**Appendix C - Group report front page**

You will include the following information on the first page of your group report.

**CMT307 Coursework 2 Group Project**

|  |  |
| --- | --- |
| **Group number** | G19 |
| **Project title** | Energy Usage Prediction |
| **Supervisor** | Hanzhi Wang |
|  | 21126107 MP  1840386 Matt  1823122 Ed  1823789 Ben  21088472 Bilal  21014978 Mingtao  21123124 Zainab  21099797 Joy |

1. Introduction

Nowadays, where global demand for energy increases and prices increase, there is also a strong need to reduce waste when possible. Thus, a prediction of energy usage is an important step that can allow humans to use energy more efficiently, and consequently more economically. This is crucial to reduce the negative consequences of global warming. A successful prediction of energy usage would reduce wastage, which is highly advantageous for energetic companies, governments bodies, and human beings.

Energy usage prediction is a great task for Machine Learning (ML), which can be seen as nonlinear time series with numerous complex factors. These include a variety of building stocks, weather changes, and periods of the year such as holidays. Also, there might be sudden changes in energy consumption due to unexpected events like equipment failure or blackout. Thus, it is extremely important to develop models which will have a wide exposure to various factors and will bring accurate results. Precise predictions will help us in achieving the following goals:

* Creating a reliable ML model that can effectively predict the use of energy in various categories of buildings in the following areas: hilled water, electric, hot water, and steam meters, and different weather conditions
* Gaining valuable insights into factors affecting a building’s energy demand, which allows managers to improve energy efficiency
* Enable managers to identify anomalously high/low energy consumption and alert them to problems with buildings
  1. Summary

2. Literature Review

The use of Machine Learning (ML) techniques in predicting future energy demands is a field that has been widely explored. Through various trials and studies, artificial neural networks (ANN) have been determined as one of the more effective techniques and are now readily used to produce accurate results (Seyedzadeh et al., 2019). After research, it has become clear that recurrent neural networks (RNN) are particularly efficient when using historic energy usage data as the input (Tun, Y.L et al., 2021). RNN’s loop-like structure produces a time delay, which is especially effective when utilizing temperature data (Sun, Y et al., 2020.).

Whilst RNN has been widely used within this field, it is also commonly acknowledged that the basic model of RNN has its limitations and drawbacks. Since we are interested in long term energy prediction as well as short term, a naïve RNN has a tendency to forget old information due to the commonly known vanishing gradient problem. To tackle this problem, we instead implement LSTM-RNN model (Berriel, R.F et al., 2017).

The LSTM-RNN model was introduced by Hochreiter and Schmidhuber (1997). In the LSTM model, the summation units of the RNN model are replaced by memory units, providing the LSTM model with the capacity to store and recall information for longer (Heidari, A et al., 2020). The LSTM model has been successfully implemented to forecast energy demands and produced accurate results, some examples of this include (Wang, J.Q et al., 2020) and (Rahman, A et al., 2018).

There was some literature that has made us aware of some of the potential drawbacks of using this model. (Rahman, A et al., 2018) found that the LSTM model assumes knowledge of future weather conditions and does not consider any potential changes in weather. Hence, should the weather differ significantly from our weather training data, there most probably be a loss in accuracy in the model. Secondly, there is a number of studies that noted difficulty in hyper-parameter tuning for this model. For example (Kim, T et al., 2019) noted it took a large amount of trial and error to find the optimal parameters. (Ding, Z et al., 2021) noted that it took a combination of trial and error, grid search, random search and Bayesian optimization in order for the optimal parameters to be found.

Thirdly, we decided to implement a decision tree model. There was a sufficient number of successful studies that give enough confidence in this model. However, from our readings we understood for the ease of use and the fact that decision trees are typically computationally inexpensive compared to other models, we may be giving up a small amount of performance (Amasyali, K et al., 2018). Based on our research we are under the impression that producing accurate results with LSTM-RNN and KNN, may be difficult and time-consuming thus it was right to potentially lose a small amount of accuracy, as this will allow to explore different techniques and produce further results for analysis and discussion. Nevertheless, there are still examples of decision tree models that produced accurate results. (Yu, Z et al., 2010) used a decision tree model for energy demand prediction in buildings and their model provided 92% accuracy. Also, (Tso, G.K et al., 2007) found that out of a neural network model, regression analysis and decision trees, it was in fact the simpler decision tree model that produced the best results. It is worth noting that this study was carried out in 2007 and hence there have been developments in machine learning techniques since then. Regardless, we are optimistic for all of our models and are looking forward to see the outcomes.

