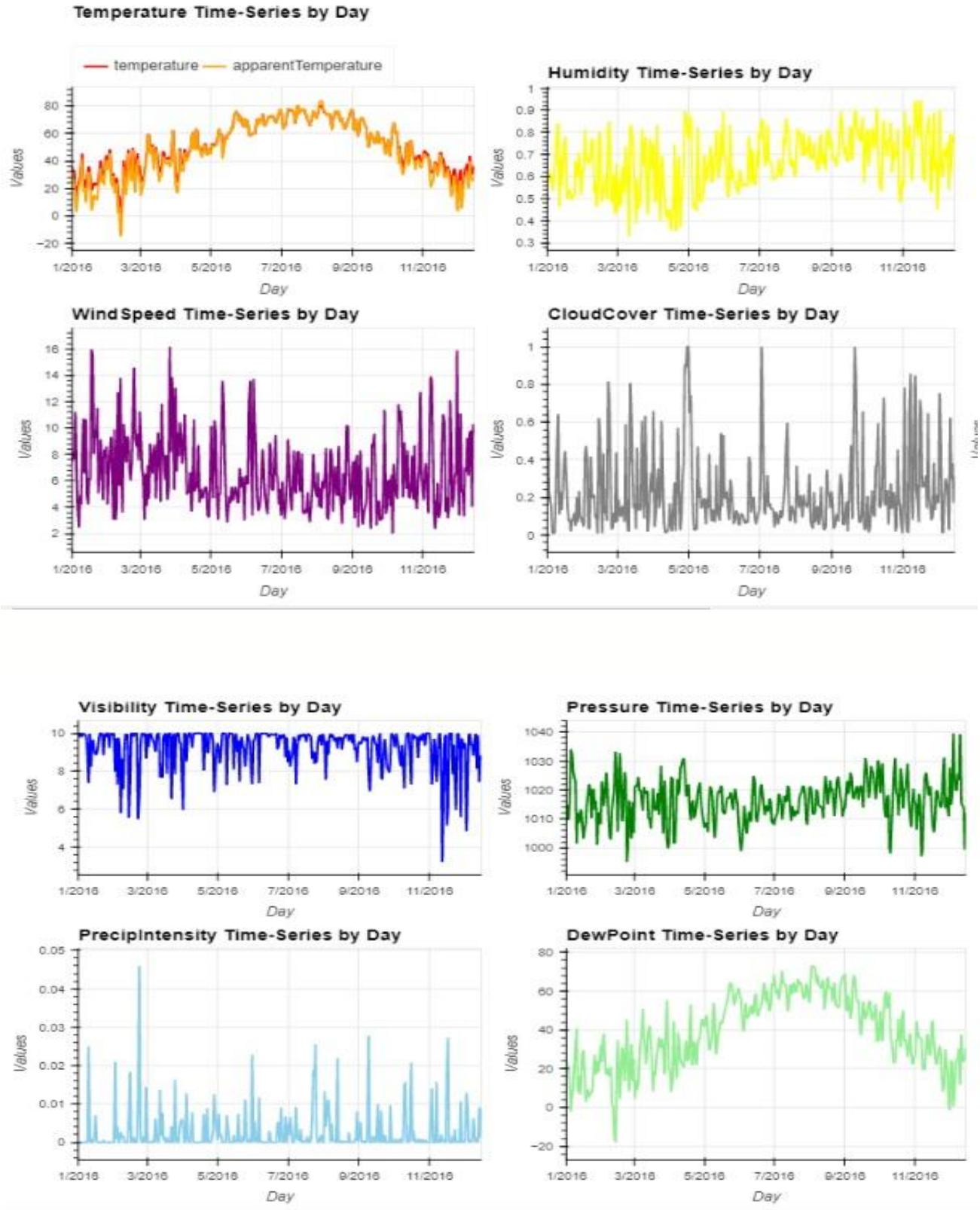


Fig 3.1.14 Weather data Time series by Day

3.2 Data Preprocessing:

An important step in the process is to make sure that the data is complete and satisfies all the requirements for data analysis. Various pre-processing techniques have been used for the same such as removing invalid rows, changing the column names into more convenient names, performing aggregation on certain columns, converting unix timestamp to proper date format, replacing missing values in columns with the next valid observation, removing unwanted columns and duplicate columns etc. The dataset was resampled to daily and monthly datasets based on the analysis requirements.

Steps involved in Data Preprocessing:

3.2.1.Data Cleaning:

Machine learning algorithms can't handle missing values and cause problems. So they need to be addressed in the first place. There are many techniques to identify and impute missing values.
If a dataset contains missing values and is loaded using pandas, then missing values get replaced with NaN(Not a Number) values. These NaN values can be identified using methods like *isna()* or *isnull()* and they can be imputed using *fillna()*. This process is known as Missing Data Imputation.
The observed row that contains null values has been dropped.
The columns 'Furnace 1 [kW]', 'Furnace 2 [kW]' were combined into an individual column 'Furnace [kW]'.

The observed invalid values in a column named 'Cloud Cover'. The invalid values were replaced with Pandas *dataframe.bfill()* to fill values. It will backward fill the values that are present in the pandas dataframe. Then the datatype was changed to float.

The Columns 'House overall [kW]', 'Solar [kW]' were the duplicates of column 'use [kW]', 'gen [kW]' respectively. Therefore the columns 'House overall [kW]', 'Solar [kW]' were dropped from the dataset.

The Unix Timestamp in column 'time' has been converted to date time format using pandas.

3.2.2.Label encoding columns with categorical data

The dataset contained multiple labels in column “icon” and “summary”. To make the data understandable or in human-readable form label encoding was used.Label Encoding refers to converting the labels into a numeric form so as to convert them into the machine-readable form.

3.2.3.Correlation

Correlations are often concerned about the relationships between two or more variables (or features) of a dataset. Each data point in the dataset is an observation, and the features are the properties or attributes of those observations.

Correlation analysis, also known as bivariate, is primarily concerned with finding out whether a relationship exists between variables and then determining the magnitude and action of that relationship.

