# Analysis

There are three data sets, each for every building, namely the electricity data set, water data set, and the gas data set collected from the University energy manager. Each dataset contains 48 columns of half-hourly readings being recorded and stored for later analysis. This chapter would discuss the analysis and preprocessing carried out before model training can be carried out. There are 4 steps that are necessary to complete the analysis process are; the cleaning/preprocess phase where the data is reformatted and any null values are removed, the decomposition phase where we take a closer look at the data and denote any trends occurring, checking if the data is stationary (without trends) or not which is expected in the case of this project, and finally prepare the dataset for training. The next subsections will go through analysing the building datasets and noting any findings.

# Queens Building Dataset

## Preprocess/Cleaning

A black screen with white text

Description automatically generated with low confidence

Queen’s Building Electricity Reading before cleaning

As the figure above shows, the raw data collected is recorded every 30 minutes from the assigned meter and is not comprehensive to understand, so a preprocessing is essential.

A picture containing table

Description automatically generated

Queen’s Building Electricity Reading after cleaning

After cleaning the data, it is now easy to work with and we can move to the next step in our analysis.

## Visualize

Visualization of the data can be achieved in various ways but to save on time we will call upon the statsmodel library which is used in carrying out statistical tests. To use the library, we must resample or data into a daily format giving us the figure.

A picture containing graphical user interface

Description automatically generated

Electricity Daily Dataset

From the figure above we can see that the library was able to display 4 different plots, the original data, the trend of the data, the seasonality, and the residuals (leftovers). The seasonality fails to get detected by the library; a further resampling is essential to observe the seasonal trend.

Graphical user interface

Description automatically generated

Electricity Monthly Dataset

From this new figure we can see a more defined seasonal trend and it is exhibiting a clear downward trend as the year increases. The seasonality also shows more usage in the winter and spring months and significantly less usage in the summer months.

## Check for stationarity

Using the above figure, we can see that our data is in fact non-stationary which increases the desired model’s predictive accuracy. Another way of testing is by performing an Augmented Dickey Fuller (ADF) test, a unit root test, which is a sort of statistical test. This test is conducted under certain conditions:

* Null Hypothesis: Data is non-stationary
* Alternate Hypothesis: Data is stationary

This test may show that the series is non-stationary if the null hypothesis fails to be rejected. The condition to reject the null hypothesis is:

* If the test statistic < critical value and p-value < 0.05

|  |  |  |  |
| --- | --- | --- | --- |
|  | Test Statistic | P-Value | Critical Values |
| Electricity | -2.983 | 0.036552 | 1% -> -3.432  5% -> -2.862  10% -> -2.567 |
| Gas | -5.183 | 0.000009 | 1% -> -3.432  5% -> -2.862  10% -> -2.567 |
| Water | -4.54 | 0.0002 | 1% -> -3.432  5% -> -2.862  10% -> -2.567 |

From the table we can see that all the datasets for the Queen’s Building shows that it is non-stationary (has trends & seasonality)

## Correlation

We need to see how the 3 datasets are correlated with each other and to achieve this we use a Pearson correlation function:

Where

r – correlation coefficient

- values of the x-variable in the data

- mean of the values of the x

– values of the y-variable in the data

– mean of the values in y

Chart, treemap chart

Description automatically generated

From the figure all datasets are positively correlated with the electricity and water datasets have the highest correlation. Further examining the correlation, we can see that this is not the case as the values never intersect but Electricity and Gas have the same pattern

Chart, line chart

Description automatically generated

Example showing correlation between Electricity and Water Datasets

Chart

Description automatically generated

Example showing correlation between Electricity and Gas Datasets

## Feature Importance

XGBoost has a feature importance function which determines how each feature contributes to the selected target variable during regression.

Target Variable – Electricity

Chart, bar chart

Description automatically generated

The two most important features are Gas and Water which is expected, and temperature plays an equal role in the electricity usage.

