Project title

Name

Submitted to

The University of Roehampton

In partial fulfilment of the requirements

for the degree of

MASTER OF SCIENCE IN COMPUTING

DATE (e.g., AUGUST 2021)

Abstract

Global warming is wide spread across the globe due to the rising population and manufacturing process. Due to this sea evels are risng causing major seashore cities being drowned in 50 years. Many countries have pledged to reduce the pollution or carbon footprint by the respective countries one of which is UK. UK has pleged to be carbon zero by 2050.

One major step towards the reduction of global warming is decreasing the reliablity on the non-renewable sources and increse the use of reneable sources of energy like Soalr, Wind, Nuclear, Hydro, etc. United Kingdom has already reduced the dependency on coal and oil for energy generation for the electricity grid.

This work aimed at helping the reduction of dependency on non-reneable sources of energy by the consumers. There are many products avaailable in market to switch to the use of solar, wind, etc but none of them are hybrid where using both the home made power and grid power while being cost efficient. I have developed an carbon and cost efficient software for the consumer.

This software used the open sources APIs which provide the electricity generated data and the agile tariff data. Software combines both the forecasting of the energy generation data and the agile tariff data to optimise and suggest ideal tme for the user which reduces the reliability of the non-renewable or carbon rich energy sources. The time series forecasting used the classical machine learning techniques like Random Forest, Extreme Gradient Boosting techniques to accurately forecast the energy generation data while reducing Mean Absolute Percentage Error (MAPE) and scrap the agile tariff forecast data from the energystats.uk website and recommedn the user with ideal time to user to the home appliances.

*Keywords: Global warming, Time series forecasting, Webscraping, Mean Absolute percentage error, Optimisation*

Declaration

I hereby certify that this report constitutes my own work, that where the language of others is used, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions, or writings of others.

I declare that this report describes the original work that has not been previously presented for the award of any other degree of any other institution.

Signed

\*\*\*Your name here\*\*\*

\*\*\*Date\*\*\*

Acknowledgements

Add any acknowledgements here. It is customary to thank your supervisor for their support.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas porttitor congue massa. Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna.

Table of Contents

[List of Figures viii](#_Toc80021021)

[Chapter 1 Introduction 1](#_Toc80021022)

[1.1 Aims and Objectives 2](#_Toc80021023)

[1.2 Breakdown of Report 3](#_Toc80021024)

[Chapter 2 Legal, Social, Ethical and Professional Issues 4](#_Toc80021025)

[Chapter 3 Literature Survey 5](#_Toc80021026)

[3.1 Motivation 5](#_Toc80021027)

[3.2 Renewable energy sources 7](#_Toc80021028)

[3.3 Utilisation of renewable energy 8](#_Toc80021029)

[3.4 Smart home technologies 9](#_Toc80021030)

[3.5 APIs 11](#_Toc80021031)

[3.6 Time series forecasting 12](#_Toc80021032)

[Chapter 4 Software development 16](#_Toc80021033)

[4.1 Data fetching 18](#_Toc80021034)

[4.2 Data processing 22](#_Toc80021035)

[4.3 Time series forecasting 24](#_Toc80021036)

[4.4 Optimisation methods 25](#_Toc80021037)

[4.5 Testing 26](#_Toc80021038)

[Chapter 5 Results 28](#_Toc80021039)

[5.1 Low carbon data results 31](#_Toc80021040)

[5.2 High carbon results 33](#_Toc80021041)

[5.3 Ratio data results 34](#_Toc80021042)

[5.4 Testing results 38](#_Toc80021043)

[Chapter 6 Conclusion and Future work 41](#_Toc80021044)

[Appendix A Project Initiation Document 46](#_Toc80021045)

[Appendix B Project Proposal Form 47](#_Toc80021046)

[Appendix C Midpoint Review Feedback 48](#_Toc80021047)

[Appendix D Project Plan 49](#_Toc80021048)

[Appendix E Supervision Meeting Records 50](#_Toc80021049)

[Appendix F 51](#_Toc80021050)

[6.1 UK Grid info 51](#_Toc80021051)

[6.2 Balancing Mechanism Reporting Service (BMRS) 51](#_Toc80021052)

[6.2.1 Transparency Data and REMIT 52](#_Toc80021053)

[6.2.2 BMRS API and Data Push 53](#_Toc80021054)

[6.2.3 Legacy BMRS Data 54](#_Toc80021055)

[6.2.4 Replacement Reserve Data 56](#_Toc80021056)

[6.3 PV\_live API 57](#_Toc80021057)

[6.4 Energystats.uk And Agile Octopus 58](#_Toc80021058)

# List of Figures

[Figure 1 Greenhouse gases emitted from various sectors [2] 1](#_Toc80018781)

[Figure 2 National Average Agile tariff variation throughout a day 6](#_Toc80018782)

[Figure 3 Sample pie chart of demand vs grid generation differentiated by various energy sources 8](#_Toc80018783)

[Figure 4 Project wireframe 17](#_Toc80018784)

[Figure 5 User persona map 17](#_Toc80018785)

[Figure 6 Daily pricing from Octopus Agile Tariff showing all regions [39] 22](#_Toc80018786)

[Figure 7 Data collected from BMRS FUELHH API 29](#_Toc80018787)

[Figure 8 Data collected from PV\_live API 29](#_Toc80018788)

[Figure 9 Segregation of low carbon and high carbon dependent energy generated 30](#_Toc80018789)

[Figure 10 Sample plot of the movement of the low carbon and high carbon energy 30](#_Toc80018790)

[Figure 11 Movement of the low carbon to high carbon ratio in 24hrs 31](#_Toc80018791)

[Figure 12 Test train split of the low carbon data for model input 31](#_Toc80018792)

[Figure 13 Naive Forecasters : Model predictions vs Actual 32](#_Toc80018793)

[Figure 14 AutoARIMA : Model predictions vs Actual 32](#_Toc80018794)

[Figure 15 Random Forest : Model predictions vs Actual 32](#_Toc80018795)

[Figure 16 XGBOOST regressor : Model predictions vs Actual 33](#_Toc80018796)

[Figure 17 Test train split for high carbon data 33](#_Toc80018797)

[Figure 18 Naive Forecaster : Model predictions vs Actual 33](#_Toc80018798)

[Figure 19 Random Forest : Model predictions vs Actual 34](#_Toc80018799)

[Figure 20 XGBOOST regressor : Model predictions vs Actual 34](#_Toc80018800)

[Figure 21 Test Train split for ratio data 34](#_Toc80018801)

[Figure 22 Naive forecaster : Model predictions vs Actual 35](#_Toc80018802)

[Figure 23 Random Forest : Model predictions vs Actual 35](#_Toc80018803)

[Figure 24 XGBoost regressor : Model predictions vs Actual 35](#_Toc80018804)

[Figure 25 Comparison of results-1 36](#_Toc80018805)

[Figure 26 Comparison of results-2 37](#_Toc80018806)

[Figure 27 Tariff details fetched from energystats.uk 37](#_Toc80018807)

[Figure 28 Program generated email sent to the user - 2 Test cases 38](#_Toc80018808)

[Figure 29 Testing use cases for various functions result 39](#_Toc80018809)

[Figure 30 Actual ratio values of energy/tariff 39](#_Toc80018810)

# Introduction

Global warming is the root cause of major climatic changes occurring around the world. As the global warming increased, it lead to increase of the surface temperature by 1 degree Celsius in the last 100 years [1]. The projections from latest IPCC report include the raise of surface temperature up to 6 degree Celsius [2]. The global warming is caused due to rapid industrialisation and progress in the last few decades [2]. This increase in surface temperatures is raising the sea levels. It is expected that the sea levels raise by 12 to 26cms by 2050 drowning most sea cities [3].

Global warming is caused by emission of the greenhouse gases mainly Carbon di-oxide, Methane and Nitrous oxide. Most greenhouse gases are emitted by the energy usage (Figure 1). Energy sector is an important sector for the developing mankind powering all the industries and homes by generating and distributing the electricity [4].

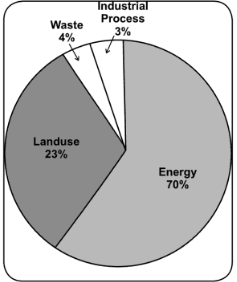


Figure 1 Greenhouse gases emitted from various sectors [2]

In the recent past, several countries come together and formed Paris agreement and pledged to reduce carbon emissions in 2015 [5]. World has to reduce emissions by 4 times in 1/3rd of the time to save earth [6]. To honour the commitment UK government has switched to renewable sources of energy such as solar, wind, hydro, nuclear, biomass, etc. UK has planned to go carbon negative by the end of 2033 [7].

The efforts on this project is put towards helping the UK goal of turning carbon negative by pushing the demand towards the renewable sources of energy. The aim and objective of the report are mentioned in the following section.

## Aims and Objectives

The aim of the project is to present the global warming situation and drive the demand of electricity from UK households towards renewable sources of energy while saving the electricity bill on the utilized energy. A tradeoff between reducing carbon footprint and reducing electricity bill burden is to be found out motivating the individuals to help the UK pledge to reduce carbon foot print.

The objective of the project is to develop an automated tool which fetches, scraps and process the energy generated data by fuel provided trusted sources, collect weather and news data, predict the energy generation for the next 24hrs using machine learning techniques and suggest an ideal time to the user for using the home appliances reducing the carbon footprint while reducing the electricity charges burden on the consumer.

For e.g. It's better to use washing machine on a sunny day not just to dry things quickly outside but because then to make use of peak solar energy capacity and reduce the carbon footprint of the grid. This tool combines live data sources for example on weather forecasting with open API’s on energy production the algorithm works out when best to plan on doing your laundry and inform your users in good time.

## Breakdown of Report

The report is broken down into multiple sections to introduce, research, develop, test and produce the results generated. They are:

1. Legal, professional, ethical and social issues are identified and documented that could be potential for this software.
2. Literature survey is done on the chosen topic by introducing the motivation, summary over the current products available and the potentiality in the field are identified
3. Software development section outlines all the development steps in terms of data fetching, pre-processing, model generation and testing
4. Results section tabulates all the results generated by the code, software and testing results
5. Conclusion and Future work summarise al the work done in project and few future potentialities that can be done as future steps are presented.

# Legal, Social, Ethical and Professional Issues

Software development comes with inherent issues of social, legal, ethical and professional issues. These issues are to be minimised to make the software right for production. Few issues were also identified in the current software developed. They are:

1. The software is an open tool leading to Digital Ownership issues since the software is directly installed in the consumer’s computer or machines
2. To use the software the users must provide personal information such as current location and choices of the user.
3. Software required constant access to the internet to download, process and suggest the users in good time.
4. The software depends on machine learning which leads to incorrect predictions if the date provided to the machine learning model is biased or erroneous
5. The software uses the open sourced API completely for the data which can be inaccessible or moved to paid version which is a roadblock for the software
6. Any technical issues with the software are accountable by only one developer involved in the design and development
7. The software doesn’t have inbuilt cyber security support to protect against the cyber crimes
8. Large scale deployment of the software makes it obsolete since the demand in the suggested timings increase therefore increasing the dependency on non-renewable or interconnectors to meet the demand

# Literature Survey

## Motivation

We depend on carbon sources, mostly carbon for electricity generation. Coal is the highest used source for energy generation in UK in 2016. It accounts for the 42% energy generated in 2016. It is a non-renewable energy source, and prices are also rising [8]. Although the dependency on these carbon rich fuels has reduced from 1990, this is not enough to curb the alarming rate of rising global temperatures [9]. It would be more efficient to reduce carbon dioxide emissions by switching to renewable energy sources such as Solar, wind, water, etc. The grid consists of both renewable and non-renewable energy sources. Depending on the supply and demand on the contour, the amount of Coal raised in solar power capacity changes [9]. Goal is to use this data efficiently from the energy side of our devices. Detailed information about solar generation is already available on various websites via the APIs for receiving and analysing data [10]. This data was collected and used to find the optimal time for such a "smart home" device as a washing machine or other appliances. On a warm sunny day, and the amount of energy generated by solar power is too large. Thus, it would be good to use the washing machine in the room.

In the recent years, the environmentalists raised caution against burning fossil fuels causing global warming which is raising earth surface temperature by few degrees and sea levels by few cms which could drown few major cities across the world such as Miami, Shanghai, Osaka and other cities by 2100 [11]. This concern has let to global adaptation and pledge to reduce global warming. In this interest the grid has adopted various renewable sources of energy such as wind, solar, geothermal which have least carbon footprints. UK national grid expects to be least reliable on fossil fuels by the year 2030 by switching to alternate sources safe for environment [7].