1. Description of the task and dataset

Dataset comes from Kaggle’s ASHRAE competition. It consists of readings from over 1,000 buildings over a three-year timeframe. The dataset consists of five CSV files. The building data file consists of 6 variables that provide information on buildings’ primary use, covered area, built year and floor count with a number of values ranging from 0 to 1448. Next, there are training and test data files for weather readings. They consist of 9 variables that provide information on air temperature, cloud coverage, dew temperature, precipitation depth, sea pressure, wind direction and wind speed. Lastly, there are two files, test and training both of which provide details on buildings meter readings. Overall there are thirty-four columns that have four data types: decimal, integer, date and string.

To measure the quality of developed models the following evaluation metrics: Root Mean Squared Logarithmic Error (RMSLE).

The RMSLE is calculated as

Where:

– RMSLE value(score)

n- number of observations in the public/private dataset

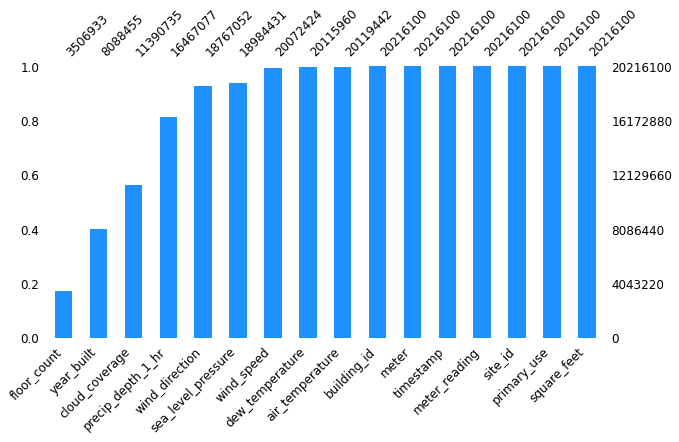
– prediction of target

– actual target for i

log(x) – natural logarithm of x

RMSLE was used because it is a common metric for regression problems. It is an extension of Mean Squared Error (MSE). However, for the energy prediction, RMSLE is better because it is more robust to the outliers. When a relative error is considered, also RMSLE is a more preferable choice because it considers the relative error between the Predicted and the Actual value and the scale of the error are not significant. On the other hand, the RMSE value Increases in magnitude if the scale of error increases. The most unique property that differentiates RMSLE from MSE is fact that the RMSLE penalizes the underestimation of the actual value more severely than it does for the Overestimation. preferable because the dataset has a wide range of target variables

To ease the analysis of the dataset, merging techniques were used. A merge was performed on three train data files and two test files to obtain a single test and training dataset. It is crucial to observe that nine out of sixteen variables have large missing values that need to be adjusted at a later stage. (Fig-1) In order to explore timely variations, we broke the timestamp variable into six new columns: hour, day, dayOfWeek, dayOfYear, month and year in order to explore timely variations in data.



**Figure 1-Missing Values in Training Data Set**

3.1 Exploratory Data Analysis

Once the data was combined we started exploring the target variable of meter reading. A log transformation of the variable was taken to adjust for high skewness and then plot a density graph that shows a good variation in values along with a high number of 0-meter reading values. Then an exploration of any seasonality changes was made by plotting meter readings against time.

Initial distribution of meter reading was explored. It is skewed to the left thus log transformation was applied to reduce the skewness.

Chart, histogram

Description automatically generatedGraphical user interface

Description automatically generated with medium confidence

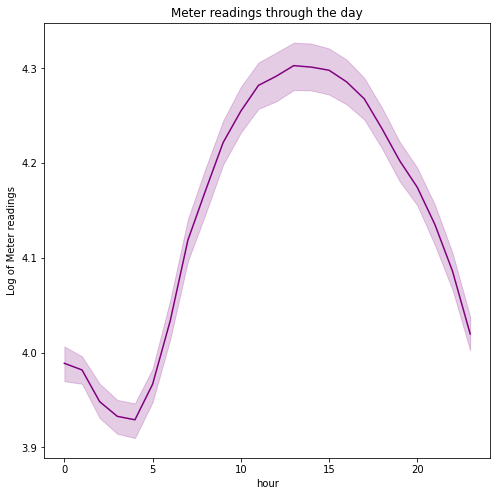


Figure 1 shows energy usage throughout the day. The energy usage is reasonably very low during the early morning hours and fairly high during the evening as it is the peak working time of all operating sites such as educational institutions and industrial areas.