# Hugh Aston Building Dataset

## Preprocess/Cleaning

A black screen with white text

Description automatically generated with low confidence

Hugh Aston Electricity Reading Before Cleaning

A picture containing table

Description automatically generated

Hugh Aston Electricity Reading After Cleaning

## Visualize

Graphical user interface

Description automatically generated

Electricity Monthly Dataset

We can see from the figure above that there is a clear trend occurring. There was a downward trend happening from 2013 before picking back up in 2015 and maintaining that momentum till 2019 where it starts to drop again which might be an indicator of a better energy saving tactics employed in the building.

## Check for stationarity

|  |  |  |  |
| --- | --- | --- | --- |
|  | Test Statistic | P-Value | Critical Values |
| Electricity | -6.193937 | 0.000000 | 1% -> -3.432  5% -> -2.862  10% -> -2.567 |
| Gas | -5.006579 | 0.000022 | 1% -> -3.432  5% -> -2.862  10% -> -2.567 |
| Water | -6.168843 | 0.000000 | 1% -> -3.432  5% -> -2.862  10% -> -2.567 |

From the table we can see that all the datasets for the Hugh Aston Building shows that it is non-stationary (has trends & seasonality)

## Correlation

Chart, treemap chart

Description automatically generated

We see the same pattern occurring also within the Hugh Aston dataset as the Queen’s Building dataset where the highly correlated features do not show any correlation visually.

Chart

Description automatically generated

Electricity and Gas Correlation

As the electricity consumption rises so does the gas consumption which is expected during warm and cold months.

Chart, histogram

Description automatically generated

Electricity and Water Correlation

## Feature Importance

Chart, bar chart

Description automatically generated

By applying XGBoost regressors, the important feature of the dataset is shown above. Similarly with the previous dataset, Water and Gas features show to be greatly important to the Electricity consumption. The only difference is that the air pressure shows to be more important than the temperature which would need further examination.

# Gateway House Building Dataset

## Preprocess/Cleaning

A black screen with white text

Description automatically generated with low confidence

Gateway House Electricity Dataset before cleaning

A picture containing table

Description automatically generated

Gateway House Electricity Dataset after cleaning

## Visualize

A picture containing graphical user interface

Description automatically generated

Electricity Monthly Dataset

There is a very distinct downward trend occurring which shows that the Gateway House electricity consumption rates are increasingly getting better through excellent energy saving tactics. This shows that even without the pandemic happening the rate of consumption will always improve.

## Check for stationarity

|  |  |  |  |
| --- | --- | --- | --- |
|  | Test Statistic | P-Value | Critical Values |
| Electricity | -3.446605 | 0.009469 | 1% -> -3.432  5% -> -2.862  10% -> -2.567 |
| Gas | -4.618122 | 0.000120 | 1% -> -3.432  5% -> -2.862  10% -> -2.567 |
| Water | -5.722182 | 0.000001 | 1% -> -3.432  5% -> -2.862  10% -> -2.567 |

From the table we can see that all the datasets for the Gateway House Building shows that it is non-stationary (has trends & seasonality)

## Correlation

Chart, treemap chart

Description automatically generated

We see the same pattern occurring also within the Hugh Aston dataset as the Queen’s Building dataset where the highly correlated features do not show any correlation visually.

Chart, bar chart

Description automatically generated

At the start of the dataset the electricity and gas features seem to struggle but it started exhibiting normal behaviours afterwards till 2020 when the sudden drop in usage happened.

Chart, histogram

Description automatically generated

## Feature Importance

Chart, bar chart

Description automatically generated

By applying XGBoost regressors, the important feature of the dataset is shown above. Similarly with the previous dataset, Water and Gas features show to be greatly important to the Electricity consumption. This is extremely like the Queen’s Building dataset and shows that temperature impacts the rate of consumption.