Major source of electricity across the UK comes from the national grid. National grid is a central platform that collects and distributes electricity to homes and hubs [9]. The grid collects electricity from various sources such as coal, nuclear, wind, solar, geothermal, etc. The UK grid also have interconnectors from Norway, France, Denmark, Netherlands, etc to overcome the deficiency of the electricity supply during high demand days [12]. Over the day the amount of energy generated from each unit varies. The combination of all the sources of energy is adjusted for demand across the houses at different periods in a day [13]. The energy from the national grid is distributed to 14 Distribution Network Operators (DNO) who are responsible to transmit and distribute to local distribution via towers and cables [14]. The agile tariffs are set by DNO and collected from the households. All the data related generation, distribution and tariffs are available by Balancing Mechanism Reporting Service (BMRS) via open APIs [15]. Figure 2 depicts the agile prices set by one of the DNO.

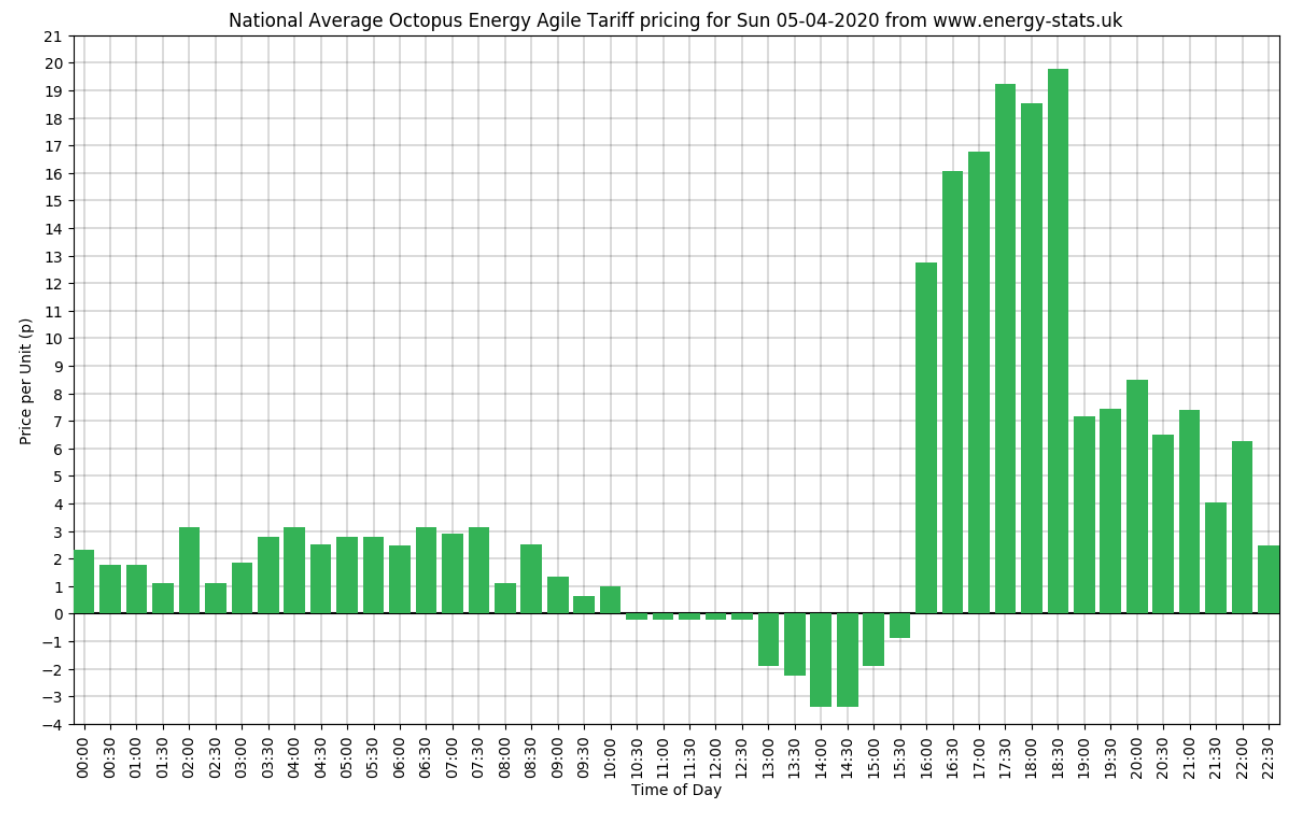


Figure 2 National Average Agile tariff variation throughout a day

Application Program Interface (API) allows multiple applications to communicate between each other. They communicate via request and response method. Modern day APIs adhere to HHTP and REST which can be treated as products to be used in the Software Development Lifecycle (SDLC) [16]. APIs add layer of security for transferring data since the data is only transmitted in smaller packets only when necessary. This reduces the risk of the cybercrimes while increasing security layers [17].

Solar-API offers an approach that focuses on using all the energy output and resources in the network [8], and a cheap intelligent home protection system, for security and video surveillance in the house, is powered by green energy [18].

## Renewable energy sources

However, so far not much work has come out in the field of renewable energy, predictions and should be used for smart home devices. To describe the current market, the generation is migrating towards renewable energy from conventional coal-based electricity generation. In many developed countries, the grid that provides electricity to millions of homes are dependent both on the renewable and non-renewable sources of energy. A webtool [19] based on the grid information displays the amount of energy generated from various energy sources at given point in time. The figure below shows the time chart of the energy demand vs production from various sources of energy in UK grid at 6:00 BST

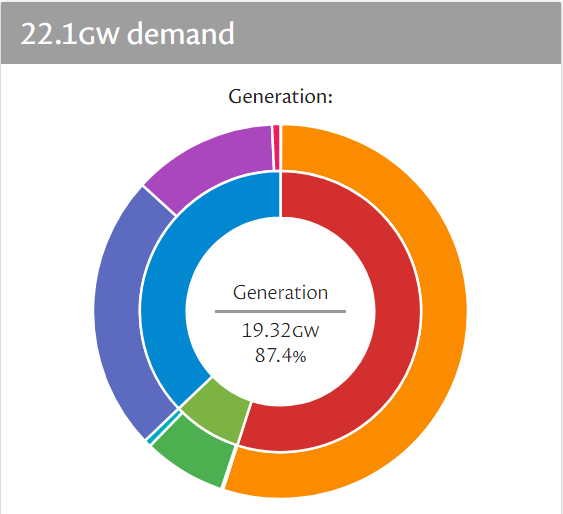


Figure 3 Sample pie chart of demand vs grid generation differentiated by various energy sources

The amount of energy generated from various sources of energy varies during the day varies with the amount of the natural sources available such is solar energy, wind energy, geothermal energy, etc. These forms of renewable sources are available only during time of the day or year such as solar energy is available during the day light [18]. The renewable sources cost less than the conventional coal-based sources [19]. The grid also charges accordingly to the energy generation from these sources. These challenges make it difficult to completely rely on the renewable sources of energy for grid.

## Utilisation of renewable energy

To make use of these renewable energy sources and make them cost-effective methods many households around UK and USA are using roof panels on their respective homes and using solar energy [20]. This only has a capital of installation charges where the solar panels price has also decreased in the recent years and efficiency of the panels has also increased thanks to the efficient manufacturing methods and mass production of these in countries like China [21]. Companies have produced inverters and storage batteries to save this energy generated by the solar roof panels. Inverters automatically switch from grid to the solar panel energy depending on the availability of the solar energy captured.

## Smart home technologies

There are few companies/tools/apps which facilitate the use of the appliances to reduce the power bill and carbon footprint. We will explore the possibilities to capture the amount of solar energy from the photovoltaic cells spread over the farms. In this project we will try to build an app that notifies the user when to use the high-end appliances at home to save both money and environment. The app is based on the APIs available online which helps us fetch data. There are devices available for mass public to fix in their roof top homes to monitor the generation of the solar power and switch automatically from grid to the solar power if it generated enough power to run the house. They are broadly classified into 3-types:

**Solar-powered solar systems** [22]- these systems use an inverter to change the low DC voltage generated by photovoltaic modules into alternating current, the voltage, and frequencies that are required for the power supply can be 110 V or 230 V, 50 Hz or 60 Hz depending on the country in which you live. Power is therefore reversed at a higher voltage than the mains, so that energy flows online as well as its own electrical charge. If the current is turned off, the inverter will turn off to protect people who are working on the power lines, which means that there is an electric charge on the power supply.

**Off-Grid Solar Systems** [23]- These systems as the name suggests are not connected to the grid, instead they are built with large battery banks, and store the electricity generated by PV panels in these battery banks. These systems use electrical inverters to convert DC power from PV panels into AC power to grid power and the amount of land you live in. These systems are designed to power your cargo power from the PV system automatically, but only when one can power the power switch to connect the cargo to the power grid when the battery banks are removed, and the PV panels also fail to generate power for any reason.

**Hybrid Solar Systems** [23]- These systems have been designed to provide the best of both systems described above. They are tied to the grid", meaning that they are not only able to push excess electricity back into the grid via a bidirectional electric meter, and at the same time, they will also be supplied with a battery pack. These systems can be programmed to control the load power from the default photovoltaic power output, and then use the excess power inside the capacity on the unit. And then every transition to the power supply can be transmitted to the network. And if the grid fails, solar panels and batteries feed the load, and photovoltaic cells and accumulators are not or do not exhaust, the electrical load will be fed from the grid. Such hybrid systems are much more expensive than the other two types mentioned above.

Existing smart home devices provided by various companies have enhanced the ability for consumers to watch their electricity spending day-by-day and appliance-by-appliance. These smart meters help the consumers to make ideal choices and change the patterns of consumption to reduce the electricity bills [24]. Another company named Agile Octopus has taken the battle step ahead to provide consumers with agile pricing and notify the users plunge pricings to save electricity bills and Octopus Go, tariff designed for EV owners offering super cheap night rates [25].

While there are several products available in the market none of them give an advantage to the customers by reduce carbon footprint while saving the electricity bills. This project focuses on the aspect to develop a software which make uses live/historic data available on the open APIs regarding the grid energy and recommend the user with appropriate timings to use the home appliances which saves both nature and money to the consumer.

## APIs

Technology has become a common sight in today’s modern era. It has been difficult to foresee any business from micro to macro without use of software. Application Programming Interfaces (API) have become of such innovation of technology which has enhanced the human’s ability to provide a micro service which can be used to transfer data efficiently. APIs are existing to help two or more machines communicate with each other. They provide communication channel for a program between the front-end and back-end [26].

APIs can be build using several platforms which enables the communication or data transfer in software. For e.g. a website communicates with the database using an API. Single API can enable communication with several devices or software for time efficient use of the software. These days the posits are pre-implemented and open sourced for several online to make use of these data [17]. These such APIs from BMRS, Sol cast and Shielfield Solar are few open APIs used in this project

## Time series forecasting

Decision Tree is one of the supervised learning algorithms types. The target variable has already been defined in supervised learning algorithms [27]. A decision tree is a predictive model, and it is also known as reduction tree or classification tree. This is mostly used in problems of classification. It operates for definite as well as continuous variables of output and input. In this method, the sample is composed of two or more homogeneous sets. Every internal node in the decision tree reflects the "test" on an attribute (i.e. false or true).

Three types of nodes are there in a decision tree, Decision nodes – It is represented as squares usually, Chance nodes – It is represented as circles, End nodes – It is represented as triangles. For making predictions, the decision tree machine learning methods are used. They are constructed by dividing training data into progressively smaller samples. In machine learning, the decision trees are the most commonly used classifier. Based on the feature value tree-structured decision tree classify instance by sorting them. A node inside a decision tree shows the selected feature, it is used to divide input information, and branches indicate the node values. In the last couple of years, C4.5 is became the standard tree decision method [28].

Random forest [29] has evolved from decision trees (DT) as a regressor. In fact, it is composed of numerous DT’s. To distinguish a new instance, every DT offers the input data regression. Random forest will gather the regressions and, as a consequence, selects the most voted forecast. Each tree's input is selected from the initial dataset. Moreover, from the optional features, a features subset is randomly chosen to develop the tree at every node. Without pruning, each and every tree is grown. Ultimately, random forests make a robust regressor, possible for a significant

amount of weak or weakly-correlated regressors.