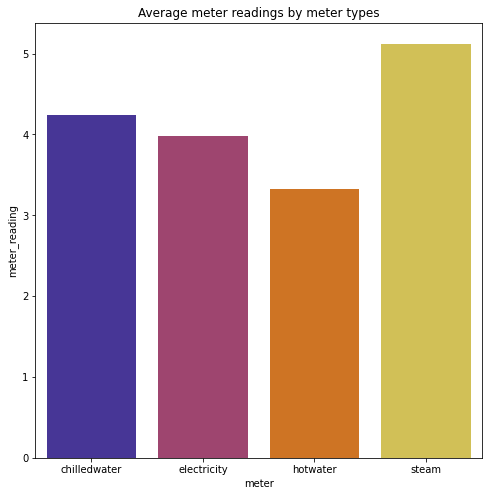
Chart, line chart

Description automatically generated

Figure 2 shows the energy usage per annum. It is observed that energy consumption is fairly low during the starting months of the year, rises sharply in the spring season and fluctuates during the summer season. June-September summer months report the highest level of energy consumption which may be due to the high AC usage in offices and institutions

Both bar plots show the distribution of meters at the building (fig.a) and energy consumption from each of these meter types(fig. b). Electricity meters are the most commonly used by chilled water and steam meter types. Also, steam and chilled water meter types consume the greatest energy followed by electricity type. It might be useful to replace hot water meter type with steam as it can greatly save energy.

Chart, bar chart

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface, chart, line chart

Description automatically generated

Since data assumes energy consumption in various types of buildings, it is worth exploring meter reading distribution based on primary usage in different areas. It is seen that educational institutions, offices and retail sites have the most energy consumption during the morning and evening time of the day and the least consumption during the night. This result is expected as these sites mostly have fixed operational timings. For entertainment and public assembly sites, there is low energy usage during night times and greater energy usage during evening times. For residential areas, there is a sharp decline after midnight and then meter reading keeps on increasing and reaches a high level and remains stable until midnight. This result is directly related to the higher level of activities being performed throughout the day in a house that utilizes various appliances. Also, an overall analysis of the graphs shows that utility, industrial, healthcare and food sales sites report higher levels of energy consumption whereas worship areas and retail sites consume a lower level of energy.

3.1.1 Analysis of weather data

Chart, histogram

Description automatically generatedChart, histogram

Description automatically generated

A density plot graph shows that air temperature variable follows a normal distribution. The mean value of air temperature is between 14 and 15 degree Celsius and most values lie between the range of 0-30 degrees. Cloud\_coverage is measured between a 0 to 9 scale where 0 means it is a clear sky and 9 means it is rainy. It can be observed from the density plot that most of the cloud coverage is zero. Sea level pressure follows a normal distribution with most values in the range of 1000-1025. Dew temperature has a skewed distribution with most values between 0-25 degrees

Chart, histogram

Description automatically generatedChart, histogram

Description automatically generated

Chart, radar chart

Description automatically generated.

This windrose diagram shows that for a majority of the building sites, the wind mostly blows from the north direction with its speed between 0 to 3.8 m/s., followed by then south direction. Also, the North East direction has the minimum wind pressure

