Group 4 Assignment 2 Report

BY

Sining Bao
Benjamin Wang
Pamela Yang

DATA0006
The University of Melbourne

10 December 2023

ACKNOWLEDGEMENT

We express our sincere gratitude to Dr. Geela Chee for her invaluable guidance and educational support in the realm of Python data analytics over the past 12 weeks. Additionally, we extend our appreciation to Mr. Daniel Pham for his facilitation of workshops and his insightful responses to our queries. Their expertise and commitment greatly contributed to our learning experience.

Table of Content

1. Executive Summary **4**
2. Introduction **5**
3. Data Collection **6**
4. Exploratory Data Analysis (EDA) **6**
5. Data Cleaning and Preprocessing **9**
Pre-Processing: Scalers 10
Preprocessing: Imputation 10
Preprocessing - Indexing 11
6. Methodology **12**
Regression Analyses 12
K Nearest Neighbour Analysis 12
7. Analysis and Results **13**
Linear Regression 13
Decision Tree Regression Analysis 14
KNN Analysis 15
8. Discussion **17**
9. Conclusion **17**
10. Visualizations and Tables **18**

1. Executive Summary

This report highlights the process and outcomes of efforts by Sining Bao, Benjamin Wang and Pamela Yang to model and predict the Maximum Daily Energy Demand for the state of Victoria, Australia.

It was hypothesized that temperature and humidity would have the highest correlation with the utilization of Heating, Ventilation and Air Conditioning which, in turn, accounts for approximately between 20-50% of a building's energy use.[1]

Analysis was conducted on weather and energy data collected from between November 2022 and April 2023 to determine the relationship between temperature, humidity, wind speed, rainfall, air pressure and energy demand.

To validate the effectiveness of various analytical and modeling techniques, three key processes for predictive modeling were undertaken:

1. Linear Regression Modeling
2. Decision Tree Regression Modeling
3. K Nearest Neighbour Modeling

The study concludes that regression modeling encountered difficulties attributed to the absence of a linear correlation between weather patterns and energy demand, leading to inconsistent accuracy. In contrast, the K-Nearest Neighbor technique demonstrated relatively precise modeling, particularly in relation to maximum daily temperature and relative humidity.

However, it is highly recommended that additional data collection into other potential correlatory parameters, such as sunshine or evaporation, would be highly beneficial in enabling a greater degree of certainty in predicting daily maximum energy demand.

2. Introduction

In light of increased population growth within Victoria, and the general transition to renewable energy, the impact of environmental factors on maximum daily energy demand has become a highly pertinent point of discussion[2].

The aim of this study was to examine the effect of temperature, alongside other environmental factors in predicting the maximum daily energy use of Victorians, based on data extracted from the Australian Energy Market Operator[3].

[1] https://www.energy.gov.au/households/heating-and-cooling
[2]https://www.energy.gov.au/sites/default/files/Australian%20Energy%20Statistics%202021%20Energy%20Update%20Report.pdf
[3]https://aemo.com.au/energy-systems/electricity/national-electricity-market-nem/nem-forecasting-and-planning/forecasting-and-planning-data/nem-electricity-demand-forecasts/2017-electricity-forecasting-insights/summary-forecasts/maximum-and-minimum-demand

This objective was predominantly accomplished through the utilization of Python predictive modeling techniques based on Victorian weather and energy data collected between November 2022 and April 2023.

Data cleaning, preprocessing and both regression and classification predictive modeling methodologies were employed in an effort to effectively utilize input data, and accurately model maximum daily energy demand.

This study focused on the relationship between weather and environmental factors purely on energy usage, where additional factors such as energy pricing were excluded.