The operator of the random forest is accessible in Modelling then Classification and Regression then Tree Induction and then Random Forest. The mechanism is identical to the other ensemble models that require the user to specify the number of base trees to be built. Meanwhile, the model of the internal base will always be a DT, There's no specification of an explicit internal subprocess. Explicit internal subprocess specification is required for boosting or bagging ensemble models [30]. All variables for the tree, such as the size of the leaf, the depth and the Random Forest Operator may specify the split standard. The number of trees is the main parameter specifying the number of base trees.

Random forest fits many classified trees into a data set and then connects all the trees predictions. The algorithm starts with several bootstrap samples selected from the data. Larger sample values lead to the stability of classifications and measurements of variable significance. Observations which does not occur in a sample of bootstrap in the original data set are defined as out of-bag observations. A bootstrap sample would be the classification tree, but only a small number of randomly selected variables are available for binary partitioning at each node. To ensure that the fitted classification trees in the random forest have smaller pairwise correlations, the lower value of randomly selected variables for classification is taken. The trees are fully grown, and each of them is used to predict the out of-bag findings. The expected class of an observation, which ties divided randomly, is estimated by majority vote of the out-of-bag predictions from the observation. Using the out of-bag projections, accuracies and error rates are worked out for each observation and then averaged over all observations [31].

Since the renewable sources especially solar panels have become mainstream in the market many forecasting models have been commercially deployed for solar panel to forecast the amount of energy is generated using Photo Voltaic (PV) cells. In 2010, forecasting model adaptive Nonlinear Autoregressive model with Exogenous

Inputs (NARX) network [32] is proposed which uses the historic solar energy generation data and the weather data such as clouds data to train and forecast the solar energy that can be produced depending on the panels size and location. The NARX model is a time series deep learning model with three layers (input, middle, output) of neurons and training has Mean Absolute Percentage Error (MAPE) of ~17%.

In 2014, paper published proposing a hybrid solar energy forecasting technique [33] which predicts short term photovoltaic power generation depending on the weather patterns. The model with clear sky predicts depending on the cutting plane instantaneous intensity coefficient calculation while during non-clear sky the energy is forecasted by quadratic curve calculations measuring surface radiations and the time. The MAPE for the is ~10%

In 2016, paper [34] proposed Back Propagation Neural network (BPNN) and linear regression technique for forecast wind speeds to aid the energy production calculation from windmills. The Neural network model proposed had MAPE of 26.33% while the linear regression model has MAPE of ~40% which are decent models with lot of room for upgradation.

Since the individual energy by fuel types follow particular patterns during day and time, it is comparatively easy to predict solar energy forecasting rather than total energy forecasting as the variability of the all the fuel types is different adding complexity to the final forecasting. I’m proposing a novel random forest and xgboost based forecasting model having higher accuracies in predicting the ratio of low carbon to high carbon.

The detailed project plan and design are detailed in the following section.

# Software development

In the previous section the overall survey of the existing infrastructure and the opportunity available in the market are detailed. In this section I have detailed the project plan and the technical elements required in the project are highlighted. To summarise this section, have given brief idea of collecting data from open source APIs, ethical web scraping, time series prediction methods and optimisation methodology followed for the project.

The user story that defines the work to be done in the project:

As an Engineer, research and develop a tool useful to automatically the alert the user timings for the usage of washing machine while reducing carbon footprint and electricity bill

Description as a Gherkin Story:

**Agenda***: Develop tool to automatically alert the user to use washing machine*

**Scenario***: User wants to know the ideal time to use the washing machine at home*  
**Given***User’s home is connected to the grid which dynamically adjusts between the various sources of energy*   
**And***the information about the energy sources by percentage and the flexible tariff details are available*  
**When***user checks his mobile/email for the notification from the app*  
**Then***according to the alerts raised by the tool the user can use the washing machine or other high voltage appliances to save environment and cost*

The skeleton wireframe of the workflow is as follows:

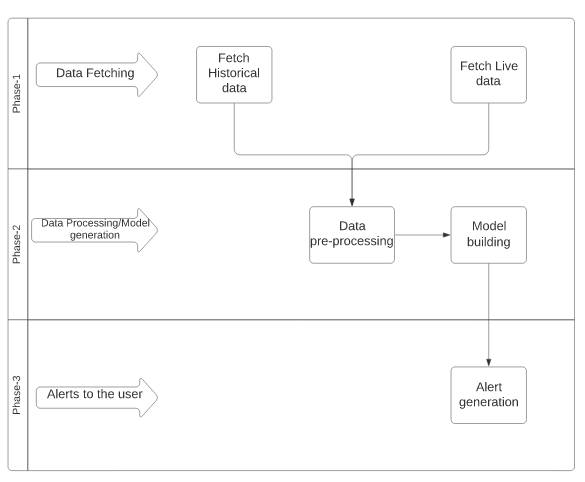


Figure 4 Project wireframe

The user personal mapping is as follows:

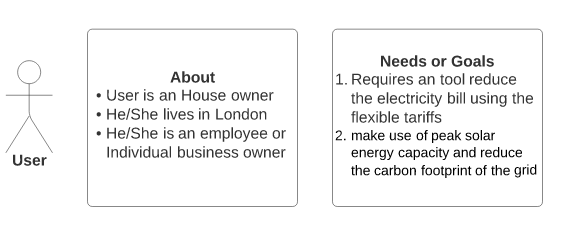


Figure 5 User persona map

## Data fetching

The core of the project is fetching data from all the existing APIs. There are many rules to access the APIs. Most of them were open while premium data APIs are paid versions. According to programmable web [35] there are around 24237 APIs registered around the globe by 2021. There are many energy related APIs both free and paid. Since our objective of the project is to rely on open sources and related to energy data of United Kingdom. The open APIs used to fetch all the data required for the project are listed below:

1. https://solcast.com/ (API used to fetch live and historic solar related data)

2. https://www.bmreports.com/bmrs/?q=help/about-us (API includes data summary of energy sources, system prices and demand)

3. <https://webhose.io/topics/news-api/> (API is used to fetch news about the weather)

4. https://openweathermap.org/api (This API is used to fetch the live weather data to process the alerts)

5. <https://developer.accuweather.com/accuweather-forecast-api/apis> (API is used to fetch the hourly weather forecast for the next 12 hrs)

The fore mentioned APIs are used in this project to fetch all the data required.

Solcast API is used to fetch data related to the solar energy generated, historic data. Most of the data available is about the forecasting and solar modelling from a meteorologist’s perspective working with solar energy generation sector. Solcast claims to have data about the solar energy generated to be highest quality in the field. The site gives the user an option to access the solar irradiance and PV power as well across the globe. The site gives limited free access to the data via API calls. This data is accessed via python using the ‘pvlive\_api’ library [36]. This is an open library which is wrapper over the solcast API gives ease to access the data. The sample code for accessing the data is as mentioned below

pvl = PVLive()

df = pvl.between(start=datetime.datetime.now(datetime.timezone.utc) - datetime.timedelta(days=30), end=datetime.datetime.now(datetime.timezone.utc), dataframe=True)

As mentioned, the historic data of past 30 days is retrieved from the API. The output dataframe ‘df’ looks as below

| **pes\_id** | **datetime\_gmt** | **generation\_mw** |
| --- | --- | --- |
| 0 | 2021-07-26 13:30:00+00:00 | 6240.0000 |
| 0 | 2021-07-26 13:00:00+00:00 | 6420.0000 |
| 0 | 2021-07-26 12:30:00+00:00 | 6270.0000 |
| 0 | 2021-07-26 12:00:00+00:00 | 6300.0000 |
| 0 | 2021-07-26 11:30:00+00:00 | 6120.0000 |

Where,

pes\_id - Public Electricity Supplier (national data collected),

datetime\_gmt – Date and Time index in GMT time zone format, and

generation\_mw – Amount of solar energy generated nationwide in Megawatts (MW)

The second API used is the BMRS data from Elexon portal [15]. BMRS is a primary source of data regarding the Great Britain Electricity balancing and settlements arrangements. The sole purpose is to enable the participants in the bids with active information as well as the historic data. The data categories available on the site are:

1. Electricity Data Summary
2. REMIT
3. Transparency
4. Transmission
5. Generation
6. Demand
7. Balancing

Data used for this project is Electricity Demand, Generation and Price dynamics of the electricity tariffs are fetched from the website. There is an exhaustive list of APIs available on BMRS website which can be found in Appendix F. The data used for this project is the ‘FUELHH’ (halfHourlyOutTurnGenerationByFuelTypeService). Library named ‘httplib2’ is used to fetch the data from the API get url:

[https://api.bmreports.com/BMRS/FUELHH/<VersionNo>?APIKey=<APIKey>&FromDate=<FromDate>&ToDate=<ToDate>&ServiceType=<xml/XML/csv/CSV](https://api.bmreports.com/BMRS/FUELHH/%3cVersionNo%3e?APIKey=%3cAPIKey%3e&FromDate=%3cFromDate%3e&ToDate=%3cToDate%3e&ServiceType=%3cxml/XML/csv/CSV)>

This URL fetches data of the energy generated by fuel type except solar energy from ‘FromDate’ to ‘ToDate’ in the frequency of 30min. The output can be fetched in the form of csv or xml. I have fetched last 30 days data in the form of csv and saved the data in ‘historic\_generation.csv’.

The fore mentioned APIs are used to fetch the data related to the UK grid electricity. By default, the user location is set to London. The next APIs are Openweather API [37] and Accuweather API [38] are related to weather APIs. The data fetched from the Openweather is the historic weather data and Accuweather provide hourly forecasts of weather for the coming 12 hours. The API URLs are as follows:

Openweather: api.openweathermap.org/data/2.5/forecast?q={city name}&appid={API key}

Accuweather: [http://dataservice.accuweather.com/forecasts/v1/hourly/12hour/{city](http://dataservice.accuweather.com/forecasts/v1/hourly/12hour/%7bcity) name}?apikey={API key}

Where

City name = name of the location

API key = unique key for the account created by the user.

The data related to historic energy generation by fuel type, weather forecasts and weather history is fetched. This data is used for energy generation forecasting as described in the following sub section.

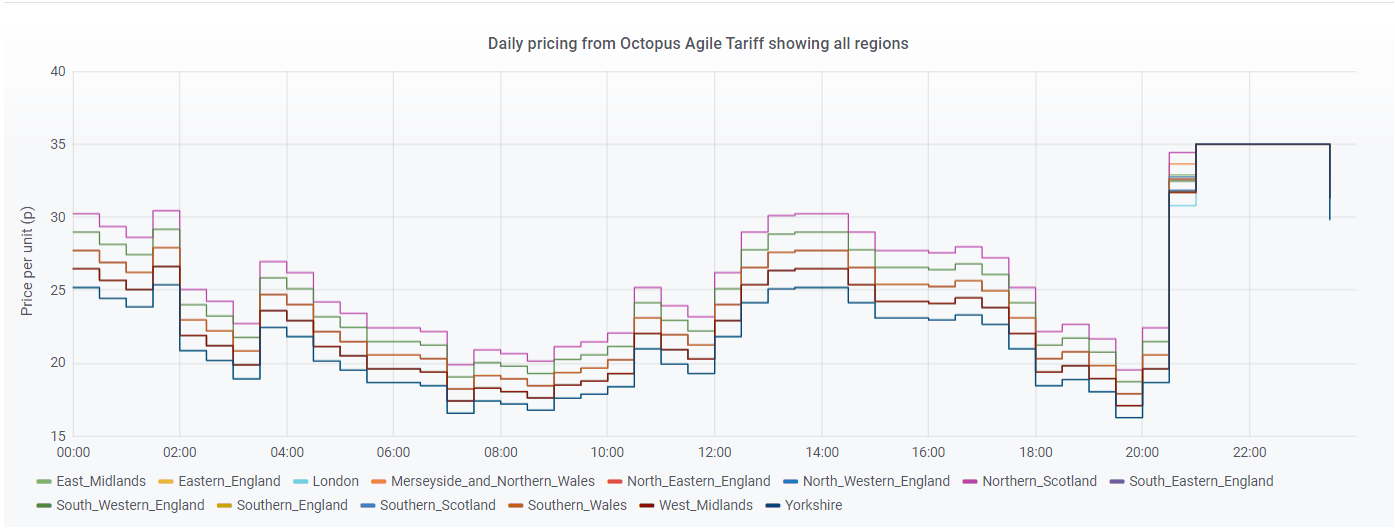
The objective of the project to reduce the carbon emissions while reducing the electricity bills of the consumer. The energy data is now available, I must fetch the data related to the electricity tariffs of Great Britain. As mentioned in the literature survey the national grid energy is distributed to DNOs while DNOs supply, distribute electricity to local operators. The tariffs are set by the DNOs in the Great Britain. While the tariffs vary with the location the consumer is in, the movement pattern of the price is similar to all locations as shown in Figure below.