3.2.2 Analysis of building data

Histogram

Description automatically generated with low confidenceChart, histogram

Description automatically generated

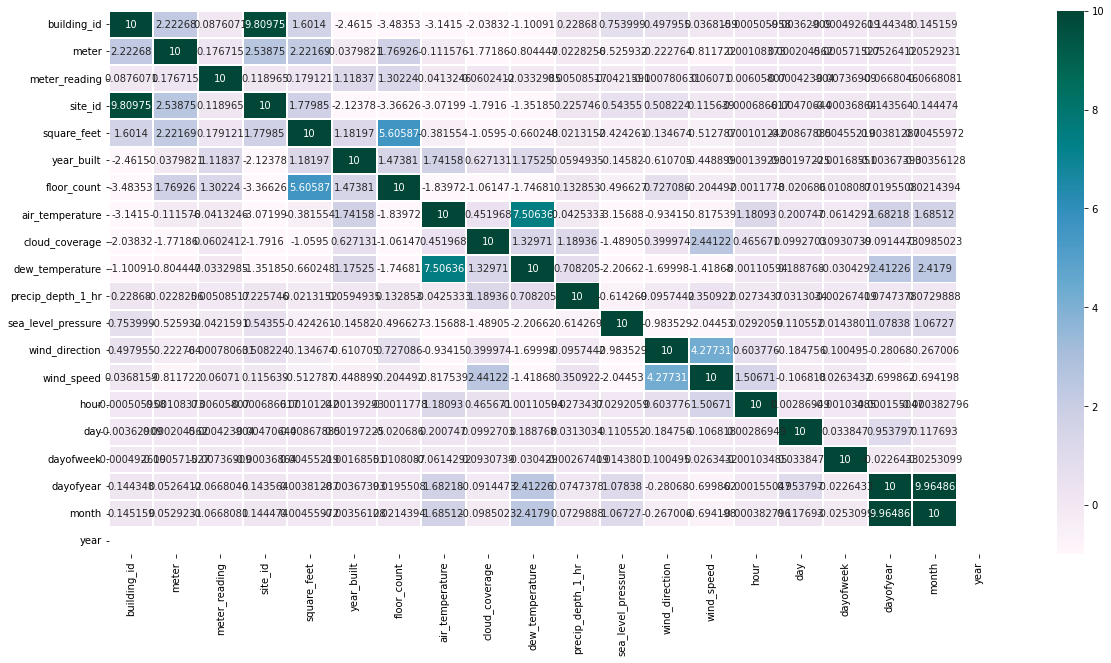
Chart, histogram

Description automatically generatedChart, line chart, histogram

Description automatically generated

It is clear that most of the buildings were built around 1975. The growing trend is maintained before 1980. Another growing trend is observed after 2000 with a peak around the 2010 year. Average meter readings greatly fluctuate with the year built variable with no clear trend. Such fluctuations were difficult describable and there is no valid rationale why they happened. Moreover, most of the buildings have 0-15000 covered square feet area and they have two floors. For a floor count greater than 10, meter reading has a drastic increase, then around 14 floors have a sharp decrease and then again for floor count greater than 15 there is a drastic increase.

3.2.2 Correlation Matrix



Correlation matrix was plotted to understand relationships between feature variables. We can conclude that Square\_feet and year\_built have a positive corelation with the target variable. The greater the size of the building, the more energy it consumes. Also, Air\_temperature is highly correlated with dew\_temperture and Square\_feet is correlated with floor\_count. However, most of the features are less in correlation with meter reading.

1. Methodology

**Lightgbm**: A gradient boosting framework, using tree-based learning algorithms. By using gradient based one side sampling and exclusive feature bundling, the lightgbm a very lightweight and fast algorithm. It is mainly implemented on large datasets due to it being prone to overfitting small datasets. Whilst a decision tree splits the tree tree-wise, the lightbgm algorithm splits the tree leaf-wise.

**RNN**: A recurring neural network is a form of machine learning that returns on itself to increase accuracy. For a typical neural network, firstly, an example from a dataset is loaded (in this case, the training data supplied). The network takes that example, and applies mathematical formulae to it using random variables, yielding a predicted result. Using the validation data, this prediction can be compared, and the difference between them will give an error. Returning the error back through the same path will adjust the variables, and this is all repeated until the variables are defined with minimal error. The recurrent neural network takes this a step further, by instead of taking in one example at a time and producing one result, it takes multiple neural networks which feed information to each other. This allows for a low time complexity and makes it suitable for dealing with large datasets.

**Decision Tree**: It uses rules learnt during its training phase to make decisions. You begin with the root node, which splits off into two different regions. These regions also split into further two different regions and this process continues. The decision tree uses the rules supplied to decide which region to go to at a node, and this process is continued until all the rules are applied or until there are no data points left. The decision tree time complexity is of form O (n log(n) \* d), where d is dimensionality in the data. A random forest uses multiple decision trees with an element of randomness for more accuracy, however, due to the multiple decision trees the computational memory cost is significantly higher than that of a single decision tree.