3. Data Collection

This study incorporates two datasets for analysis:

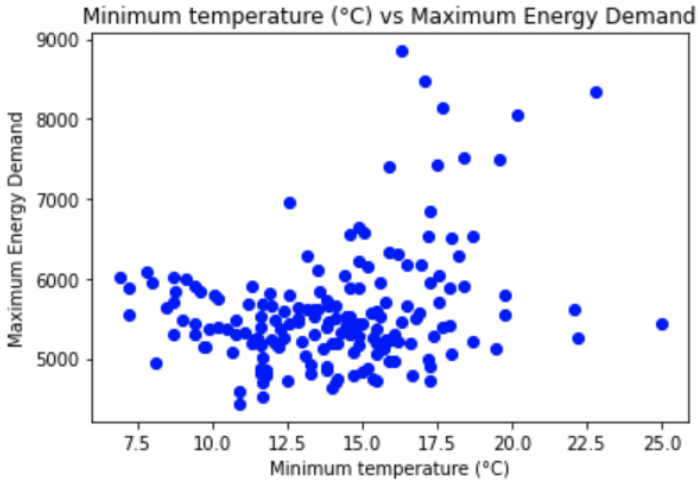
- weather.csv
  This file encompasses twenty essential weather indicators recorded daily for the city of Melbourne spanning the period from November 2022 to April 2023. The data has been extracted from the Bureau of Meteorology and collated into a single file. There were 22 weather variables collected within this file.
- price_and_demand.csv
  Within this file, one can find energy price and demand data for the state of Victoria, captured at half-hour intervals from November 2022 to April 2023. The information has been sourced from the Australian Energy Market Operator.

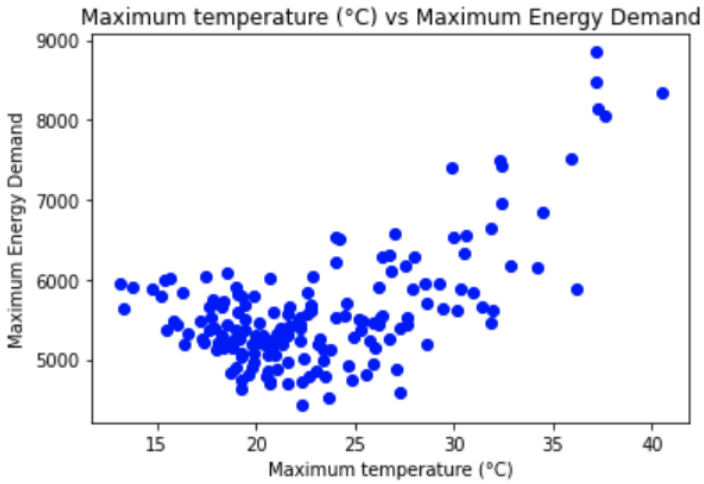
4. Exploratory Data Analysis (EDA)

Exploratory Data Analysis was conducted primarily through Multivariate Graphical Analysis based upon background hypotheses regarding energy consumption.

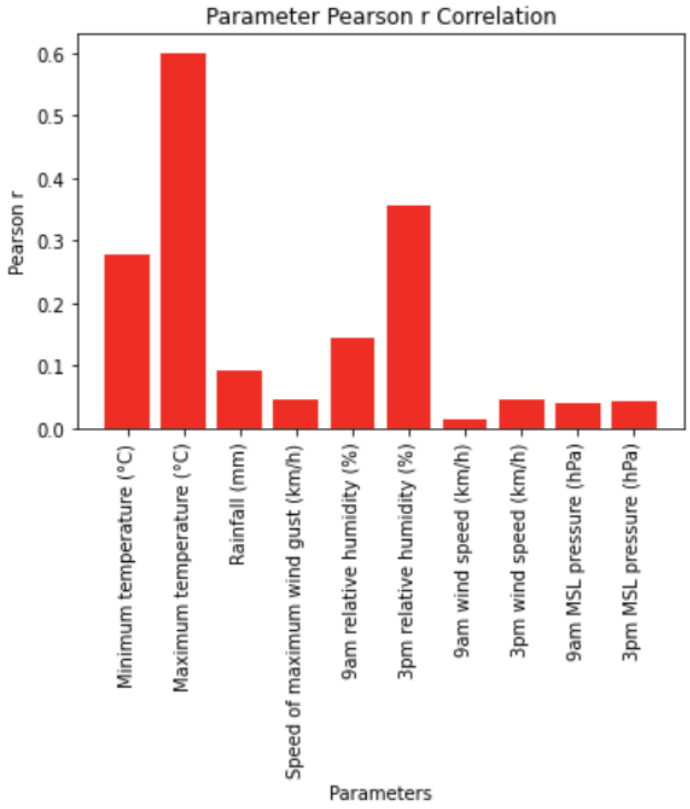
The primary hypothesis was based on the background that most industries' primary drivers for energy demand were Heating, Ventilation, and Air Conditioning (HVAC) equipment. HVAC demand, in turn, is hypothesized to be driven by temperature increases.