Figure 6 Daily pricing from Octopus Agile Tariff showing all regions [39]

This data is available live on Agile Octopus [25] website which is a paid service to be setup by customers to know about the usage of the electricity on various appliances at different time in a day. The pricing data is available for the public for free on energy stats website [39] for free. There isn’t any restriction on scraping the website. According to the website the daily wholesale energy prices and providing the next day’s pricing sometime after 4pm GMT for the day before. To fetch this data automatically, python scraping tool is built to scrap the website at specific times in a day and download the required tariff file. Since the daily movement of the prices from various DNOs is similar for comparative purposes I have choose only Eastern England tariff data. All the required raw data is fetched from the APIs and Web scrapping tool. The following sub sections explains about the pre-processing steps on the data.

## Data processing

The previous section explained about collecting data from multiple sources using python libraries such as ‘httplib2’, ‘selenium’ and ‘PV\_live’. These libraries connected directly to the APIs or website end points and fetched data. This section explains on how the raw data collected is processed for model building and optimisation.

The raw data collected from various sources in different formats like csv, xml and json. These formats are to be adjusted to combine whole data and pre-process the data. The data fetched form the BMRS website is in the form of either xml or csv, data from PV\_live is in data frame format which is a python format, weather data is in the form of json and tariff data from agile octopus is downloaded as .csv file. These data formats are converted into pandas data frame format in python which increases the speed of processing the data.

Another challenge in the data is that the time calculated in the data is in different time zones. Some of them are in Greenwich Mean Time (GMT) while others are in British Summer Time (BST). In Great Britain, the standard time zone followed is GMT while during the Summer period for Day Light Saving. To ensure uniformity all the data values are converted to Coordinated Universal Time (UTC). UTC is primary standard for time followed by the world, regulates clocks and time. In general, GMT is same as UTC format while BST is UTC + 1hr. The difference between the GMT and BST is 1 hr. The user is provided with input ‘Day\_light\_savings’ which is a Boolean value which adjusts the values with respective to time zone and recommendations are generated in standard UTC format.

After bringing all the data from BMRS and PV\_live data for the past 30 days in different data frames, data is merged over the ‘datetime’ to form a single dataframe. The dataframe consists of the datetime values and the past energy generated values by fuel type. The energy sources in the dataframe are SOLAR, BIOMASS, WIND, NUCLEAR, PUMPED HYDRO, NON-PUMPED HYDRO, OIL, GAS TURBINES, COAL, INTERNCONNECTORS, OTHERS. These sources are categorised into LOW CARBON and HIGH CARBON where,

LOW CARBON = SOLAR + BIOMASS + WIND + NUCLEAR + PUMPED HYDRO + NON-PUMPED HYDRO

HIGH CARBON = OIL + COAL + GAS TURBINES + INTERCONNECTORS + OTHERS

The ratio of the LOW CARBON to HIGH CARBON is calculated as a new variable ‘ratio’,

This ratio value is given as input for time series forecasting module with ‘datetime’ as index value.

## Time series forecasting

The data is pre-processed as mentioned in the previous subsection and ready as input for time series forecasting model. Multiple papers were published on time series forecasting of the energy generation for individual fuel type. The sum of energy generated by all the fuel types is novel. Multiple forecasting models are trained with the input data namely Autoregressive Integrated Moving Average (ARIMA), naïve forecaster, Random Forests, Extreme Gradient Boosting.

The libraries used for the model training are ‘pmdarima’, ‘sktime’, ‘scikit-learn’ and ‘xgboost’

Pmdarima – Statistical library to fill the hole in python’s time series analysis capabilities like ARIMA models which is equivalent of R's ‘auto.arima’ functionality

Sktime – Provides specialized time series algorithm tools which are also compatible for scikit-learn library

Scikit-learn – Python machine learning module offering various machine learning models for training

Xgboost – Python version for implementing Extreme Gradient Boosting algorithm.

3-different models are trained using the above algorithms.

1. Low carbon energy generation is forecasted for the next 24hrs
2. High carbon energy generation is forecasted for the next 24hrs
3. Ratio of Low carbon to High Carbon is forecasted for the next 24hrs

These models are developed in conjunction and the performance of these models are presented in the Results section.

## Optimisation methods

Now the energy generation forecasting results are available using the various time series forecasting models. I have combined the forecasting results with the tariff data forecasts scraped from energystats.uk to form a data frame. This dataset would contain the low carbon energy generation forecasts, high carbon energy generation forecasts and the ratio of low carbon and high carbon forecasts from the 3-different models developed in the previous subsection.

To achieve the final objective of reducing the carbon emissions while having the lowest tariffs as possible. To quantify the carbon emissions, I have divided the low carbon to the high carbon ration and assigned to a variable ‘ratio’ and the tariffs are assigned to the variable ‘price’. I have to maximise the ratio and reduce the tariff. I have opted to choose a simple optimisation algorithm with equal weights to ratio and price as shown below.

This implies,

Our goal is to maximise the ‘objective’ as higher as possible. This is by finding the trade off point between ratio being higher and price being lower. The point where the ‘objective’ is highest is the ideal point for the consumer to use the applications. This is because the at this point the ratio is higher while having lowest tariffs as possible. Since the user would require more than an hour time for using various appliances, depending on the ‘user\_input’ which is the time window the consumer would require over the day this window is used to find the moving average over ‘objective’ variable. The highest value of moving average is the recommended as the ideal time for the consumer to start using the appliances for ‘user\_input’ hours.

The ideal time along with the weather report at that particular time fetched from the Accuweather hourly forecasts is sent as an email to the user automatically using ‘smtplib’ a library used in python to send automatic emails from the python code. The python code is set to run automatically every day at user specified time using inbuilt windows software named ‘task schedular’ or ‘crontab’ in macOS.

## Testing

The software developed is tested overall to validate and test the integrity of the code from failing. 3-tests are conducted over the code developed and published on the GitHub. Unit testing, Integration testing and Functional testing.

The whole testing code is written using ‘pytest’ and ‘unittest’ libraries in python. The code is tested on API level where the data is fetched from the APIs. The APIs are returned with a response code whenever a connection is established to signify the connection with the API. According to the standard rules if the response code is ‘200’ then the connection is successfully established with the API gateway. This response is tested win the ‘api\_fetch.ipynb’ file to test the connectivity all the APIs used in the software.

To check the inbuilt functions written ‘pytest’ is used to check the test cases and return for possible issues with the functions. The user defined function ‘dates\_list’ and ‘header\_list’ are defined to make a list of dates dynamically depending on the current date and create headers for the data fetched o the csv format respectively. These functions are passed and testes with various use cases to check the integrity of the code.

While the unit testing is done APIs and the integration tests on the user defined functions, the functionality of code is tested. The software is deployed on multiple machines with various user inputs, the output recommendations of the software are tested with actual values to test the validity of the software. All the unit, integration and functional testing results are tabulated in the Results section.

# Results

In the previous section, the design and developed procedure are laid out in detail. In this section I will present all the results generated during the development process and in detail show case the important code and plots. Along with the development, unit and functional testing is also done on the software to validate the results.

In Figure 7, data collected from the BMRS FUELHH API is presented. It fetches data in 22 different columns which are datetime and energy generated by 17 different fuel type. These 17 different fuel types are segregated into low carbon and high carbon data and summed up in the next step. This dataset from BMRS lacks the solar data which is to be separately fetched from the PV\_live library in python as mentioned in the previous sections. The data from PV\_live is fetched (as shown in Figure 8), columns renamed and merged with the BMRS data frame to form dataset with all the data available.

Once the data is available, the 18 fuel types are segregated and added to form low carbon and high carbon variables and shown in Figure 9. The low carbon and high carbon are plotted against in a single plot (Figure 10) to show the variability in the data and how low carbon complements the high carbon energy. From Figure we can conclude that low carbon and high carbon data are inversely proportional to each other. This is expected because the availability in low carbon or renewable energy is adjusted in the generation in high carbon data or non-renewable energy data and vice versa. Depending on the availability one source the other sources are adjusted accordingly to meet the electricity demand on UK grid.

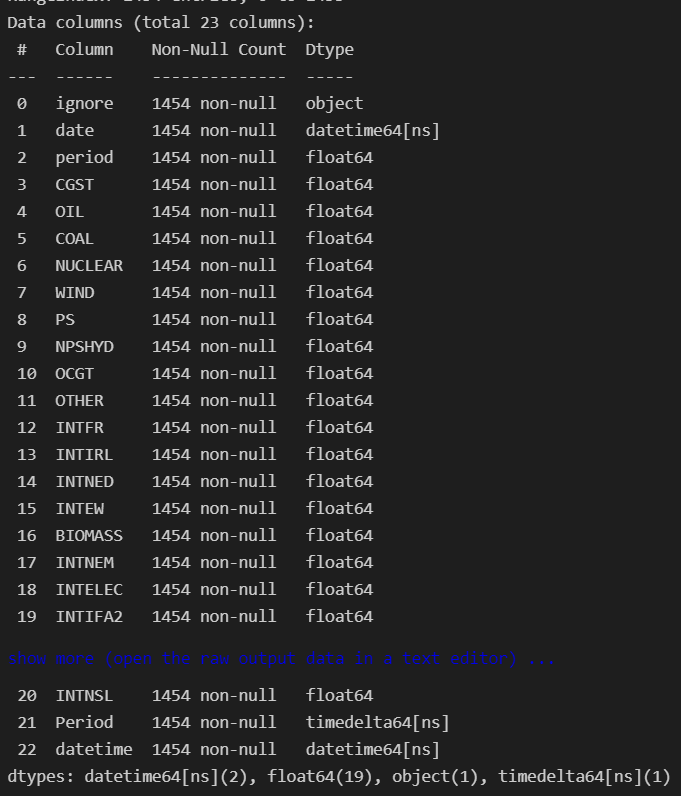


Figure 7 Data collected from BMRS FUELHH API

PV\_live data

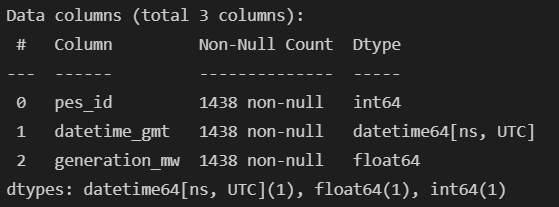


Figure 8 Data collected from PV\_live API

Low carbon and High carbon variables

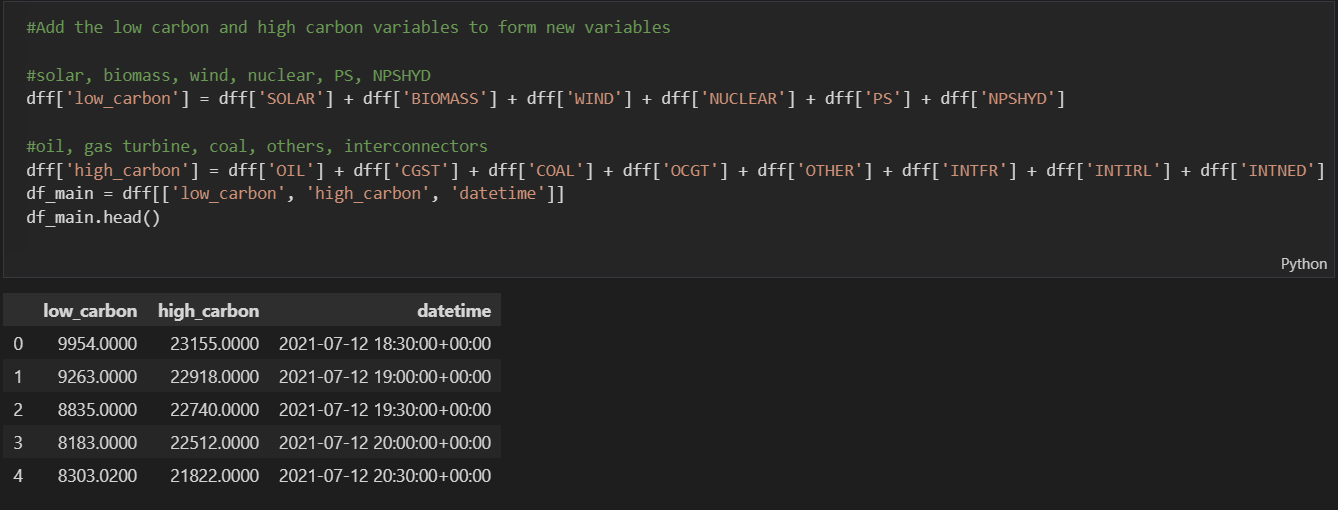


Figure 9 Segregation of low carbon and high carbon dependent energy generated

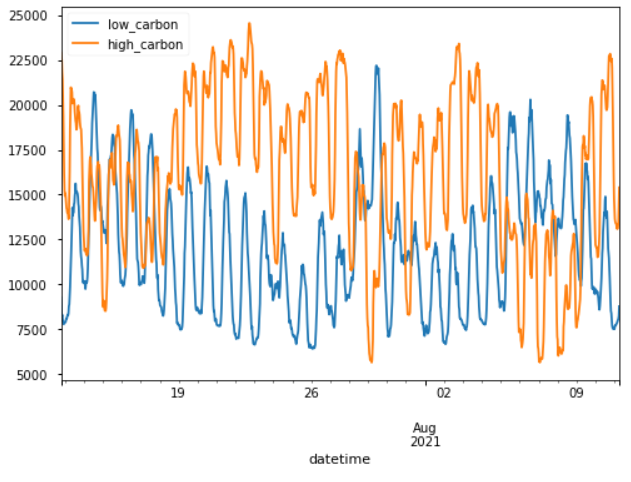


Figure 10 Sample plot of the movement of the low carbon and high carbon energy

Figure 11 shows the variation of the ratio in a single day. It signifies the pattern followed by the ratio as the low carbon being available the most during the daytime as solar dominates the energy generated during day light and coal/oil demand dominates during the unavailability of day light or other sources of energy.