1. Experimental Setting

As part of the pre-processing, one of the main factors was memory usage due to the large nature of the data. Converting the data to feather files and applying a function to reduce the memory usage (mainly by converting int64s to int8s, or float64s to float32s) allowed to run the programme more efficiently with very little loss of data. The exact method is provided in the “Memory\_Management.py” file. The data was also merged with the weather data and the building data to allow for more features to be selected, and the timestamp was broken down into hours, days, weekdays, day of week, and month which allows us to get the exact date in integer values and based on our early data analysis, we found that the energy usages vary based on this features (See figure) . A log transformation was also applied to the meter reading and square feet, which was done to account for the large range of values and make the training of the model more effective. The feature selection allowed us to remove various features from the data to acquire a better result from the data.

5.1 Data Preprocessing & Feature Selection

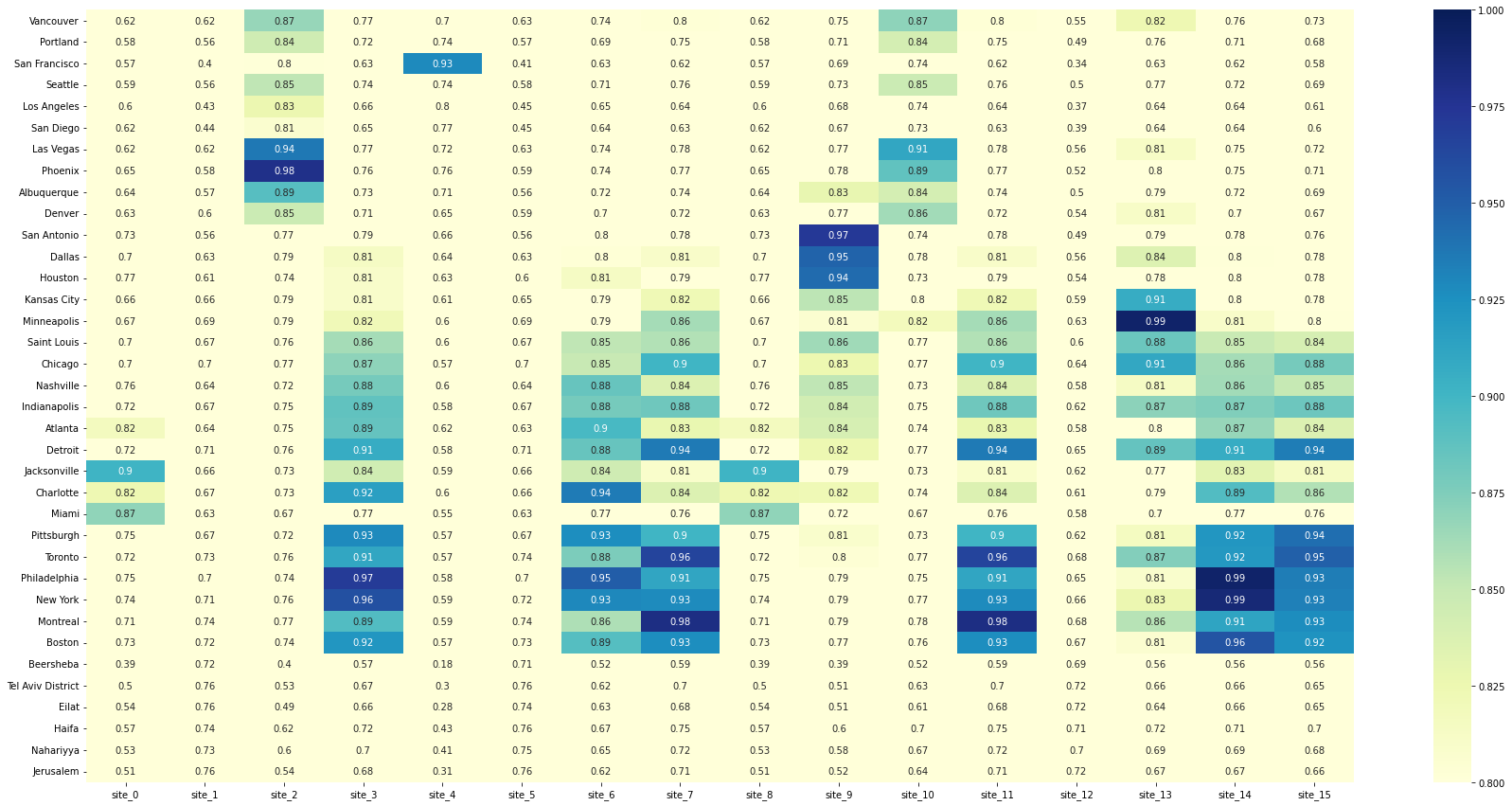
Various Data Preprocessing and feature selection with feature engineering procedure were followed to clean the data and feed it into the machine learning algorithms. It was observed by deep diving into the data that there can be many combinations of data cleaning that can be done and the models can be fed with the preprocessed data using similar model parameters to get higher accuracy, which is in this case is an RMSLE (lower is better).