Correlation between features:

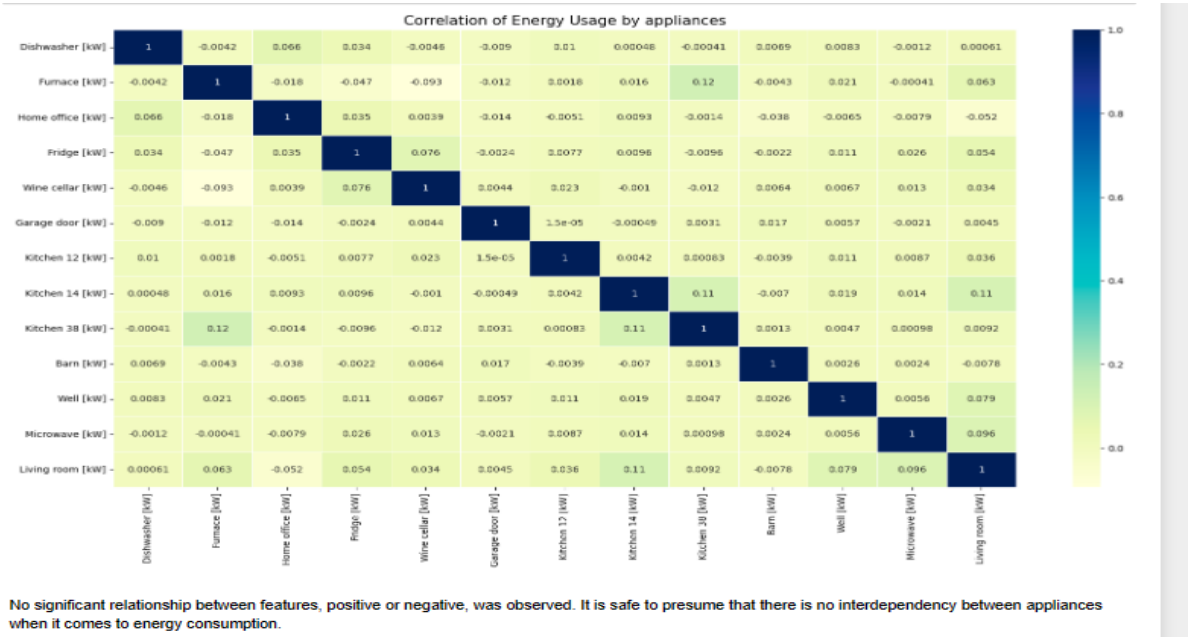
Indicates the relationship between features, values ranging from -1 to 1. There are two key components of a correlation value:

- Magnitude : The larger the magnitude (closer to 1 or -1), the stronger the correlation
- Sign : Negative indicates inverse correlation. Positive indicates regular correlation.

When the data is split into weather and devices assuming that in weather data there will be some correlation because every condition in weather depends on others for example Temperature and Humidity so the correlation can be plotted between these features.

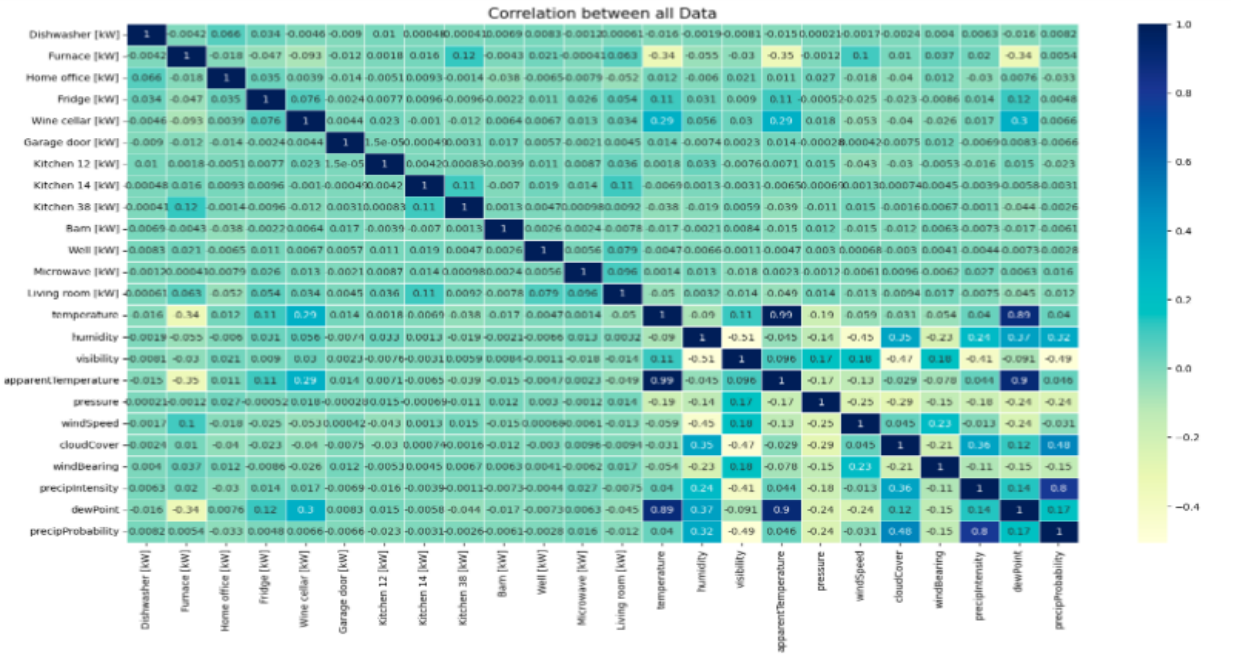
Here correlation is used to identify whether there is any kind of significant relationship between appliances, between weather data, and overall (i.e. interdependence between weather data and appliances) as follows:

- Correlation between energy data
- Correlation between weather data
- Correlation between all data



No significant relationship between features, positive or negative, was observed. It is safe to presume that there is no interdependency between appliances when it comes to energy consumption.

Fig 3.2.3.1 Correlation of Energy Usage by appliances



Weak correlation between wine cellar and weather features like dewPoint(0.3), apparentTemperature(0.29) and temperature(0.29). Relationships observed between other features as well, but not as significant.

- Some appliances are affected by weather information.
- Fridge is related to temperature, apparentTemperature and dewPoint.
- Wine cellar is related to temperature, apparentTemperature and dewPoint.
- Furnace is related to temperature, apparentTemperature, windSpeed and dewPoint.