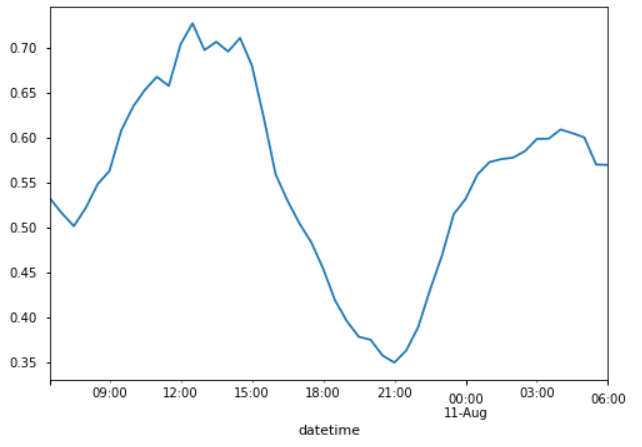


Figure 11 Movement of the low carbon to high carbon ratio in 24hrs

Once the data is merged and available for training, 3-datasets are trained using 4-different models using k-fold cross validation which means data is trained on single dataset multiple times to test the accuracies. The performance is measured in terms of MAPE. The results are tabulated in the table below along with the multiple screenshots of the results predicted compared to actual.

## Low carbon data results

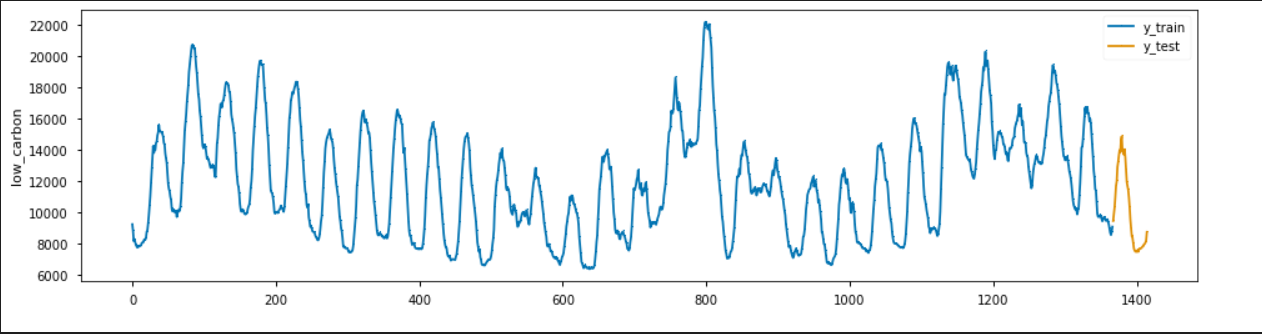


Figure 12 Test train split of the low carbon data for model input

Naïve forecaster

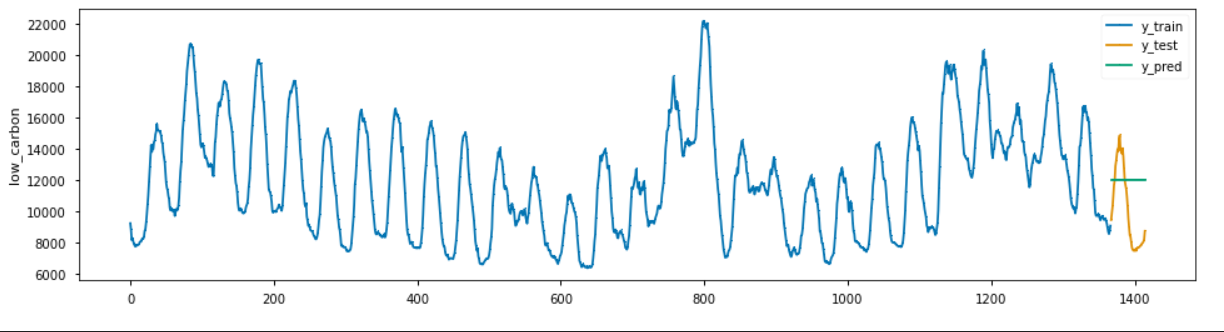


Figure 13 Naive Forecasters : Model predictions vs Actual

AutoARIMA

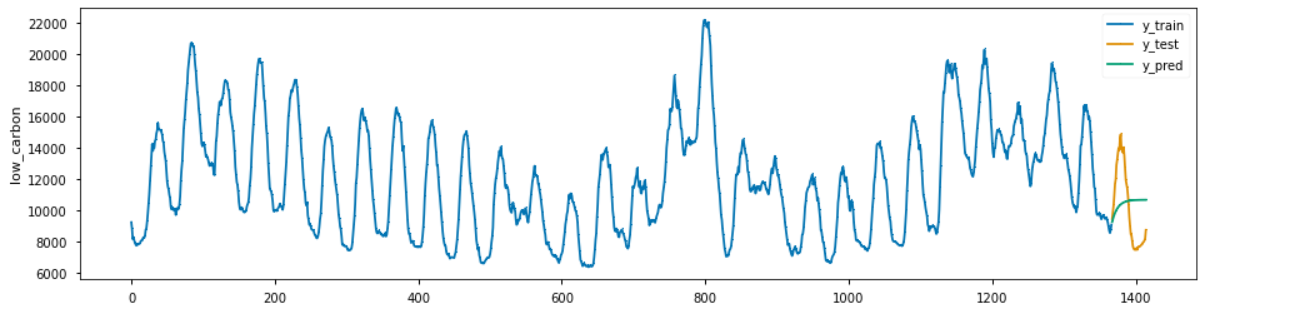


Figure 14 AutoARIMA : Model predictions vs Actual

Random Forest

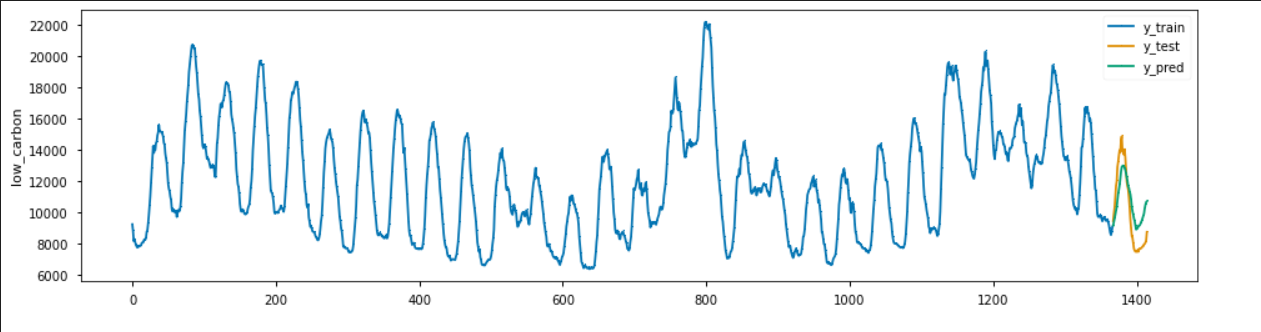


Figure 15 Random Forest : Model predictions vs Actual

XGBoost regressor

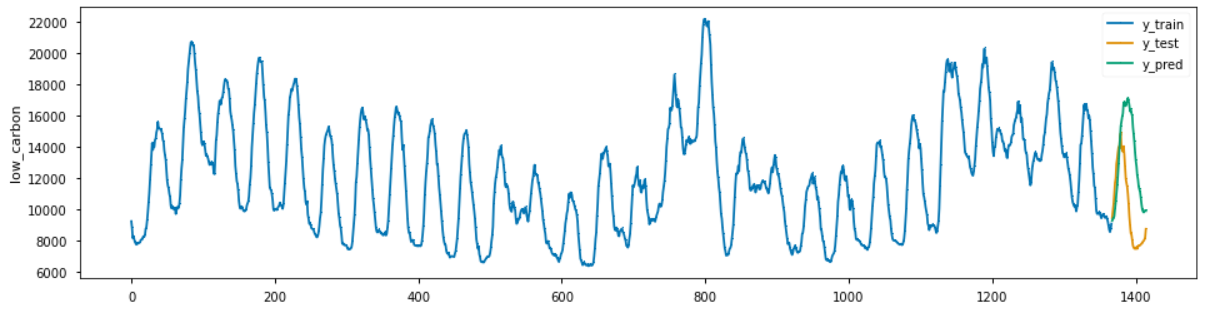


Figure 16 XGBOOST regressor : Model predictions vs Actual

## High carbon results

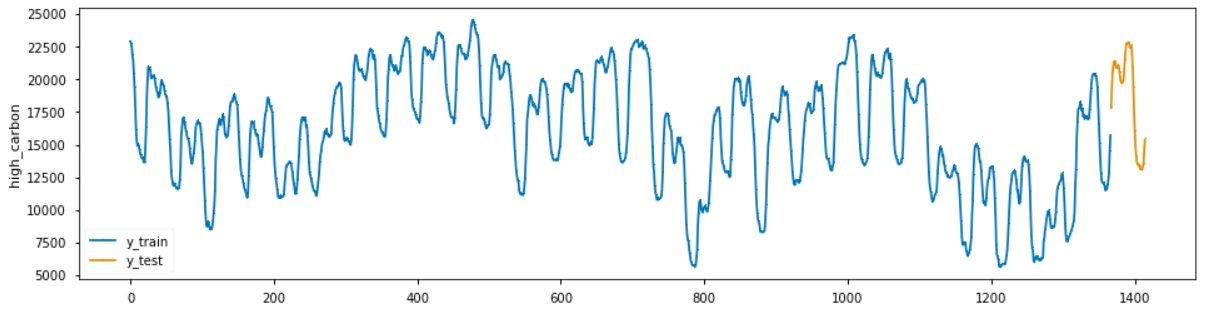


Figure 17 Test train split for high carbon data

Naïve forecaster

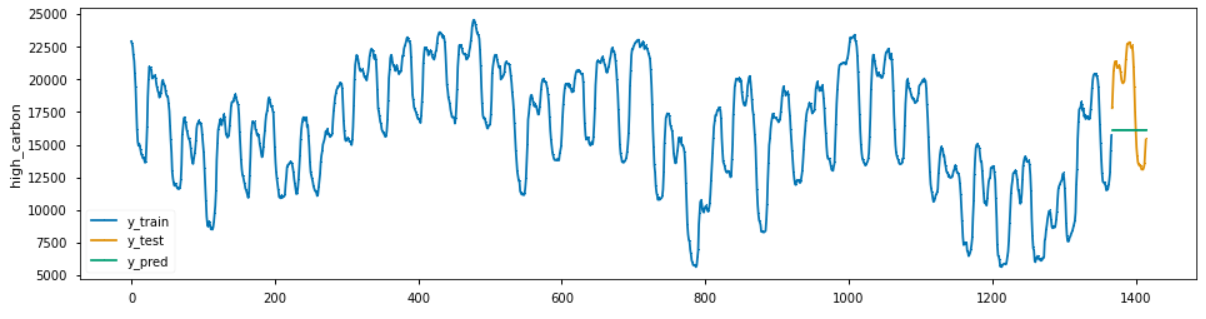


Figure 18 Naive Forecaster : Model predictions vs Actual

Random Forest

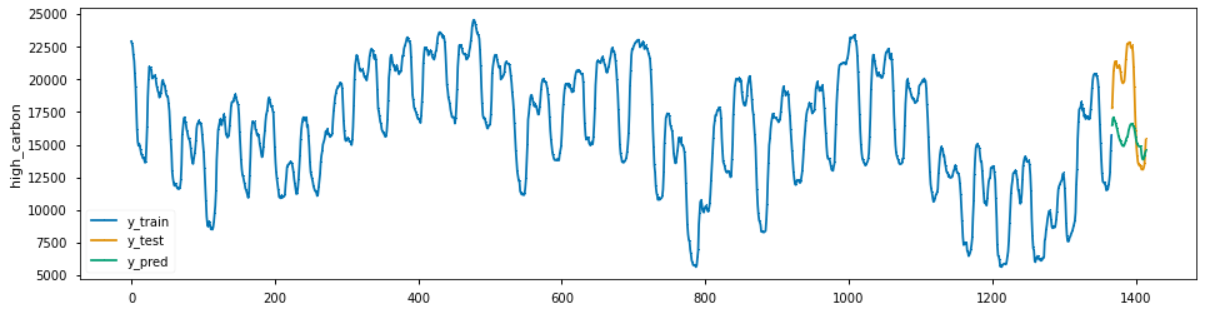


Figure 19 Random Forest : Model predictions vs Actual

XGBoost regressor

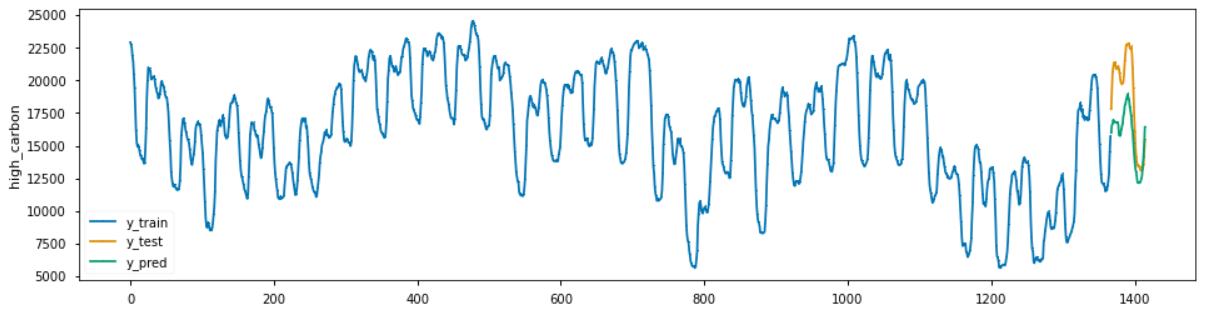


Figure 20 XGBOOST regressor : Model predictions vs Actual

## Ratio data results

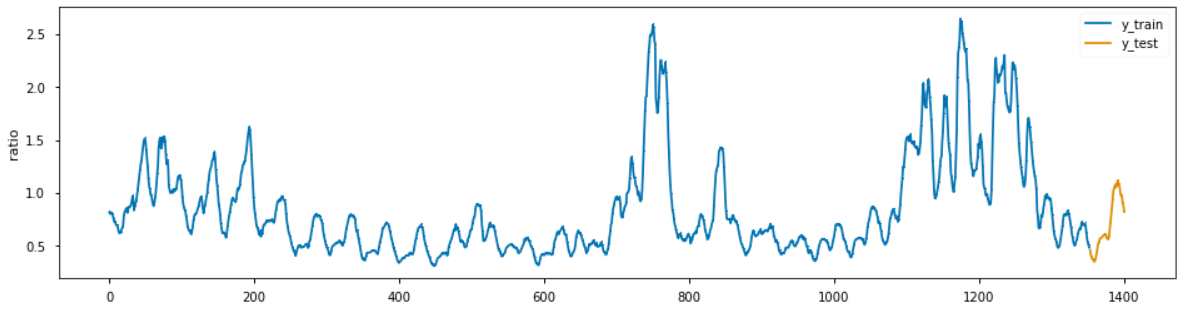


Figure 21 Test Train split for ratio data

Naïve forecaster

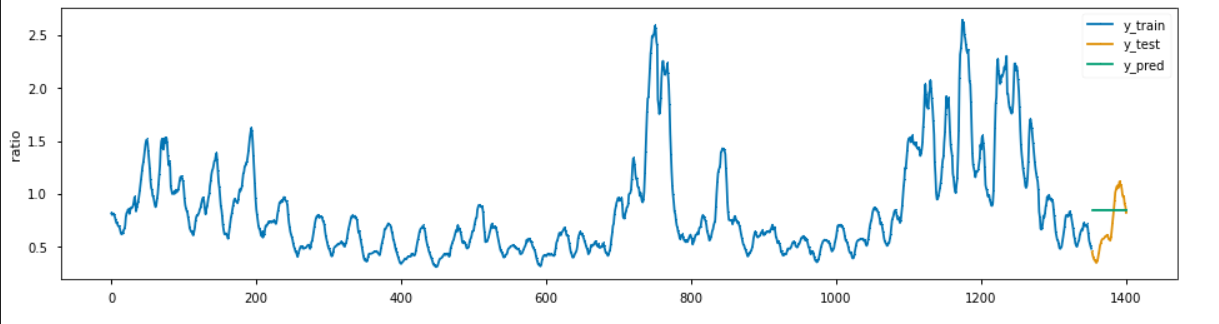


Figure 22 Naive forecaster : Model predictions vs Actual

Random Forest

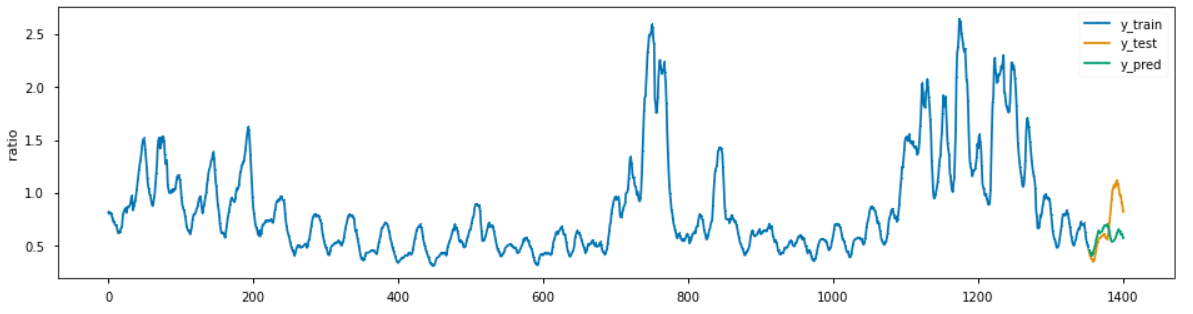


Figure 23 Random Forest : Model predictions vs Actual

XGBoost regressor

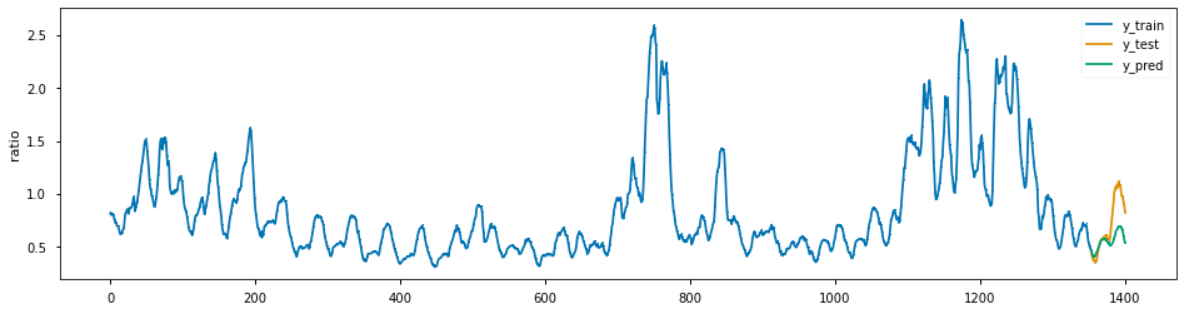


Figure 24 XGBoost regressor : Model predictions vs Actual