5.1.1 Site Analysis

In the weather data it was observed that at 16 sites, an hourly weather report was provided, but the site id has been encoded with integer value ranging from 0 to 15. Using an external weather report from the Kaggle dataset - “Historical Hourly Weather Data 2012-2017”, it was imperative that the site’s can be divided into zones, based on their geographical location and time can be adjusted as well. It was important as it was suggested from the initial data analysis that the energy utilization depends highly on the topographical season and on the hour of the day. The technique that was focused on the analysis is to match the temperature from the historic data with the weather data. It is to be noted that the longitude and latitude with which the site id’s matched with the historical data is not precise but was close enough to make a conclusion.

The working python codes for this analysis is provided in the “site\_analysis.py” file.

A spearman correlation plot was made between the temperature on particular dates and hours, between the Historic data and the weather data.



Based on this analysis, and using the hour offset (offset.hour()) method from pandas library, the date has been changed according to the local timezone. Also, a feature (Flag) “IsHoliday” was created as it was imperative that certain commmercial building will consume less energy during holidays and weekend compared to non-holiday weekdays, for this particular task a method was created using Holidays library imported inside the python workspace.

5.1.2 Feature Creation and Outlier Treatment

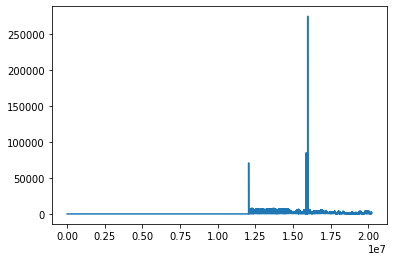
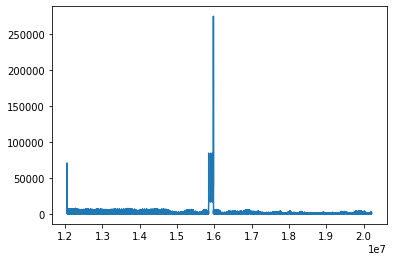
To remove the correlation between “floor\_count” and “square\_feet”, a new feature “floor\_area” was created by dividing “square\_feet” by “floor\_count”.

To train the model based on “site\_id” and “meter” (which is the type of meter in the original dataset), a new feature “groupNumTrain” was created, using this feature the dataset was divided into 60 batches and 60 models were made.

Each model was then fitted and predicted on the 60 batches of test data, this method helped by using lesser memory during both training and prediction of some models.

Since all the electricity meter reading is zero until May 20 2016 for site id = 0 and building id <= 104, this particular set of data was removed from training.

All the corresponding rows of building id which have zero meters reading from the start to a certain date were removed. Figure (x) shows the difference between before and after removal of zero meter reading, from building id 954.



5.1.3 Site-0 Correction

Due to a miscommunication, the electric meter readings for site 0 were not properly converted to units of kWh and are in kBTU. For models that are sensitive to units/absolute values, the issue can be avoided with the right conversion factors. Multiply by 0.2931 to get to model inputs into kWh like the other sites, and 3.4118 to get back to kBTU for scoring. (Dane, 2019)

5.1.4 Preprocessing Date and Time

As it was pretty evident from the initial data analysis that the meter reading changes with Date and time, hence the data and time has been converted into hour, day, weekday, month and day of the week.