Fig 3.2.3.2 Correlation between all Data

3.2.4.Isolation Forest

Isolation forest is a machine learning algorithm for anomaly detection. It's an unsupervised learning algorithm that identifies anomalies by isolating outliers in the data. It is based on the Decision Tree algorithm that isolates the outliers by randomly selecting a feature with the dataset and then randomly selecting a split value between max and min values of that particular feature. Using Isolation Forest not only helps us detect anomalies faster, but it also requires less memory compared to the other algorithms

Isolation Forest performs Isolation Forest algorithm to identify the outliers in generated and used energy attributes. The algorithm was applied to observe the outlier patterns for the subsets of energy_per_hour, the whole dataset. The splits for the isolation forest were performed using the max and min values of each of the features. The anomaly scores were calculated using the decision function and the anomaly attribute using the predict feature of isolation forest. The graph below shows the isolation forest plot that has the anomaly scores of the generated energy per hour and displays the regions that have outlier values in red, where 1 shows the data point is normal while -1 represents the outliers.

Scatter plot below shows the isolation forest plot that has the anomaly scores of the used energy per hour and displays the regions that have outlier values in red.

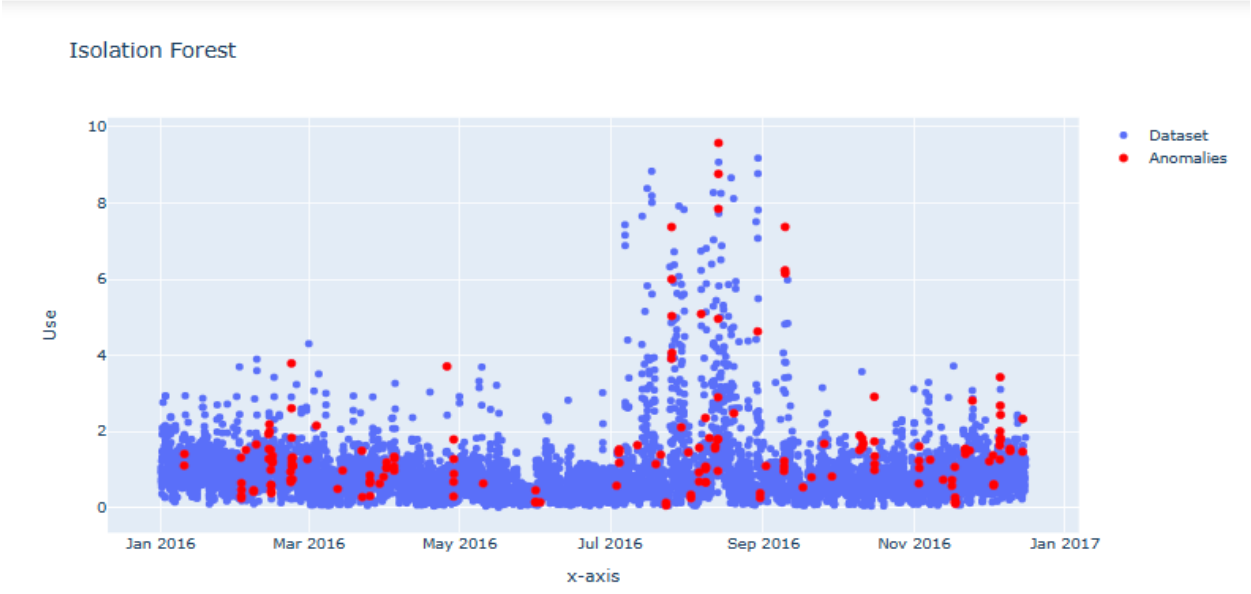


Fig 3.2.4.1 Scatter plot of Anomaly score of the used energy per hour

Scatter plot below shows the isolation forest plot that has the anomaly scores of the whole dataset and displays the regions that have outlier values in red.

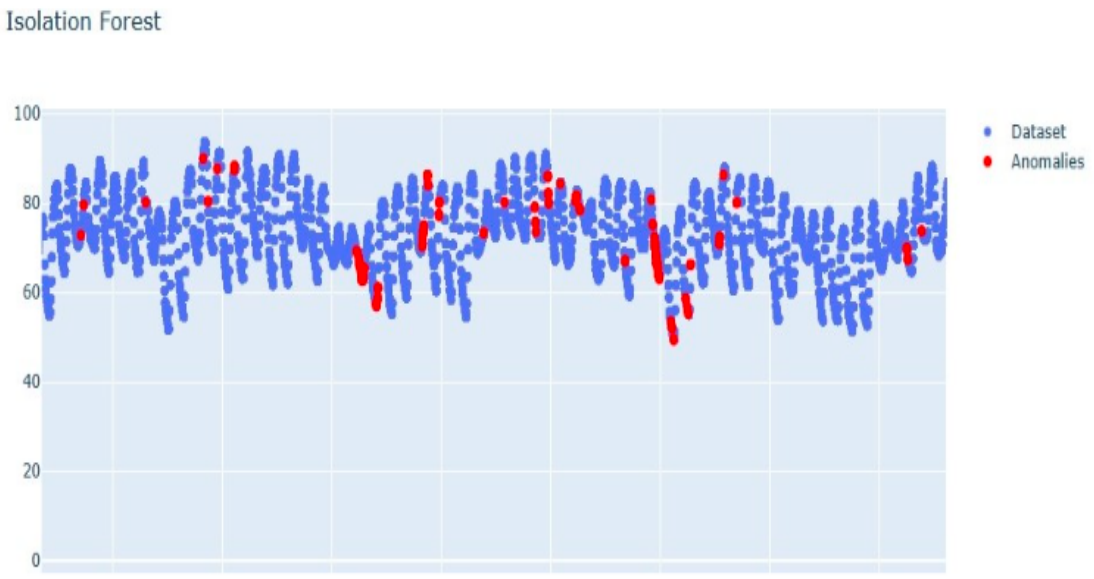


Fig 3.2.4.2 Scatter plot of Anomaly score of whole dataset

3.2.5.Machine Learning

In descriptive statistics, a time series is defined as a set of random variables ordered with respect to time. Time series are studied both to interpret a phenomenon, identifying the components of a trend, cyclicity, seasonality and to predict its future values. In this Case study we are going to concentrate on Models Like Isolation Forest & ARIMA that will help us to get some more insight from the data and also build a model to predict the future needs

3.2.5.1Arima(Auto Regressive Integrated Moving Average)

ARIMA is one of the easiest and effective algorithms for performing time series forecasting. It is a statistical analysis model that uses time series data to either better understand the data set or predict future trends. We have used ARIMA in our project to predict future overall energy consumption per hour in a smart home.