# Weather Dataset

The analysis of the weather dataset has been broken into 3 sections; Preprocessing, Visualization and Feature Selection. The last section will be the final data which will then be added to our building data for training as regressor.

## Preprocessing

### View Data

Graphical user interface, text

Description automatically generated

From the above figure we can see that there are a lot of features that could be used and some showing null values. The next step is to remove any null and incomplete data.

### Remove null value

This can be achieved by using the built in dropna() function that is available in python and specifying the method of removal which in this case would be ‘any’ null values found in the columns. The resulting data then becomes:

Text

Description automatically generated with medium confidence

### Encode Categorical features

There are still some features to be discussed which are weather id, main, description and icon. These were found to be categorical data and so will be encoded. One of the features (weather\_id) had no sensible data that could be useful and was dropped from the table.

A screenshot of a computer screen

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated with low confidence

### Remove unnecessary features

This is the final stage of the weather dataset preprocess, and we took at look at the features that bore no significance to the data. By using the drop function, we selected the following features:

* Latitude and Longitude: these are just coordinates and do not serve well as regressors since it cannot be categorically encoded.
* dt: this is the time of data calculation which is a duplicate of the dt\_iso column that serves as the index.
* timezone: these are just shifts in seconds so there is no significance
* city\_name: this is a weather dataset of Leicester.
* clouds\_all: shows the percentage of cloudiness
* wind\_deg: wind direction has no effect on building energy consumption, so it was dropped.

Our final dataset before proceeding to the final step is now:

A screen shot of a computer

Description automatically generated with low confidence

## Visualization

We now must calculate the correlation between the features and plot it as a heatmap to show the feature importance. From the figure below we can see that there is a negative correlation between the humidity and temperature features. The features hardly intersect as shown below. We can see a high correlation between the dew point and temperature features. As the temperature rises, so does the dewpoint, making it somewhat similar

Chart, waterfall chart

Description automatically generated

Heatmap between features in Weather Dataset

Timeline

Description automatically generated with low confidence

Example showing the negative correlation between Temperature and Humidity

Graphical user interface, chart, line chart

Description automatically generated

Positive correlation between Temperature and Dewpoint

Below shows the final dataframe with the features and respective attributes. The count shows the total length of the dataset, mean, maximum value and the rest of the variables are also given. As shown the values are all different ranges which would make training less accurate and difficult.

Graphical user interface, text

Description automatically generated

## Feature Selection

For this project, the temp, dew\_point, pressure, humidity, and wind\_speed was chosen as the features that would be used in training and testing the algorithms. The decision on these features was they have the most relevance when putting energy consumption into consideration. For example, during cold days the electricity usage is expected to reduce and vice versa.

# Normalization

Feature scaling is the process of converting the values of different numeric objects to be within similar ranges. Scaling is used to prevent overfitting and biased results of supervised learning models. For example, if a model uses linear regression and the features are not scaled, some features end up having a higher impact than others, affecting the predictive performance. Therefore, it is important to scale your features. The only algorithms that do not need scaling are random forests and Decision trees since they are scaling invariant.

What is normalization? This refers to scaling features into specified ranges [0,1] in the case of min-max scaler. Normalization is important when the data is needed in bounded intervals. Below is a formular for normalizing based on the min-max scaler:

Where:

– represents the values

– minimum values

– maximum values

## Min-Max Scaler

Min-Max Scaler is a normalisation class from the sklearn.preprocessing package. The Min-Max Scaler estimator will fit on the training data set when normalising the training and test data sets, and the same estimator will be used to transform both the training and test data sets.

Text

Description automatically generated

## Standard Scaler

Unlike the min-max scaler that normalises the values around a certain range, the feature columns are centred at mean of 0 with a standard deviation of 1 using the standardisation approach, giving them the same properties as a typical normal distribution. It keeps relevant information about outliers and makes the algorithm less sensitive to them. The formular for standardization is:

Where:

– represents the values

– mean of values

– standard deviation of values

Text

Description automatically generated