5.1.5 Missing Value Imputation

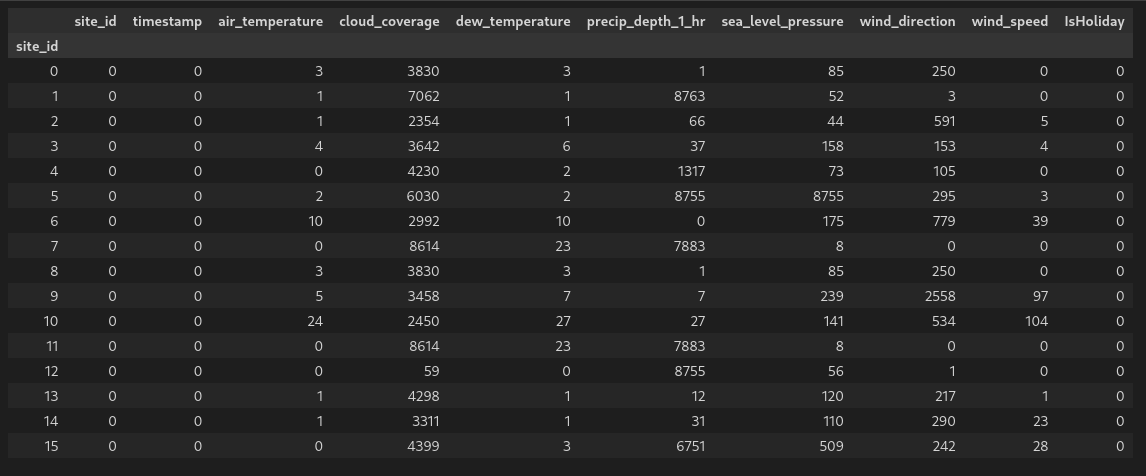
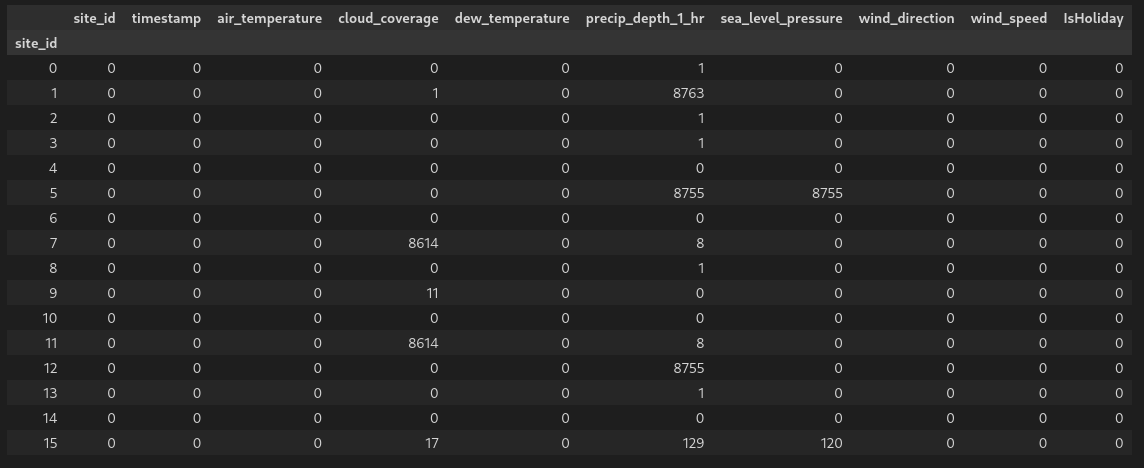
The figure shows that the weather data has many missing values, interpolate method from the pandas was used with forwarding fill implementation to remove as much missing data as possible.

Figure below shows the number of missing values in each site.

Some site\_id have high missing values in some features, because some property has never appeared in specific site\_id, and missing values remains for these features.

5.1.6 Adding Lag Features.

As we have time series kind of data, it has been imperative to add lag features in our data. A lag features is a fancy name for a variable which contains data from prior time steps. If we have time-series data, we can convert it into rows. Every row contains data about one observation and includes all previous occurrences of that observation. (mikulskibartosz, 2019)

Lag features has been added to air\_temperature, cloud\_coverage, dew\_temperature, precip\_depth\_1\_hr, sea\_level\_pressure, wind\_direction, wind\_speed.

5.1.7 Encoding and Counting

Two features bid\_cnt and year\_cnt has been introduced by encoding and counting categorical feature, building\_id and year\_built, this feature has proved to be useful and helped lower the RMSLE in LightGBM modelling.