Auto-Regression:

Auto regression means that the previous values of the time series in order to predict the future. Past values used, determine the order of the AR model.

Example of an AR(1) model:

Y(t)= Some_Constant*Y(t-1)+ Another_Constant +Error(t)

Stationarity:

The repeating patterns or cycles of behavior over time.Stationarity' is one of the most important concepts that will come across when working with time series data. A stationary series is one in which the properties like mean, variance and covariance, do not vary with time.

Dickey-Fuller Test:

Augmented Dickey Fuller test (ADF Test) is a common statistical test used to test whether a given Time series is stationary or not. It is one of the most commonly used statistical test when it comes to analyzing the stationary of a series.

- The p-value is very less than the significance level of 0.05 and hence can reject the null hypothesis and take that the series is stationary which follows some trend in the data.

```
1. ADF :  -8.71681780972278
2. P-Value :  3.470341146977488e-14
3. Num Of Lags :  27
4. Num Of Observations Used For ADF Regression and Critical Values Calculation : 8371
5. Critical Values :
        1% :  -3.4311314247270106
        5% :  -2.8618853358458023
        10% :  -2.5669538174128843
```

Fig 3.2.5.1 Augmented Dickey Fuller test

P-value of given dataset is very low as shown above:

Main objective is to find the order of the AR, I, MA parts which are donated by(p,d,q) respectively. Everything is done automatically by the pmdarima library. PMDArima is an open-source Python library that is used for time series forecasting

and also helps in creating time series plots. It is easy to use and generates time-series forecasts on the ARIMA model..
The code snippet is given below:

```
: from pmdarima import auto_arima
```

```
: stepwise_fit = auto_arima(data_per_Hour["use [kW]"], trace= True,
                            suppress_warnings=True)

stepwise_fit.summary()
```

Fig 3.2.5.2 Stepwise fit summary of ARIMA

It supplies data to the auto_arima function. The function basically uses something called the Augmented Dickey Fuller score to judge how good a particular order model is. It simply tries to minimize the AIC score, and the output is given below:

```
Performing stepwise search to minimize aic
 ARIMA(2,1,2)(0,0,0)[0] intercept   : AIC=12258.704, Time=19.59 sec
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=13347.111, Time=3.40 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=13272.799, Time=0.87 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=13249.943, Time=1.80 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=13345.111, Time=0.39 sec
 ARIMA(1,1,2)(0,0,0)[0] intercept   : AIC=12254.175, Time=20.19 sec
 ARIMA(0,1,2)(0,0,0)[0] intercept   : AIC=13091.207, Time=12.09 sec
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=12263.257, Time=15.10 sec
 ARIMA(1,1,3)(0,0,0)[0] intercept   : AIC=12249.103, Time=48.84 sec
 ARIMA(0,1,3)(0,0,0)[0] intercept   : AIC=12846.888, Time=10.16 sec
 ARIMA(2,1,3)(0,0,0)[0] intercept   : AIC=12253.329, Time=56.74 sec
 ARIMA(1,1,4)(0,0,0)[0] intercept   : AIC=12238.852, Time=24.71 sec
 ARIMA(0,1,4)(0,0,0)[0] intercept   : AIC=12529.606, Time=10.32 sec
 ARIMA(2,1,4)(0,0,0)[0] intercept   : AIC=12250.568, Time=21.47 sec
 ARIMA(1,1,5)(0,0,0)[0] intercept   : AIC=12214.584, Time=34.35 sec
 ARIMA(0,1,5)(0,0,0)[0] intercept   : AIC=12292.198, Time=29.45 sec
 ARIMA(2,1,5)(0,0,0)[0] intercept   : AIC=12239.898, Time=42.34 sec
 ARIMA(1,1,5)(0,0,0)[0]             : AIC=12212.568, Time=7.08 sec
 ARIMA(0,1,5)(0,0,0)[0]             : AIC=12290.200, Time=4.06 sec
 ARIMA(1,1,4)(0,0,0)[0]             : AIC=12236.854, Time=6.90 sec
 ARIMA(2,1,5)(0,0,0)[0]             : AIC=12213.576, Time=13.87 sec
 ARIMA(0,1,4)(0,0,0)[0]             : AIC=12527.607, Time=4.57 sec
 ARIMA(2,1,4)(0,0,0)[0]             : AIC=12246.101, Time=19.75 sec

Best model:  ARIMA(1,1,5)(0,0,0)[0]
Total fit time: 408.089 seconds
```

Fig 3.2.5.3 Result of Stepwise fit summary of ARIMA

The best ARIMA model obtained seems to be of the order (1,1,5) with the minimum AIC score=408.089. This score can be finally proceeded to train and fit the model to start prediction.

Splitting the Dataset

Before actually training the model, the data is split into a training and testing section. It's because first train the model on the data and keep the testing section hidden from the model. Once the model is ready, it makes predictions on the test data and see how well it performs.

Shape of training and testing section

The ARIMA function is simply called to supply our data set and mention the order of the ARIMA model needed. The summary of the model is shown below:.

Dep. Variable:	use [kW]	**No. Observations:**	8369
Model:	ARIMA(1, 1, 5)	**Log Likelihood**	-6085.829
Date:	Mon, 13 Mar 2023	**AIC**	12185.658
Time:	21:57:33	**BIC**	12234.883
Sample:	01-01-2016	**HQIC**	12202.469
	- 12-14-2016		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.4795	0.026	18.194	0.000	0.428	0.531
ma.L1	-0.7048	0.026	-26.600	0.000	-0.757	-0.653
ma.L2	-0.0900	0.008	-10.890	0.000	-0.106	-0.074
ma.L3	-0.0430	0.010	-4.516	0.000	-0.062	-0.024
ma.L4	-0.0416	0.008	-5.056	0.000	-0.058	-0.025
ma.L5	-0.0860	0.011	-7.515	0.000	-0.108	-0.064
sigma2	0.2507	0.001	275.031	0.000	0.249	0.252

Ljung-Box (L1) (Q):	0.00	**Jarque-Bera (JB):**	448321.77
Prob(Q):	0.95	**Prob(JB):**	0.00
Heteroskedasticity (H):	1.50	**Skew:**	3.20

Fig 3.2.5.4 Summary of ARIMA

Model Summary

This has a whole lot of information about the model. Also it shows the coefficients of each AR and MA term. These are nothing but the value of the variables that is in the previous AR/MA model equation which were labeled as 'Some_Constant'. Generally a higher magnitude of this variable means that it has a larger impact on the output.