This was supported by a visual inspection of the correlation between both Minimum and Maximum Daily Temperature and Maximum Daily Energy Demand (see below).


Minimum temperature (°C) vs Maximum Energy Demand
Maximum Energy Demand
9000
8000
7000
6000
5000
7.5
10.0
12.5
15.0
17.5
20.0
22.5
25.0
Minimum temperature (°C)


Maximum temperature (°C) vs Maximum Energy Demand
Maximum Energy Demand
9000
8000
7000
6000
5000
15
20
25
30
35
40
Maximum temperature (°C)

The Pearson *r* coefficient was calculated for all features with respect to Maximum Energy Demand. This enabled the identification of key variables with the greatest degree of



correlation for predicting and modeling.

As illustrated (see above), “Maximum Temperature” was noted to be the most linearly correlated with Maximum Energy Demand. For the purposes of Regression Modelling, the Pearson Correlation coefficient, therefore, also served as a prioritizer of independent / predictor variables to be passed through each regression model.

5. Data Cleaning and Preprocessing

The predominant purpose of data cleaning the input data sets has been to ensure that data can be consistently passed through relevant models and processed for predictive purposes. Concurrently, data cleaning and preprocessing were also conducted with the aim of reducing any potential bias caused by excessive processes.

The majority of data cleaning and preprocessing was conducted on the "Weather" input data.

For the Energy Pricing and Demand dataset, it was deemed that minimal data cleaning and processing was required, given that both target variables, "Total Demand" and "RRP", were float data types.

Similarly, there were no missing data points detected when utilizing the *.isna()* method.

```
# Check datatype of the dataframe
price_and_demand.dtypes
```

```
REGION             object
SETTLEMENTDATE     object
TOTALDEMAND       float64
RRP               float64
PERIODTYPE         object
dtype: object
```

```
# check if there is any mising data
price_and_demand.isna().sum()
```

```
REGION            0
SETTLEMENTDATE    0
TOTALDEMAND       0
RRP               0
PERIODTYPE        0
dtype: int64
```

Data Cleaning: Deletion of Irrelevant or Missing Data

The preliminary data cleaning method employed with respect to the "Weather" input data has been the removal of missing data and irrelevant data.

Several features were noted to be completely empty in the original csv file, making their value to predictive modeling functionally irrelevant. As such, deletion has been applied to entirely remove the "Evaporation (mm)", "Sunshine (mm)" and "9am cloud amount (oktas)" features. This has been completed through the *.dropna* method for features or columns with entirely empty datasets.

Subsequently, listwise deletion has also been applied to certain data points with predominantly incomplete or missing data to ensure that complete variable data sets were

analyzed for correlation purposes. This, most notably, has been applied to the weather data point pertaining to April 24, 2024, in which 13 of 22 features were empty. The *.isnull* method has been applied to remove data points where a desired feature may return an empty value. As such, it only applies to empty data points when a desired feature has been selected.

Due to the target variable for this study being *maximum* daily energy demand, it was also deemed that the only required data for modeling was the maximum total demand recorded each day.

```
# Calculate maximum daily energy demand and average rrp
max_demand = price_and_demand.groupby("Date")['TOTALDEMAND'].max()
print((max_demand))
```

As a result, the maximum demand was derived by grouping total demand by date.

Pre-Processing: Scalers

Additional preprocessing methods were utilized for regression analysis through the implementation of scalers to normalize the datasets across a mean of 0 and a standard deviation of 1 to ensure that the data was not distorted by units of measurement and scale.

```
# Preprocessing with Standardization with Mean 0, Std 1
scaler = preprocessing.StandardScaler().fit(nrg_X_train)
nrg_X_train = scaler.transform(nrg_X_train)
nrg_X_test = scaler.transform(nrg_X_test)
```

Preprocessing: Imputation

The removal of missing or irrelevant data was completed only in situations where the majority of its relevant values were missing. This was done so in preference over the application of mean imputation to minimize the potential for distortion to the standard deviation.