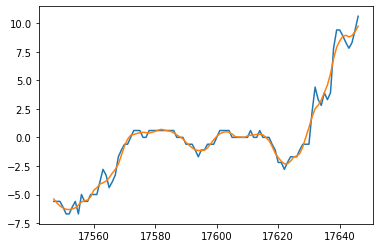
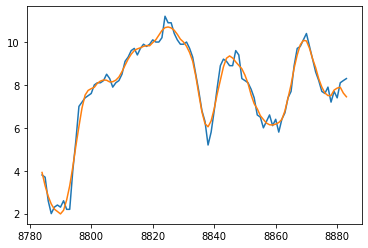
In general, encoding is an important technique in machine learning as many algortihms cannot take in categorical feature, and technques such as one-hot encoding increase the shape of the dataset, using is again memory ineffcient while training large sets of data. (SagarDhandare, 2022)

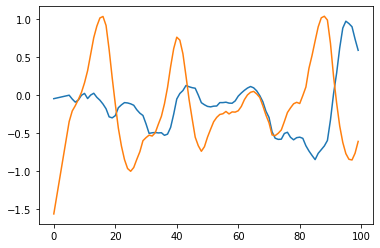
5.1.8 Smoothing Filter for Weather Data

It is also imperative to smoothen the data to remove noises from it, as the weather data is time sensitive, Savitzky-Golay filter has been used to smoothen the data.

Smoothing also reveals the trend and patterns in the data. (Ridolfi, n.d.)

Below figure reveals trend and remove noise from air\_temperature and dew\_temperature and reveals the sine inverse proportionality between air\_temperature and dew\_temperature.





5.2 Decision Trees

For this model, the log transformation was not used as converting the raw train meter readings to integer values instead produced higher accuracy rates. However, to achieve accurate results it also required a large enough maximum depth which, in turn, increases the complexity, and therefore memory usage of the model exponentially.

Chart, line chart

Description automatically generatedWith these adjustments, the hyperparameters that were found to have the greatest impact were the random state and max depth of the trees. To tune these hyperparameters, a small sample from the train data was looped and the highest accuracy values for these were found.

As expected, the random state showed a random distribution of accuracy values, and therefore the highest point the most accurate value that was tested was used. This hyperparameter is especially valuable when considering that the random state of the value has very, negative effects on memory or performance of the model.

Chart, line chart

Description automatically generated

Figure N shows an inverse logarithmic shape, with it plateauing at roughly 40. Therefore, any value above this would cause the model to be unnecessarily complex and cause performance issues. However, the memory requirements of the max depth value of 40 was far too great for our computational resources. The value of max depth 14 was the maximum possible with the computational power available and was therefore the compromise used.

1. Error Analysis

Errors to test

R2 SCORE and RMSLE

RNN:

Epochs 20 :

Training RMSLE = 1.10

valifysadauypojn RMSLE = 1.10

Kaggle = 2.33 R2 = 0.60

Epochs 50 :

Training RMSLE = 1.07

val;idatijhn RMSLE = 1.10

R2 = 0.61

Epochs 10:

Training RMSLE = 1.11

validation RMSLE = 1.11

R2 = 0.59

Decision Tree:

Max depth 14 & random state 1 & 75% percent of training data used

Kaggle = 2.291 public

6.1 Analysis

In summary, most methods produce predictions with high levels of accuracy, with the worst being LSTM with an RMSLE of 5.616, and the best being Lightbgm with an RMSLE of 1.067. For some of the models, such as the decision tree model, there were certain performance and memory issues (Due to time complexity of O (m n^2), where m is the size of the data and n is the number of layers) that prevented them from producing nearly as accurately as they could and extra hyperparameter tuning could have been performed. Whilst the RNN model is not as accurate as Lightbgm, the short running time and low memory usage makes it an effective model to use if speed is a priority. As the two best models were the neural networks, it can be concluded they were the most effective method for this task.

In terms of errors, the greatest potential for errors within this project was the dataset itself. Some columns within the training dataset had to be removed entirely (Columns with greater than 50% missing data). Columns with less than 50% missing data had the missing data points replaced with the median of that column. Whilst it would be ideal if all the data was available, by keeping as many columns as possible, this led to optimal feature selection to enhance the accuracy of the neural network models.

Each model has its own weaknesses that could also cause errors. LSTM and Lightbgm are prone to overfitting and RNN frequently have gradient vanishing and exploding problems. Decision trees are unstable meaning that if there is a small change in the training data, the optimal tree may have large changes.

Sources:

<https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/>

Jiang Su and Harry Zhang, A Fast Decision Tree Learning Algorithm, University of New Brunswick, NB, Canada

|  |  |
| --- | --- |
| **Model** | **RMSLE (From Kaggle Submission)** |
| Decision tree | 2.391 |
| Lightbgm | 1.067 |
| RNN | 2.069 |
| LSTM | 5.616 |
| Baseline Average | 3.069 |