Model Prediction

To make predictions,the model is used. Predict function is given and the starting and ending index is specified where predictions are to be made.
The predictions are where the training data ends. to stop making predictions when the data set ends, the end variable is given. To make future predictions as well, change the start and end variable to the indexes accordingly. The output obtained is given below:

```
]: <AxesSubplot:>
```

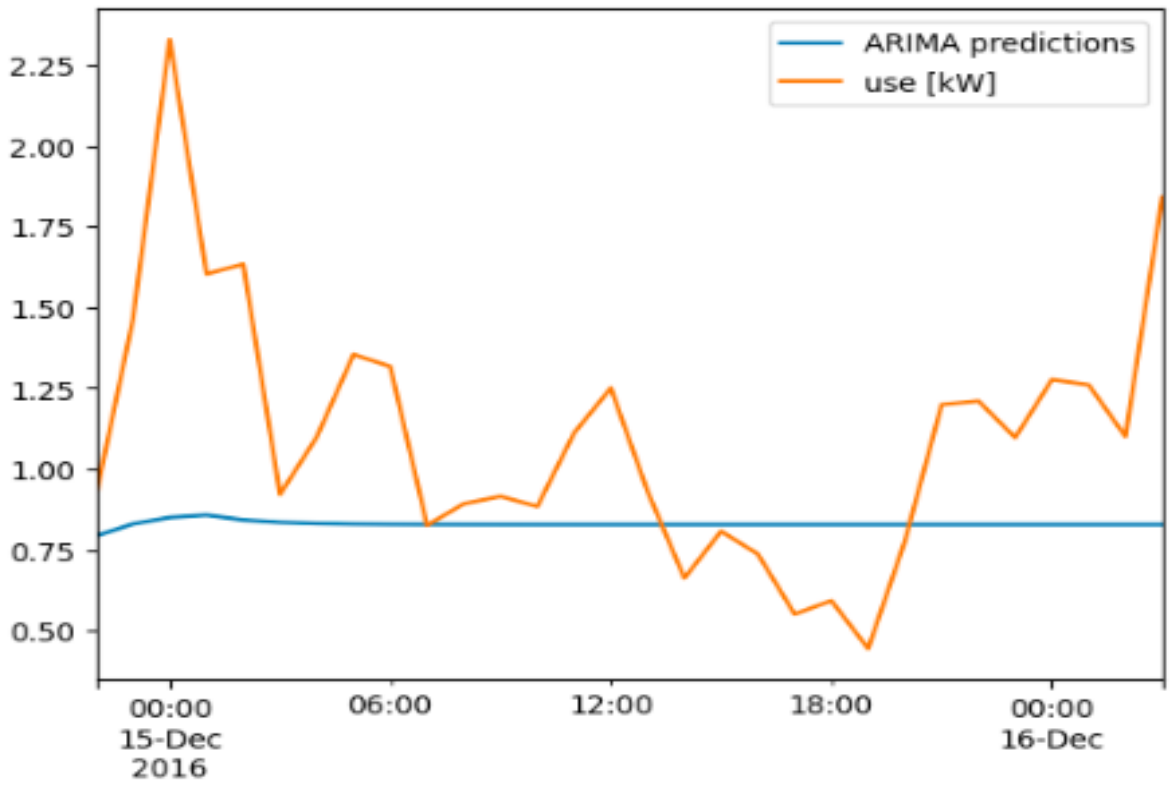


Fig 3.2.5.5 Test values vs Prediction Plot

Accuracy Metric

To check the model, root mean squared error is found. The following code snippet shows that:

```
: test['use [kW]'].mean()
: 1.1001989073933334

: from sklearn.metrics import mean_squared_error
from math import sqrt
rmse=sqrt(mean_squared_error(pred,test['use [kW]']))
print(rmse)
0.4768938989115052
```

Fig 3.2.5.6 Root Mean Squared Error

First the mean value of the data set which comes out to be 1.1. And the root mean squared error for this particular model should come to around 0.47.

The root mean squared should be very smaller than the mean value of the test set. In this case the average error is:-

0.47/1.1 *100=42.7% of the actual value.

So with that the ARIMA model is ready.

Future Forecasting

The output for future time period of 2016-12-18 to 2017-01-17 is shown below:

```
2016-12-23    1.063162
2016-12-24    1.031494
2016-12-25    1.009889
2016-12-26    0.995089
2016-12-27    0.984892
2016-12-28    0.977809
2016-12-29    0.972833
2016-12-30    0.969283
2016-12-31    0.966697
2017-01-01    0.964766
2017-01-02    0.963277
2017-01-03    0.962090
2017-01-04    0.961107
2017-01-05    0.960263
2017-01-06    0.959514
2017-01-07    0.958831
2017-01-08    0.958193
2017-01-09    0.957587
2017-01-10    0.957004
2017-01-11    0.956437
2017-01-12    0.955881
2017-01-13    0.955335
2017-01-14    0.954796
2017-01-15    0.954263
2017-01-16    0.953735
2017-01-17    0.953211
Freq: D, Name: ARIMA Predictions, dtype: float64
```

Fig 3.2.5.7 Future Forecasted Energy Usage

The prediction plot for future energy consumption for the time period of 2016-12-18 to 2017-01-17 is shown below:

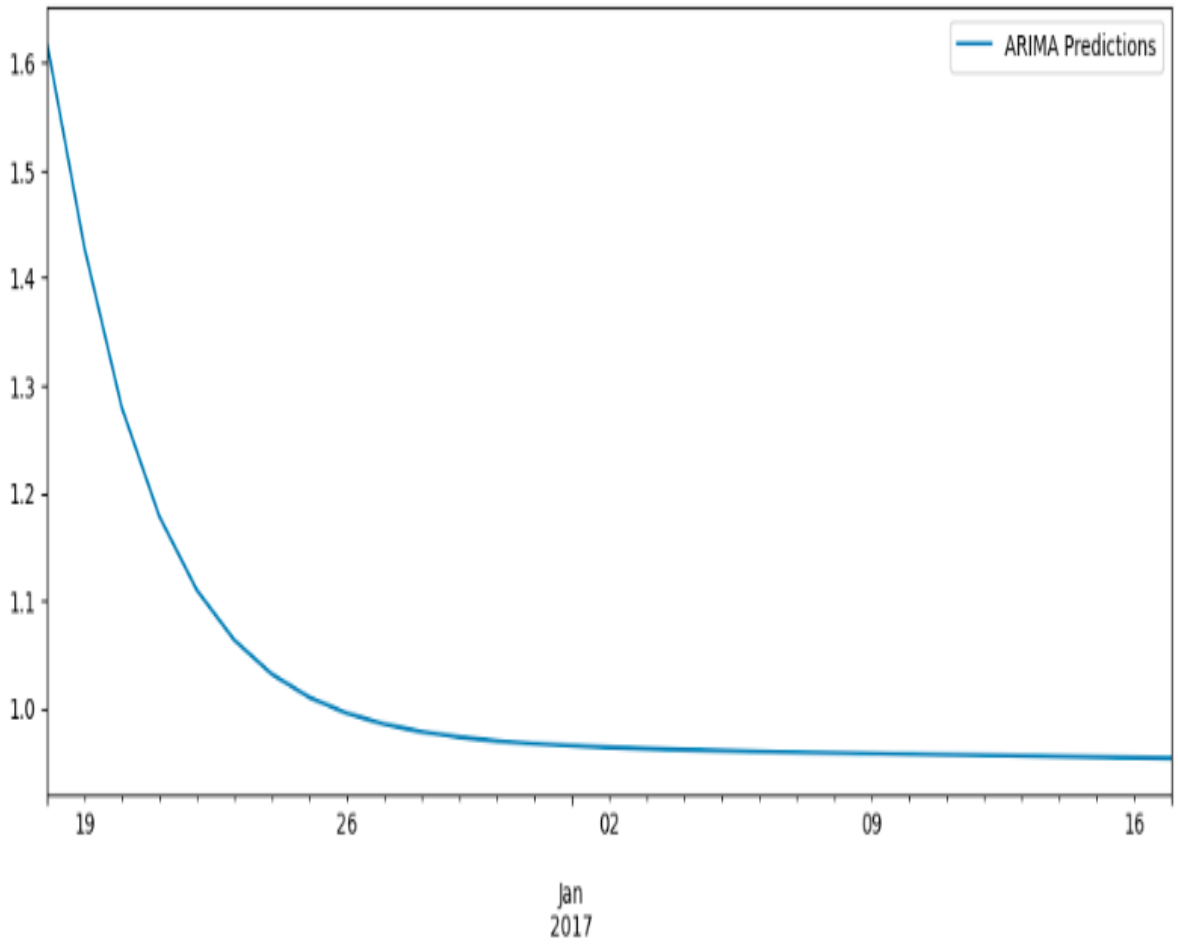


Fig 3.2.5.8 Prediction plot for Future Forecasted Energy Usage

4.Web Hosting:

A web hosting service is a type of Internet hosting service that hosts websites for clients, i.e. it offers the facilities required for them to create and maintain a site and makes it accessible on the World Wide Web. Web page is hosted using Flask and Python anywhere.

Using visual studio code is written that will take care of the server side processing. Code will receive requests. It will figure out what those requests are dealing with and what they are asking. It will also figure out what response to send to the user.It makes the process of designing a web application simpler. Flask focuses on what

the users are requesting and what sort of response to give back. Link is generated t http://127.0.0.1:5000/ in your web-browser, and the webpage (Fig 4.1) as shown below should appear. The website is then hosted in PythonAnywhere. PythonAnywhere is an online integrated development environment and web hosting service based on the Python programming language. It provides in-browser access to server-based Python and Bash command-line interfaces, along with a code editor with syntax highlighting.