Functionally, for the weather dataset, imputation was essentially irrelevant as listwise deletion removed all empty data points.

Data Cleaning: Data Type Conversion

Additional steps were taken to ensure all values were numerical float values. This process has been conducted for the purpose of ensuring that all values are suitable for analysis and modeling. As most features were already float values (see below), a viable data type for predictive analysis and modeling, the remaining ‘object’ variables could easily be utilized for future modeling.

```
weather.dtypes
```

```
Location                          object
Date                              object
Minimum temperature (°C)         float64
Maximum temperature (°C)         float64
Rainfall (mm)                    float64
Evaporation (mm)                 float64
Sunshine (hours)                 float64
Direction of maximum wind gust    object
Speed of maximum wind gust (km/h) float64
Time of maximum wind gust         object
9am Temperature (°C)             float64
9am relative humidity (%)          int64
9am cloud amount (oktas)         float64
9am wind direction                object
9am wind speed (km/h)             object
9am MSL pressure (hPa)           float64
3pm Temperature (°C)             float64
3pm relative humidity (%)        float64
3pm cloud amount (oktas)         float64
3pm wind direction                object
3pm wind speed (km/h)            float64
3pm MSL pressure (hPa)           float64
dtype: object
```

All relevant features were also converted to numerical quantities for the same purpose. An example of this is the conversion of "Direction of maximum wind gust" to its respective True Bearing Angle (see below).

```
#Replacing all wind directions with True Bearing quantities
weather_bearings = weather_nocalm.replace(['N','NNE','NE','ENE','E','ESE','SE','SSE','S','SSW','SW','WSW','W','WNW','NW','NNW'],
                                          [0,22.5,45,67.5,90,112.5,135,157.5,180,202.5,225,247.5,270,292.5,315,337.5])
```

This was done to further ensure that numerical data could be effectively passed through any predictive model based on numerical inputs.

Preprocessing - Indexing

A secondary preprocessing approach utilized for the price and demand dataset was the selection and formatting of dates for all data points.

```
# Convert "SETTLEMENTDATE" column to datetime
price_and_demand["SETTLEMENTDATE"] = pd.to_datetime(price_and_demand["SETTLEMENTDATE"], format="%d/%m/%Y %H:%M")
```

```
# Extract date from "SETTLEMENTDATE" and create a new 'Date' column
price_and_demand['Date'] = price_and_demand['SETTLEMENTDATE'].dt.date
```

This process was also conducted on the energy demand dataset to ensure that the format of indices was consistent across both to enable modeling to effectively correlate and link data.

6. Methodology

In order to develop a model which effectively predicts maximum daily energy use, This study applies Linear Regression, Decision Tree Regression and K Nearest Neighbour algorithm to analyse, train and test the data.

Regression Analyses

In light of the continuous nature of the target feature, which pertains to the maximum energy demand, a comprehensive analysis employing multiple Regression techniques has been conducted.

Linear Regression was chosen due to the numerical dependent variable and multiple numerical independent variables, and due to initial exploratory data analysis indicating some degree of linearity between Maximum Temperature and Energy Demand.

Decision Tree Regression was selected for its similar capability to model complex relationships but without presupposing linearity.

R square and Mean Standard Error were employed to assess the effectiveness of both the Regression and Decision Tree Regression model.

K Nearest Neighbour Analysis

The K-Nearest Neighbors (KNN) algorithm has been chosen for this project due to its applicability to both classification and regression tasks, coupled with its advantageous property of making no assumptions about the underlying distribution of the data. The study endeavors to categorize the target variable into discrete bins, exploring variations in bin size, the number of neighbors, and the train-test proportion for enhanced prediction outcomes. Additionally, the chi-square feature selection method has been employed to understand and select features that demonstrate optimal performance in the predictive modeling process.