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Average MAPE (%)** | **Naïve Forecaster** | **Auto ARIMA** | **Random Forest** | **XG Boost** |
| **Low carbon** | 33.3 | 32.2 | 14.7 | 30.9 |
| **High carbon** | 36.2 | NA | 26.5 | 22.3 |
| **Ratio** | 23.6 | NA | 20.3 | 16.6 |

The results tabulated in the above table are the reason to choose ratio dependent models are accurate compared to the other models. Ratio models and the predicted values are used in the final suggestions to the customer/users.

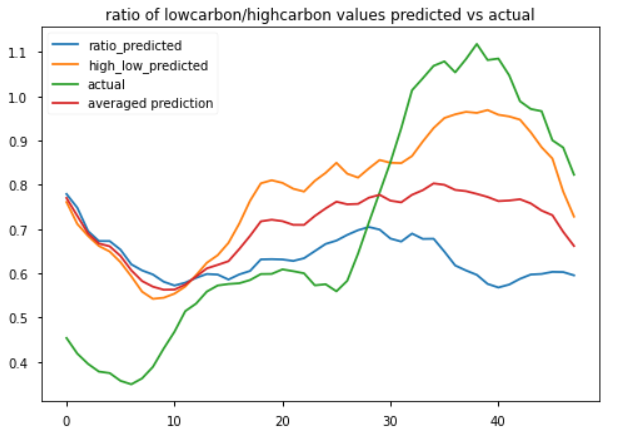


Figure 25 Comparison of results-1

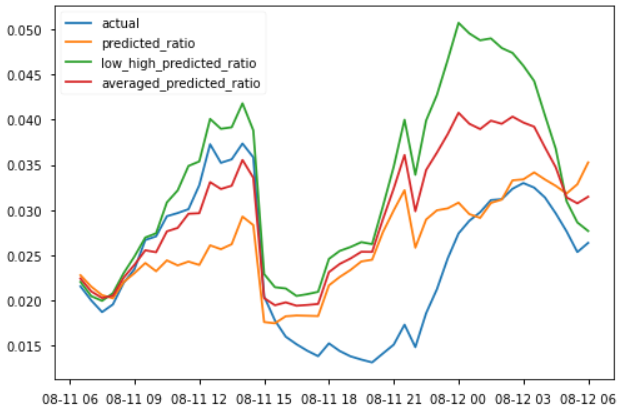


Figure 26 Comparison of results-2

Figure 25-26are plots when the predicted results are plotted against the actual values of the ratio of the low carbon to high carbon. Once the forecasts are available it is to be compared with tariffs data for optimising the values. The price data as shown in the Figure 27 depicts the dataset available for tariffs in Eastern England DNO and price is the price per unit (£)

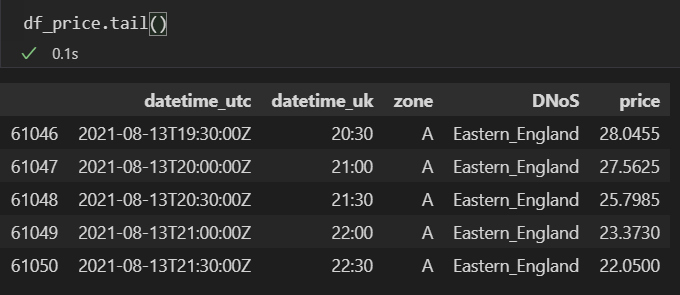


Figure 27 Tariff details fetched from energystats.uk

Obtaining the energy generation forecast and the Agile tariffs, optimisation algorithm as explained in the previous section is applied and the ideal time for user are sent via email to the user in the following format which contains ideal time and the weather at that particular time

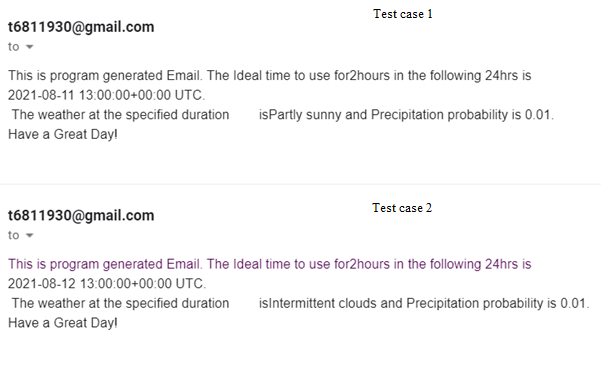


Figure 28 Program generated email sent to the user - 2 Test cases

## Testing results

The generated results and code are tested in unit level and the functional level. There are multiple test cases generated (Figure 29) and passed using the ‘pytest’ python library. Also, in ‘fetch\_api.ipynb’ the API output response is tested using assert statement to test whether the connect is successful or not. For e.g.

assert res.status\_code == 200

where ‘200’ is the response code form the connection to denote successful connection.