Website Link : http://group6project.pythonanywhere.com/

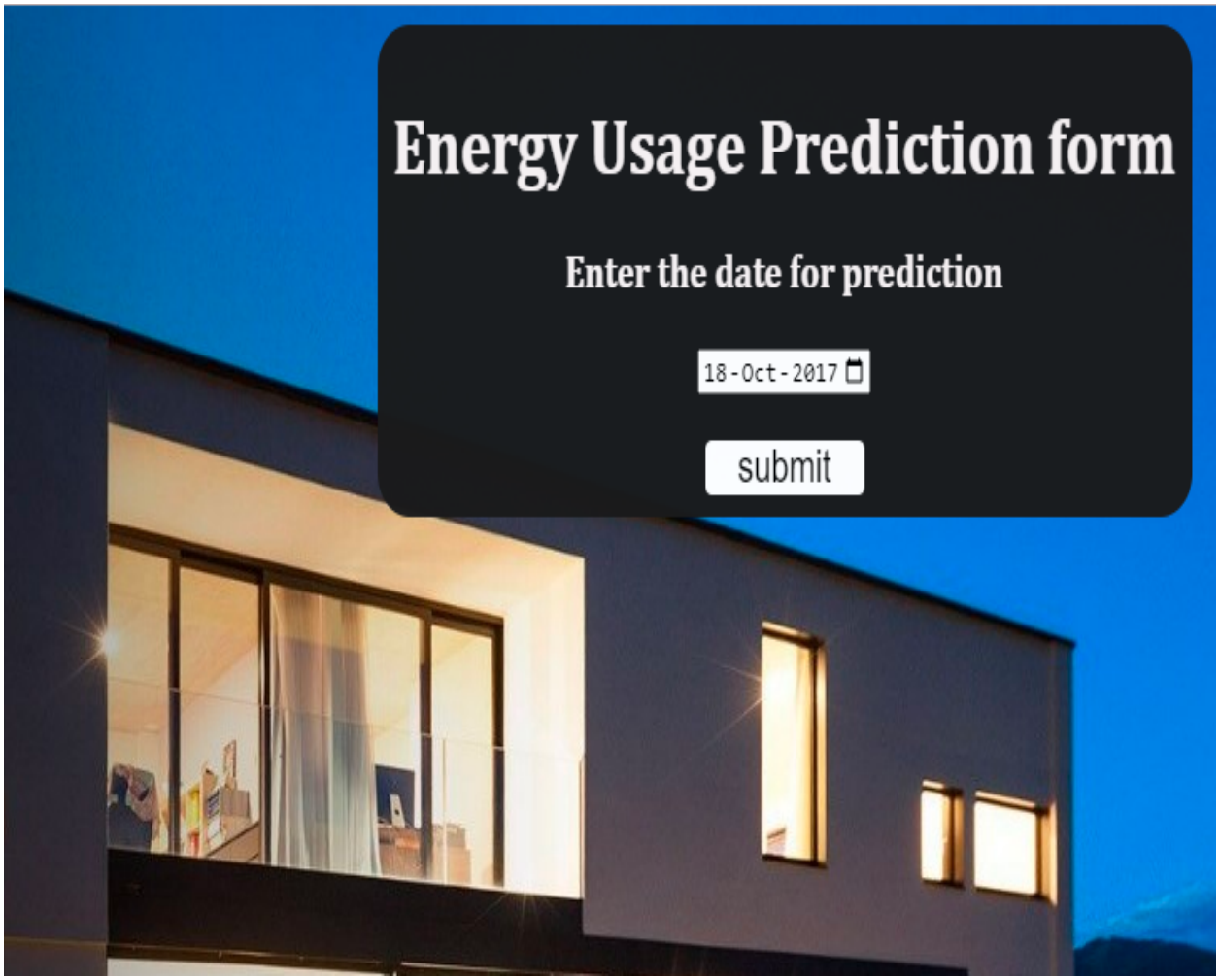


Fig 4.1 Home page of Web Application

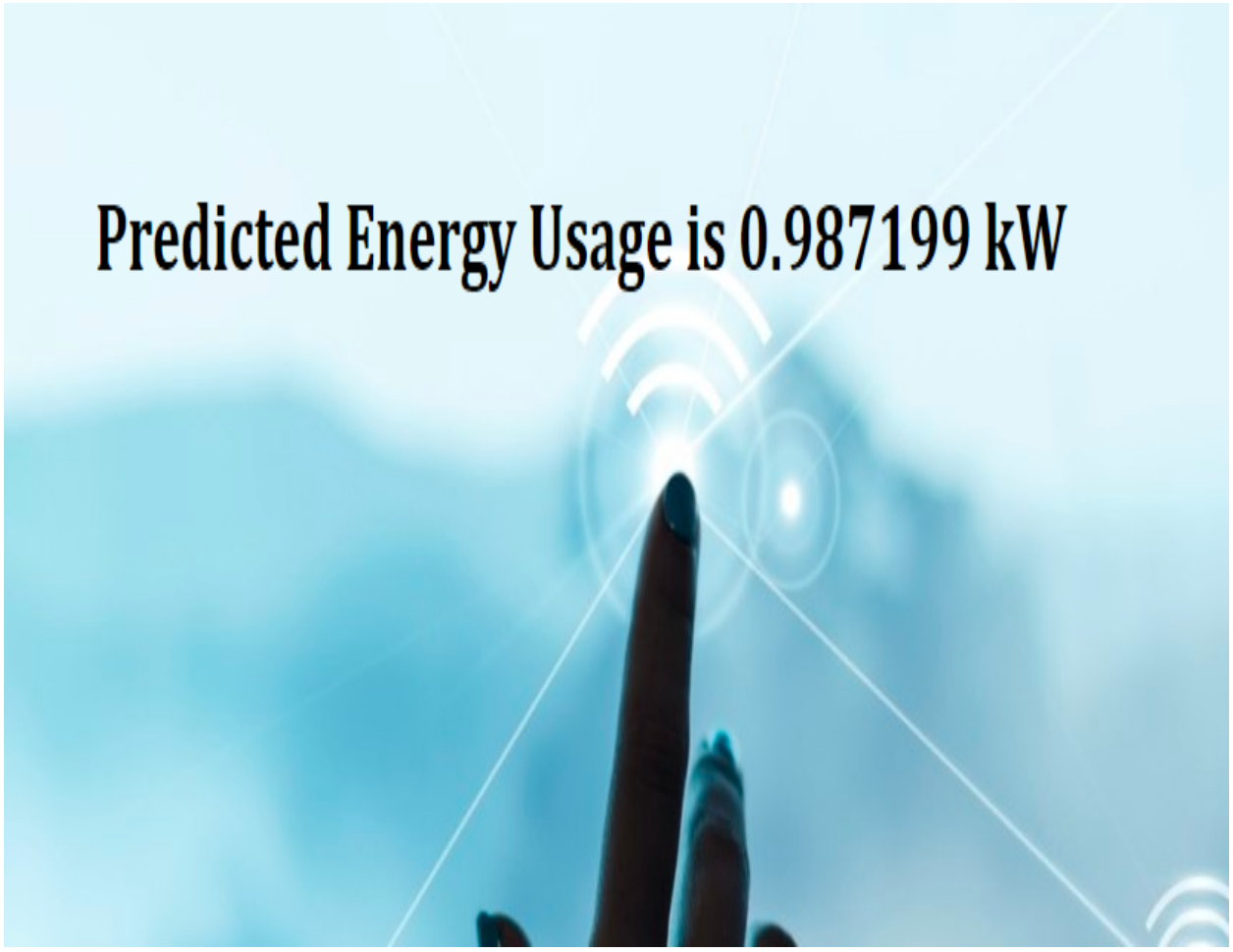


Fig 4.2 Result Page of Web Application

5. Literature Survey

5.1Dataset Used

The dataset used for this project is titled “Smart Home Dataset with Weather Information” , which has been downloaded from https://www.kaggle.com/taranvee/smart-home-dataset-with-weather-information.
The dataset contains more than 500,000 readings with the time-span of 1 minute of the energy used (in kW) by the appliances of a smart home in the year 2016, and the weather conditions of the area at that time.