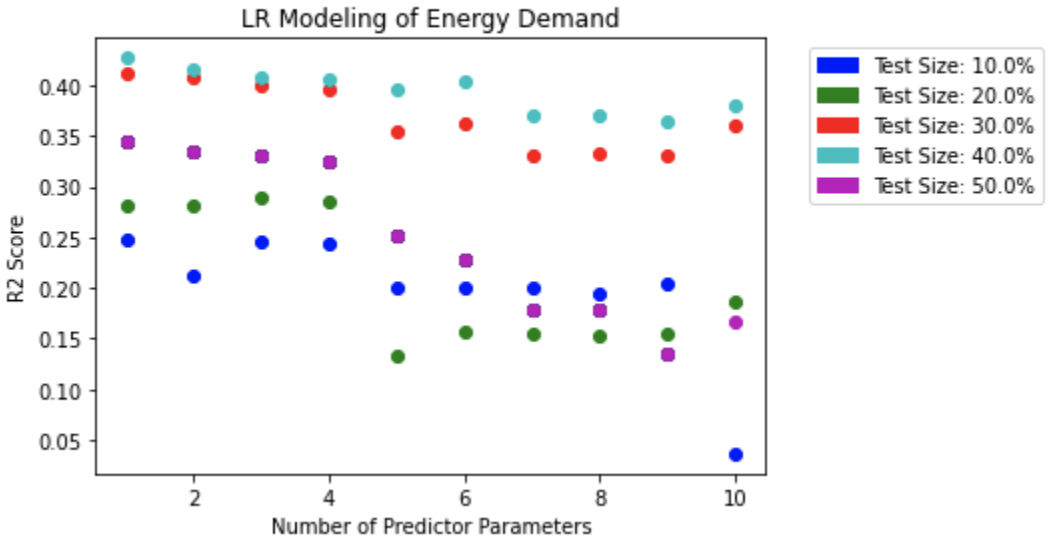
7. Analysis and Results

Linear Regression

Number of Predictor Variables vs Linear Regression R^2 and MSE

Based on the ranking of ten predictor variables from highest Pearson *r* coefficients to lowest, the r^2 score of each Linear Regression model was plotted against a variable number of predictor variables, alongside a variation in test size to determine the most 'ideal' and highest r^2 score for a given combination of Test Size and Input Parameters.

```
LR Ideal Paramaters: 1
Ideal Test Size: 0.4
R2: 0.42716155558021385
```



Based on this modeling, it can be seen (above) that a test size of 40% alongside a single predictor variable yielded the highest r^2 score.

This combination of input factors yielded an r^2 score of approximately 42% and a Mean Squared Error 526.

It should be noted that multiple reruns of this model yielded consistent results with minimal variability in R-Squared and output parameters.

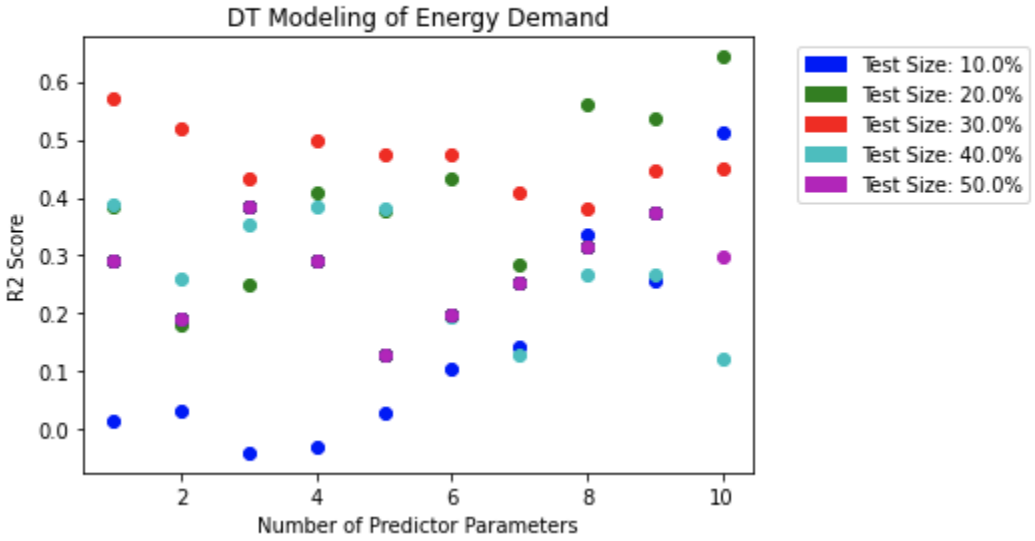
Decision Tree Regression Analysis

Number of Predictor Variables vs Decision Tree Regression R^2 and MSE

Similarly to the approach conducted with Linear Regression, the r^2 score of each Decision Tree Regression model was plotted against the number of predictor variables inputted, with a new series plotted for each test size.