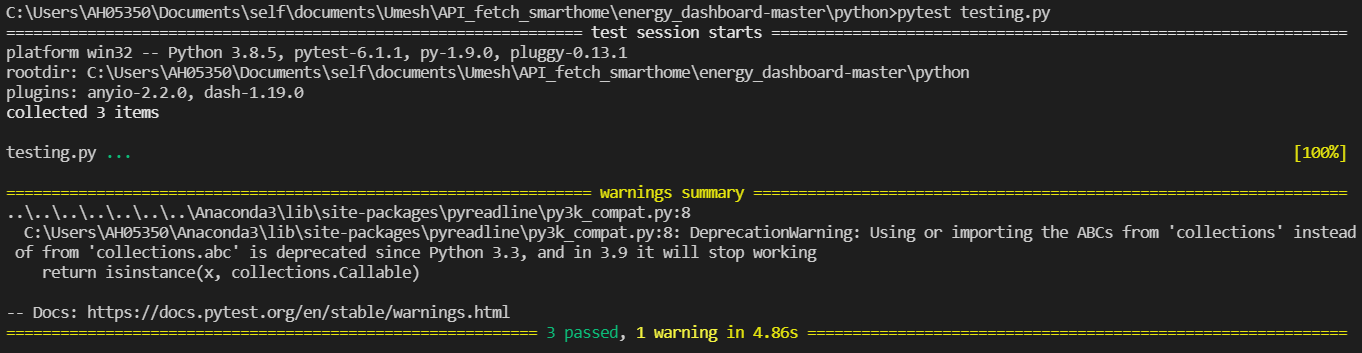


Figure 29 Testing use cases for various functions result

The APIs and integrations are tested in the ‘testing.py’ and ‘fetch\_api.ipynb’ as uploaded in the GitHub repository. The final recommendation sent to the consumer are also tested using 4-different test users, where 4 users have various user inputs to test every day. The users are given various recommendations depending upon the user input values. One of the user outputs is tested and presented below. In Figure 30, the plot signifies the actual ratio values of the energy and tariffs

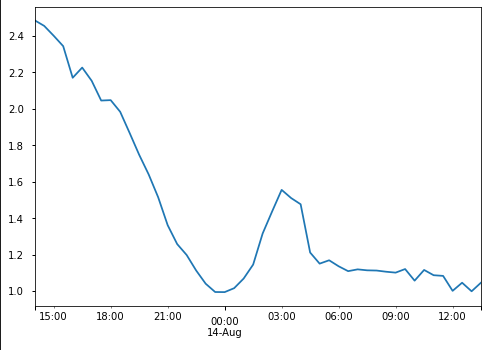


Figure 30 Actual ratio values of energy/tariff

Depending on the Figure above, we can predict that the ideal time to use the appliances would be around 14:00. In Figure 31, the program recommended email is attached which shows the recommendation of using the appliances at 14:00 for 3hours, thus validating the program recommendations are in line with the actual values.

This ends the Results section presenting the development, testing results of the software that is development. In the section, I will conclude and find opportunities of future work that can be done

# Conclusion and Future work

In this project, I have proposed a novel technique to create a Smart home notifier depending on the open source data available. This notifier helps consumers to reduce the carbon footprint and tariff charges while using the home appliances. The users have to enter the amount of time required on a given day as well as the location. The software processes the data collected from the APIs namely Solcast, Accuweather, BMRS to a similar format and merge all the data. The merged data is used to forecast the ratio low carbon to high carbon energy generation. The forecasted ratio is used in the optimisation method.

To find the details about the tariff, Agile Octopus data on electricity tariff is available on the energystats.uk website. Tis data is scraped from the website using a python web scraping library selenium. The scraped data consists of the tariff forecasts by region for the following the day. Since the tariff from different DNOs vary similarly the data from any region would be sufficient for the project.

After predicting the low carbon to high carbon energy generation ratio and scraping the tariff data. Simple optimisation method is proposed which says that low carbon to high carbon energy generated is directly proportional to the final objective and the tariff values are inversely proportional to the final objective, the ratio of the energy forecasted, and tariff values is calculated. Depending on the user input for the amount of time required in a day, moving average over the final ratio is calculated using the specified time window by the user. The maximum moving average is found and the time corresponding to that average is sent to the user along with the weather data fetched from the Accuweather API via email in specified format.

The future work consists of exploring more optimisation algorithm for finding ideal time, forecasting methodology can be applied individually for the fuel types and the low carbon to high carbon ratios are predicted depending on individual fuel type forecast for more accurate information and providing more features to the user as inputs and output can be given by other modes of communication.

References

[1] M. Venkataramanan and others, “Causes and effects of global warming,” *Indian J. Sci. Technol.*, vol. 4, no. 3, pp. 226–229, 2011.

[2] I. Haldar, *Global Warming: The Causes and Consequences*. Readworthy, 2011.

[3] J. G. Titus, “The causes and effects of sea-level rise,” *Eff. Chang. Stratos. Ozone Glob. Clim.*, vol. 1, pp. 219–248, 1986.

[4] D. E. Abrahamson, “The challenge of global warming,” Jan. 1989, [Online]. Available: https://www.osti.gov/biblio/5625979

[5] K. Mulvaney, “Climate change report card: These countries are reaching targets.” https://www.nationalgeographic.com/environment/article/climate-change-report-card-co2-emissions

[6] N. Höhne *et al.*, “Emissions: world has four times the work or one-third of the time.” Nature Publishing Group, 2020.

[7] newsroom edie, “National Grid: UK’s electricity system could go carbon-negative from 2033,” Jul. 27, 2020. https://www.edie.net/news/6/National-Grid--UK-s-electricity-system-could-go-carbon-negative-from-2033/

[8] S. Rishi, C. Priyanka, and S. Sunil, “Home-Automation System for Renewable Energy Self-Consumption Optimization,” Nov. 2015, [Online]. Available: http://computers.stmjournals.com/index.php?journal=JoSETTT&page=article&op=view&path%5B%5D=2632

[9] “Electricity Generation.” https://www.energy-uk.org.uk/our-work/generation/electricity-generation.html

[10] Government, *U.K National Grid Energy Flow API*. 2012.

[11] L. Talia, “These 11 sinking cities could disappear by 2100.” https://www.weforum.org/agenda/2019/09/11-sinking-cities-that-could-soon-be-underwater/

[12] nationalgrid, “Interconnectors,” *About interconnectors*. https://www.nationalgrid.com/our-businesses/national-grid-ventures/interconnectors-connecting-cleaner-future

[13] H. Isabelle, “Great Britain’s monthly electricity stats.” https://www.nationalgrideso.com/electricity-explained/electricity-and-me/great-britains-monthly-electricity-stats

[14] M. Matt, “DNOs: Who is my distribution network operator?,” *A short guide to DNOs and what they do*, Dec. 04, 2020. https://www.ovoenergy.com/guides/energy-guides/dno

[15] “ELEXON BMRS,” *neta - The New Electricity Trading Arrangements*. https://www.bmreports.com/bmrs/?q=help/about-us

[16] T. Berry, “Standards for energy management system application program interfaces,” in *DRPT2000. International Conference on Electric Utility Deregulation and Restructuring and Power Technologies. Proceedings (Cat. No. 00EX382)*, 2000, pp. 156–161.

[17] U. Farooq and D. Zirkler, “API peer reviews: a method for evaluating usability of application programming interfaces,” in *Proceedings of the 2010 ACM conference on Computer supported cooperative work*, 2010, pp. 207–210.

[18] M. Al-Janabi, “Design of Smart Home Model Using Solar Energy,” Apr. 2017.

[19] Simon Evans, “Solar is now ‘cheapest electricity in history’, confirms IEA,” *The world’s best solar power schemes now offer the “cheapest…electricity in history” with the technology cheaper than coal and gas in most major countries.* https://www.carbonbrief.org/solar-is-now-cheapest-electricity-in-history-confirms-iea

[20] R. Mallwitz and B. Engel, “Solar power inverters,” in *2010 6th International Conference on Integrated Power Electronics Systems*, 2010, pp. 1–7.

[21] R. Catherine, “Solar costs have fallen 82% since 2010.” https://www.pv-magazine.com/2020/06/03/solar-costs-have-fallen-82-since-2010/

[22] H. A. Sher and K. E. Addoweesh, “Micro-inverters — Promising solutions in solar photovoltaics,” *Energy Sustain. Dev.*, vol. 16, no. 4, pp. 389–400, 2012, doi: https://doi.org/10.1016/j.esd.2012.10.002.

[23] “Office of ENERGY EFFICIENCY & RENEWABLE ENERGY,” *Solar Integration: Inverters and Grid Services Basics*. https://www.energy.gov/eere/solar/solar-integration-inverters-and-grid-services-basics

[24] “SSE,” *What is a smart meter?* https://sse.co.uk/b/smart-meters?mmredirect=false&mxrd=215396

[25] *Agile Octopus*. octopusenergy. [Online]. Available: https://octopus.energy/about-us/

[26] J. Ofoeda, R. Boateng, and J. Effah, “Application Programming Interface (API) Research: A Review of the Past to Inform the Future,” *Int. J. Enterp. Inf. Syst.*, vol. 15, pp. 76–95, Jul. 2019, doi: 10.4018/IJEIS.2019070105.

[27] A. Trabelsi, Z. Elouedi, and E. Lefevre, “Decision tree classifiers for evidential attribute values and class labels,” *Fuzzy Sets Syst.*, vol. 366, pp. 46–62, 2019, doi: https://doi.org/10.1016/j.fss.2018.11.006.

[28] W. Mao, “Cultural Modeling for Behavior Analysis and Prediction,” 2012, pp. 91–102. doi: 10.1016/B978-0-12-397200-2.00008-7.

[29] L. Breiman, “Random Forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.

[30] N. Farnaaz and M. A. Jabbar, “Random Forest Modeling for Network Intrusion Detection System,” *Procedia Comput. Sci.*, vol. 89, pp. 213–217, 2016, doi: https://doi.org/10.1016/j.procs.2016.06.047.

[31] A. Mubayi, “Chapter 10 - Computational Modeling Approaches Linking Health and Social Sciences: Sensitivity of Social Determinants on the Patterns of Health Risk Behaviors and Diseases,” in *Disease Modelling and Public Health, Part A*, vol. 36, A. S. R. S. Rao, S. Pyne, and C. R. Rao, Eds. Elsevier, 2017, pp. 249–304. doi: https://doi.org/10.1016/bs.host.2017.08.003.

[32] C. Tao, D. Shanxu, and C. Changsong, “Forecasting power output for grid-connected photovoltaic power system without using solar radiation measurement,” in *The 2nd International Symposium on Power Electronics for Distributed Generation Systems*, 2010, pp. 773–777. doi: 10.1109/PEDG.2010.5545754.

[33] Z. Wang, L. He, X. Cheng, and J. He, “Method for short-term photovoltaic generation power prediction base on weather patterns,” in *2014 China International Conference on Electricity Distribution (CICED)*, 2014, pp. 213–215. doi: 10.1109/CICED.2014.6991696.

[34] K. Kiranvishnu, K. Sireesha, and J. Ramprabhakar, “Comparative study of wind speed forecasting techniques,” in *2016 Biennial International Conference on Power and Energy Systems: Towards Sustainable Energy (PESTSE)*, 2016, pp. 1–6. doi: 10.1109/PESTSE.2016.7516424.