5.2 Model Used

ARIMA (AUTO REGRESSIVE INTEGRATED MOVING AVERAGE) ARIMA is one of the easiest and effective algorithms for performing time series forecasting. It is a statistical analysis model that uses time series data to either better understand the data set or predict future trends. We have used ARIMA in our project to predict future overall energy consumption per day in a smart home. Isolation forest is a machine learning algorithm for anomaly detection. It's an unsupervised learning algorithm that identifies anomalies by isolating outliers in the data. It is based on the Decision Tree algorithm that isolates the outliers by randomly selecting a feature with the dataset and then randomly selecting values of data per hour by resampling. Using Isolation Forest not only helps us detect anomalies faster, but it also requires less memory compared to the other algorithms.

5.3Web Hosting

A web hosting service is a type of Internet hosting service that hosts websites for clients, i.e. it offers the facilities required for them to create and maintain a site and makes it accessible on the World Wide Web. We have hosted the web using Flask and PythonAnywhere.

6. Result

6.1 Model Evaluation & Valuation

Time series are studied both to interpret a phenomenon, identifying the components of a trend, cyclicity, seasonality and to predict its future values. In this Case study we are going to concentrate on Models Like Isolation Forest & ARIMA that will help us to get some more insight from the data and also build a model to predict the future needs. ARIMA is one of the easiest and effective algorithms for performing time series forecasting. It is a statistical analysis model that uses time series data to either better understand the data set or predict future trends. We have used ARIMA in our project to predict future overall energy consumption per hour in a smart home.

Augmented Dickey Fuller test (ADF Test) is a common statistical test used to test whether a given Time series is stationary or not. It is one of the most commonly used statistical test when it comes to analyzing the stationary of a series.The p-value is very less than the significance level of 0.05 and hence can reject the null hypothesis and take that the series is stationary which follows some trend in the data.PMDArima is an open-source Python library that is used for time series forecasting The best ARIMA model obtained seems to be of the order (1,1,5) with the minimum AIC score=408.089. This score can be finally proceeded to train and fit the model to start prediction.Before actually training the model, the data is split into a training and testing section. It's because first train the model on the data and keep the testing section hidden from the model. Once the model is ready, it makes predictions on the test data and see how well it performs.To make predictions,the model is used. Predict function is given and the starting and ending index is specified where predictions are to be made.The predictions are where the training data ends. to stop making predictions when the data set ends, the end variable is given. To make future predictions as well, change the start and end variable to the indexes accordingly.

To check the model, root mean squared error is found.First the mean value of the data set which comes out to be 1.1. And the root mean squared error for this particular model should come to around 0.47.The root mean squared should be very smaller than the mean value of the test set. In this case the average error is:-0.47/1.1 *100=42.7% of the actual value.

7. Conclusion

Smart Home Dataset records the energy consumption of each room/appliance in a house with an interval of one minute for the entire year of 2016. Initially while observing the dataset, there were several decisions we took in order to proceed with the data. After performing the pre-processing steps, we decided to use correlation and time series analysis to better comprehend the data and observe any patterns. Time-Series Analysis was used to understand trends and inconsistencies in time-related data. We have used it to detect abnormalities and forecast data. We added Anomaly Detection in order to observe any outliers in the attributes of generated energy and used energy. In order to perform anomaly detection, we used the Isolation Forest algorithm by using the data per hour values of the attributes. Lastly, time series forecasting has been done using the ARIMA model to predict the future values of total energy used by all the devices/rooms in the smart home. One of the difficulties we faced has been the run time taken for the ARIMA model while experimenting and testing with different values for the parameter values(p,d and q) and hence we decided to test on a smaller resampled dataset and came to the conclusion that the best parameters for the model are p=5, d=1 and q=0. Since the dataset only included the energy consumption for the year 2016, the predictions we made with the given dataset might not be that accurate for future forecasting in the later years. Having data that displays the values for 2017 might help us make better conclusions about energy consumption.

References

1. Very good data understanding & EDA & time-series modeling https://www.kaggle.com/malekzadeh/smart-home-data-processing-weather-vs-energy
2. Interpretability in Machine Learning https://christophm.github.io/interpretable-ml-book/
3. S. K. Sooraj, E. Sundaravel, B. Shreesh, and K. Sireesha, "IoT Smart Home Assistant for Physically Challenged and Elderly People," 2020, doi: 10.1109/ICOSEC49089.2020.9215389.
4. F. O. Chete, "Design and Simulation of IoT Network for Smart-Home," J. Electr. Eng. Electron. Control Comput. Sci., 2020.
5. M. Umair, M. A. Cheema, O. Cheema, H. Li, and H. Lu, "Impact of COVID-19 on iot adoption in healthcare, smart homes, smart buildings, smart cities, transportation and industrial IoT," Sensors. 2021, doi: 10.3390/s21113838.
6. M. A. Ashari and L. Lidyawati, "Iot Berbasis Sistem Smart Home Menggunakan Nodemcu V3," J. Kaji. Tek. Elektro, 2018.
7. P. C. Siswipraptini, R. N. Aziza, I. Sangadji, Indrianto, R. R. A. Siregar, and G. Sondakh, "IoT for smart home system," Indones. J. Electr. Eng. Comput. Sci., 2021, doi: 10.11591/ijeecs.v23.i2.pp733-739.
8. S. Kim, M. Park, S. Lee, and J. Kim, "Smart home forensics—data analysis of iot devices," Electron., 2020, doi: 10.3390/electronics9081215.
9. D. Vasicek, J. Jalowiczor, L. Sevcik, and M. Voznak, "IoT Smart Home Concept," 2018, doi: 10.1109/TELFOR.2018.8612078.
10. P. Verma and S. K. Sood, "Fog assisted-IoT enabled patient health monitoring in smart homes," IEEE Internet Things J., 2018, doi: 10.1109/JIOT.2018.2803201.
11. C. Paul, A. Ganesh, and C. Sunitha, "An overview of IoT based smart homes," 2018, doi: 10.1109/ICISC.2018.8398858.
12. F. James, "A Risk Management Framework and A Generalized Attack Automata for IoT based Smart Home Environment," 2019, doi: 10.1109/CSNet47905.2019.9108941.