The quantity of parameters that yielded the greatest R^2 score appeared to be a higher quantity of predictor variables, alongside a test size proportion of 20%.

```
DT Ideal Paramaters: 10
Ideal Test Size: 0.2
R2: 0.6426249746342441
```

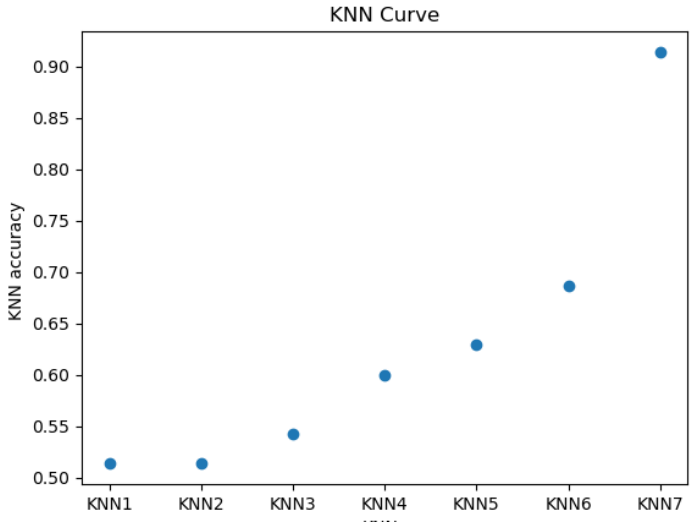


This combination of input factors yielded an r^2 score of approximately 60% and a Mean Squared Error of approximately 400.

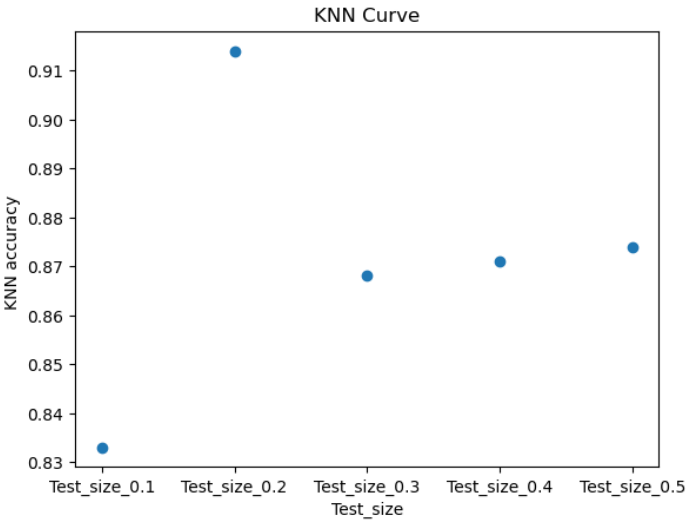
It should be noted that multiple reruns of this model yielded a high degree of variability in results, with R-Squared noted to fluctuate between 45-65% and MSE noted to fluctuate between 400 and 600.

KNN Analysis

The fluctuations in KNN accuracy stem from the diverse intervals chosen for numerical categorization. Altering these intervals results in distinct KNN accuracy values. As delineated below, numerous iterations have been undertaken with the aim of attaining an optimized accuracy score. This iterative process underscores the meticulous efforts invested in refining the model for enhanced predictive performance.