[35] “ProgrammableWeb,” *Search the Largest API Directory on the Web*. https://www.programmableweb.com/apis/directory

[36] T. Jamie, “PV\_Live API documentation.” [Online]. Available: https://sheffieldsolar.github.io/PV\_Live-API/build/html/modules.html#pvlive\_api.pvlive.PVLive.between

[37] “5 day weather forecast.” OpenWeather. [Online]. Available: https://openweathermap.org/forecast5

[38] “12 Hours of Hourly Forecasts.” Accuweather APIs. [Online]. Available: https://developer.accuweather.com/accuweather-forecast-api/apis/get/forecasts/v1/hourly/12hour/%7BlocationKey%7D

[39] Zarch, “energy stats.uk,” *Agile Tariff*. https://www.energy-stats.uk/

1. Project Initiation Document
2. Project Proposal Form
3. Midpoint Review Feedback
4. Project Plan
5. Supervision Meeting Records

## UK Grid info

## Balancing Mechanism Reporting Service (BMRS)

This BMRS is the primary channel for providing operational data relating to the GB Electricity Balancing and Settlement arrangements. It is used extensively by market participants to help make trading decisions and understanding market dynamics and acts as a prompt reporting platform as well as a means of accessing historic data. The main data categories of the website are as follows:

Electricity Data Summary: Includes 9 frequently requested reports including System Prices, Generation by Fuel Type and System Demand.

REMIT: Information provided by market participants to comply with Article 4 of Regulation on Wholesale Energy Market Integrity and Transparency (REMIT) Regulation (EU) 1227/2011.

Transparency: Data for the Transparency Regulation (EU) 543/2013 originating from the Transmission company and market participants.

Transmission: Balancing Mechanism data from the Transmission company and data includes System Operator to System Operator Trades (SO-SO), SO-SO Trade Prices and System Warnings.

Generation: Generation data including Wind Forecast Out-turn, Generation Output Useable (forecast) and Generation by Fuel Type (actual).

Demand: Demand data including Demand Day Ahead (forecast), Initial Demand Out-turn (actual) and Peak Demand data.

Balancing: Data includes system prices, Balancing Mechanism Unit (BMU) data such as Final Physical Notification (FPN), Maximum Export/Import Limits (MIL/MEL), Bid Offer data.

The list of the API names that BMRS offers are as follows as mentioned in the BMRS Data Push API User Guide:

### Transparency Data and REMIT

1 B1720 – Amount of Balancing Reserves Under Contract

2 B1730 – Prices of Procured Balancing Reserves

3 B1740 – Accepted Aggregated Offers

4 B1750 – Activated Balancing Energy

5 B1760 – Prices of Activated Balancing Energy

6 B1770 – Imbalance Prices

7 B1780 – Aggregated Imbalance Volumes

8 B1790 – Financial Expenses and Income for Balancing

9 B1810 – Cross Border Balancing Volumes of Exchanged Bids and Offers

10 B1820 – Cross Border Balancing Prices

11 B1830 – Cross border Balancing Energy Activated

12 B0610 – Actual Total Load per Bidding Zone

13 B0620 – Day-Ahead Total Load Forecast per Bidding Zone

14 B0630 – Week-Ahead Total Load Forecast per Bidding Zone

15 B0640 – Month-Ahead Total Load Forecast Per Bidding Zone

16 B0650 – Year Ahead Total Load Forecast per Bidding Zone

### BMRS API and Data Push

17 B0810 – Year Ahead Forecast Margin

18 B1410 – Installed Generation Capacity Aggregated

19 B1420 – Installed Generation Capacity per Unit

20 B1430 – Day-Ahead Aggregated Generation

21 B1440 –Generation forecasts for Wind and Solar

22 B1610 – Actual Generation Output per Generation Unit

23 B1620 – Actual Aggregated Generation per Type

24 B1630 – Actual or Estimated Wind and Solar Power Generation

25 B0910 – Expansion and Dismantling Projects

26 B1320 – Congestion Management Measures Countertrading

27 B1330 – Congestion Management Measures Costs of Congestion Management

28 B0710 – Planned Unavailability of Consumption Units

29 B0720 – Changes in Actual Availability of Consumption Units

30 B1010 – Planned Unavailability in The Transmission Grid

31 B1020 – Changes in Actual Availability in The Transmission Grid

32 B1030 – Changes in Actual Availability of Offshore Grid Infrastructure

33 B1510 – Planned Unavailability of Generation Units

34 B1520 – Changes in Actual Availability of Generation Units

35 B1530 – Planned Unavailability of Production Units

36 B1540 – Changes in Actual Availability of Production Units

37 REMIT Flow – Message List Retrieval

38 REMIT Flow – Message Detail Retrieval

### Legacy BMRS Data

1 Temperature Data

2 Bid Offer Level Data

3 Credit Default Notice Data

4 System Warnings

5 Balancing Services Adjustment Action Data

6 Balancing Service Adjustment Data

7 Rolling System Frequency

8 Market Index Data

9 Daily energy Volume Data

10 Non BM STOR Instructed Volume Data

11 Applicable Balancing Services Volume Data

12 Rolling System Demand

13 Peak Wind Generation Forecast

14 Wind Generation Forecast and Out-turn Data

15 Generation by Fuel Type (Current)

16 Generation by Fuel Type (24H Instant Data)

17 Half Hourly Outturn Generation by Fuel Type

18 Half Hourly Interconnector Outturn Generation

19 National Output Useable (2-14 Days Ahead)

20 National Output Useable by Fuel Type (2-14 Days Ahead)

21 National Output Useable by Fuel Type and BM Unit (2-14 Days Ahead)

22 National Output Useable (2-52 Weeks Ahead)

23 National Output Useable by Fuel type (2-52 Weeks Ahead)

24 National Output Useable by Fuel Type and BM Unit (2-52 Weeks Ahead)

25 National Output Useable Data (2-156 Weeks Ahead)

26 National Output Usable by Fuel Type (2-156 Weeks Ahead)

27 National Output Useable by Fuel Type and BM Unit (2-156 Weeks Ahead)

28 Initial Demand Outturn

29 Forecast Day and Day Ahead Margin and Imbalance Data

30 Forecast Day and Day Ahead Demand Data

31 Demand & Surplus Forecast Data (2-14 Days Ahead)

32 Demand & Surplus Forecast Data (2-52 Weeks Ahead)

33 Demand & Surplus Forecast Data (2-156 Weeks Ahead)

34 SO-SO Prices (SO-SO)

35 SO Trades

36 Peak Demand – Yesterday/Today/Tomorrow

37 Indicative Peak Demand Information (Using Operational Metering Data)

38 System Demand

39 Indicative Triad Demand Information (Using Settlement Metering Data)

40 Physical Data

41 Dynamic Data

42 Derived BM Unit Data

43 Derived System Wide Data

44 Detailed System Prices

45 Market Depth Data

46 Latest Acceptances

47 Historic Acceptances

48 System Messages

49 BM Unit Search

50 System Warning (Today/Tomorrow)

51 System Warning (Historic)

52 Loss of Load Probability

53 Demand Control Instructions

54 STOR Availability Window

55 Trading Unit Delivery Mode

56 Settlement Exchange Rate

### Replacement Reserve Data

1 RR Bid Data

2 RR Aggregated Information Data

3 RR Activation Data

4 RR Interconnector Schedule

5 RR Activation Data

6 RR GB Need Met

7 RR Indicative Cashflow

## PV\_live API

The PV\_Live web API (Application Programming Interface) gives admittance to approach continuous and recorded assessments of PV age on the GB transmission organization. It returns a PC cordial (JSON) reaction for use in programming, sites, applications and so on

For specialized subtleties of how to utilize the API, including models, see the PV\_Live API User Guide. For a Python execution of the API, look at the PV\_Live-API project on GitHub.

Refreshed January 2021! In January 2021 we delivered another adaptation (v3) of the PV\_Live API that incorporates some minor refinements to the API boundaries and conduct, however for most clients is a straightforward update. Local PV outturn is totalled to Public Electricity Supplier (PES) district (a.k.a GSP Group district or DNO License Area). In the event that you'd prefer to get to local outcomes totaled to individual GSPs, get in touch with us to discover more. The past variant of the API (v2) will be stopped later on, however starting at 15/01/2021 no date has been se. Any individual who has joined to the PV\_Live User Mailing List will be told ahead of the v2 API being turned off.

Caution! We are effectively fostering the PV\_Live API, so we might change the base-URL, end-focuses, confirmation, reaction design and so forth Assuming you need to be advised of any progressions to the API ahead of time, kindly email solar@shef.ac.uk and we'll add you to our mailing list (the rundown is simply used to give refreshes about PV\_Live, you will not be spammed with bulletins and we will not impart any data to any outsiders).

## Energystats.uk And Agile Octopus

Energystats.uk is an open sourced website providing more information on the UK energy generation and price plunges. These process plunges are tabulated by Octopus, a company that provides products to provide user with informed decisions about the usage and price fluctuations. The more information about the website is given below.

By a long shot the most incessant solicitation I get to the site is "The place where would I be able to get full verifiable evaluating information for Agile as well as Agile Outgoing?". "As I need to do my own nerdy examination of the figures".

Energystats.uk then, at that point normally stir something up for the enquirer and send them a few information.

Be that as it may, pushing ahead I'm currently setting up the whole informational collection as a free download for both Agile and Agile Outgoing duties for every one of the 14 local DNO regions.

The two informational collections will be refreshed each day with the new evaluating from the Octopus API, so ordinarily at some point after around 4pm. Assuming you've had the everyday chart tweet from @energystatsuk, you realize the fundamental information will have been refreshed.

Note: The configuration of the CSV fields is:

UTC Time,

UK Time (HH:MM),

Region Code,

Region Name,

Unit Price (inc VAT)

Octopus by means of their API consistently distribute their information in UTC time, so you should apply the applicable BST/GMT relying upon the season. I as of now do that, thus the UK Time HH:MM field that I've included.

On the off chance that you discover this data valuable and are considering joining Octopus Energy, if it's not too much trouble, consider utilizing my reference code to assist with supporting running expenses of the site.

Energystats.uk can likewise be found on Twitter, kindly follow us @energystatsuk for every day Agile levy estimating charts and synopses.

Additionally, make certain to look at our Dashboard, Download Historical Data and OctoChargeCalc provisions to assist you with choosing whether these taxes are an ideal choice for you.

**Download Agile Import Tariff Pricing Data**

The Agile levy began on eighteenth February 2018 and this information goes directly back to that first day.

Import information is the value you pay for purchasing power.

* csv\_agile\_A\_Eastern\_England.csv
* csv\_agile\_B\_East\_Midlands.csv
* csv\_agile\_C\_London.csv
* csv\_agile\_D\_Merseyside\_and\_Northern\_Wales.csv
* csv\_agile\_E\_West\_Midlands.csv
* csv\_agile\_F\_North\_Eastern\_England.csv
* csv\_agile\_G\_North\_Western\_England.csv
* csv\_agile\_H\_Southern\_England.csv
* csv\_agile\_J\_South\_Eastern\_England.csv
* csv\_agile\_K\_Southern\_Wales.csv
* csv\_agile\_L\_South\_Western\_England.csv
* csv\_agile\_M\_Yorkshire.csv
* csv\_agile\_N\_Southern\_Scotland.csv
* csv\_agile\_P\_Northern\_Scotland.csv

**Download Agile Outgoing Tariff Pricing Data**

The Agile Outgoing duty began on fifteenth May 2019 and this information goes directly back to that first day.

Active information is the value Octopus will repay you for sending power to the lattice. i.e. through Solar or Wind age.

See the Outgoing Overview page for more information.

* csv\_agileoutgoing\_A\_Eastern\_England.csv
* csv\_agileoutgoing\_B\_East\_Midlands.csv
* csv\_agileoutgoing\_C\_London.csv
* csv\_agileoutgoing\_D\_Merseyside\_and\_Northern\_Wales.csv
* csv\_agileoutgoing\_E\_West\_Midlands.csv
* csv\_agileoutgoing\_F\_North\_Eastern\_England.csv
* csv\_agileoutgoing\_G\_North\_Western\_England.csv
* csv\_agileoutgoing\_H\_Southern\_England.csv
* csv\_agileoutgoing\_J\_South\_Eastern\_England.csv
* csv\_agileoutgoing\_K\_Southern\_Wales.csv
* csv\_agileoutgoing\_L\_South\_Western\_England.csv
* csv\_agileoutgoing\_M\_Yorkshire.csv
* csv\_agileoutgoing\_N\_Southern\_Scotland.csv
* csv\_agileoutgoing\_P\_Northern\_Scotland.csv