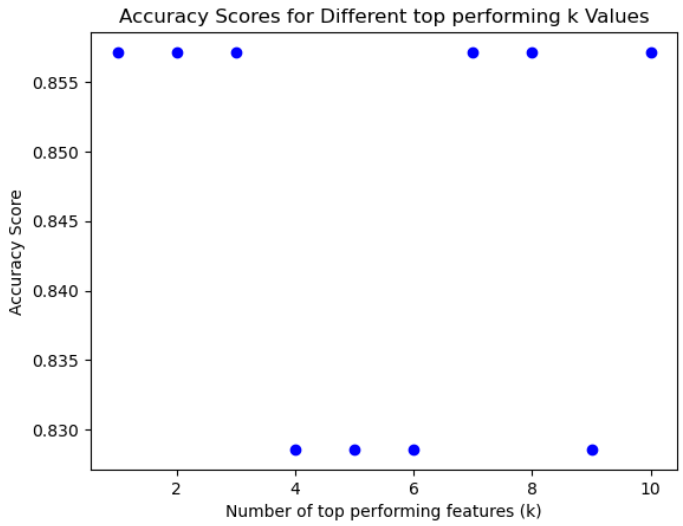


During the course of the prediction process, it has been noted that alterations on the train-test size also has influence towards KNN accuracy. When maintaining the same category, the modification of the test set size yields the subsequent result.



The above experimental procedure has revealed that the utilization of bin7, characterized by intervals [min, 6000), [6000, 7000), [7000, max] produces the highest accuracy score. Furthermore, it has been determined that the most accurate train-test split configuration is achieved with an 80%-20% division.

Utilizing the optimized binning and train-test split determined above, and to delve deeper into the KNN modeling while progressing to the feature selection phase, this study incorporates the chi-square feature selection method. This selection method is chosen for its capability to handle both numerical and categorical data seamlessly.



In interpreting the optimal number of features and their corresponding accuracy scores identified above, the results reveal a relatively stable performance across various k values. Notable, for k=3 and k=7, the accuracy scores peak at 0.857. However, it is noticeable that these points represent critical thresholds, as the inclusion or exclusion of a single feature results in a decrement of the accuracy score.

To enhance robustness and provide a cross-validated estimation, the K-fold method was implemented for both the top three (k=3) and top seven performing features (k=7). The outcome shows that the top three features have a higher accuracy score (0.845) compared to the top seven features (0.804).

Combined with the observation that the accuracy scores for k=1, k=2, and k=3 are uniformly high, and the scores exhibit a consistent stability for other values, this suggests that opting for the top three best performing features, namely maximum temperature, rainfall, and 3pm relative humidity, suffices for the KNN model.

8. Discussion

The findings of this study indicate that, in comparison to linear regression, decision tree regression, and K-nearest neighbor modeling, Despite the conventional understanding that maximum energy demand is a continuous dataset and that regression algorithms would theoretically yield superior results, the discrete KNN algorithm demonstrates better performance, achieving an accuracy rate of over 80% in predictions. In contrast, both linear and decision tree algorithms yield suboptimal results, characterised by low to moderate R-squared values and high mean square error.
Further exploration of the KNN algorithm, incorporating k-fold cross-validation and chi-square feature selection, reveals that among the key weather indicators considered, namely maximum temperature, rainfall, and 3pm relative humidity, these three features exert the most significant influence on energy use prediction.

Limitation of study

- Feature Limitation: Incomplete data poses a constraint on the inclusion of certain crucial predictors. Notably, variables such as sunshine and evaporation, which could potentially contribute to energy use prediction, are excluded from the analysis due to their unavailability in the dataframe.
- Due to the tight time constraint applied to the scope of this project, additional time to support more thorough analysis and evaluation of analytical and predictive methods may have provided a greater degree of accuracy and precision to the final output model.
- A greater sample size with a longer scope of time may enable a greater understanding of parameter correlation and, in turn, a clearer view on which predictive modeling tools would have been most appropriate for predicting energy demand.

9. Conclusion

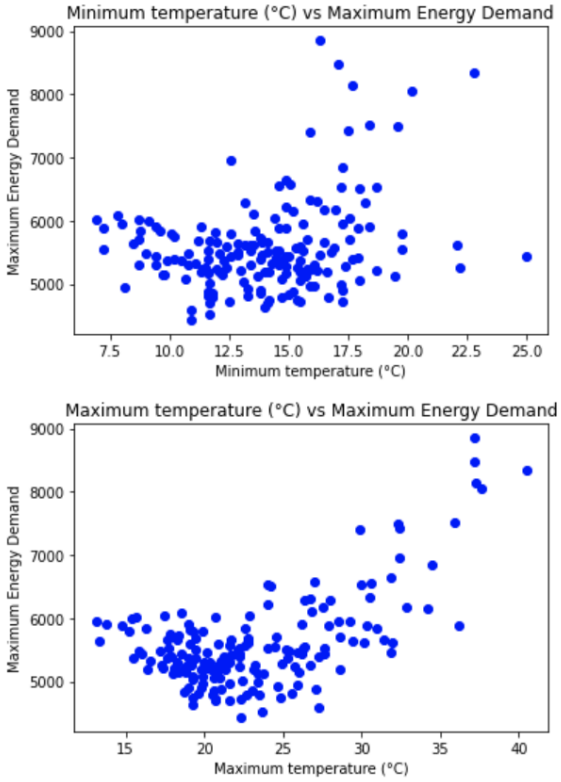
The finding of this study suggests that, contrary to the conventional belief that regression algorithms would be more suitable for continuous datasets, the discrete KNN algorithm out performed Linear Regression and Decision Tree Regression. The interpretation of these results suggests that the dataset under consideration is characterised by a certain degree of noise. The absence of certain columns in the dataset exerts influence on the ultimate outcomes, thereby culminating in the observed findings.

Further exploration of the KNN algorithm, including k-fold cross-validation and chi square feature selection, identified maximum temperature, rainfall, and 3 pm relative humidity as the three features with the most significant impact on energy use prediction.

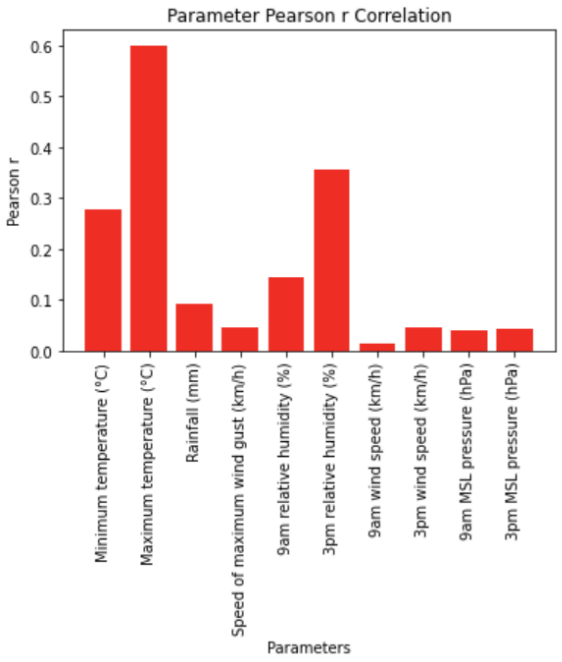
In conclusion, this study highlights the effectiveness of the KNN algorithm in predicting maximum daily energy use and emphasizes the importance of considering discrete algorithms for such analyses. Further research should focus on expanding the dataset and incorporating additional predictors to further enhance the accuracy and robustness of energy use predictions.

10. Visualizations and Tables

a. Temperature vs Maximum Energy Demand Visualisation



b. Pearson *r* Correlation between Various Independent Variables and the Target Variable



c. Price_and_demand.csv Data Types and Missing Data

```
# Check datatype of the dataframe
price_and_demand.dtypes
```

```
REGION             object
SETTLEMENTDATE     object
TOTALDEMAND       float64
RRP               float64
PERIODTYPE         object
dtype: object
```

```
# check if there is any mising data
price_and_demand.isna().sum()
```

```
REGION            0
SETTLEMENTDATE    0
TOTALDEMAND       0
RRP               0
PERIODTYPE        0
dtype: int64
```

d. Price_and_demand.csv Data Cleaning to remove irrelevant demand data

```
# Calculate maximum daily energy demand and average rrp
max_demand = price_and_demand.groupby("Date")['TOTALDEMAND'].max()
print((max_demand))
```

e. Scaler Preprocessing of Train-Test Data

```
# Preprocessing with Standardization with Mean 0, Std 1
scaler = preprocessing.StandardScaler().fit(nrg_X_train)
nrg_X_train = scaler.transform(nrg_X_train)
nrg_X_test = scaler.transform(nrg_X_test)
```

f. Weather.csv Data Types

```
weather.dtypes
```

```
Location                          object
Date                              object
Minimum temperature (°C)         float64
Maximum temperature (°C)         float64
Rainfall (mm)                    float64
Evaporation (mm)                 float64
Sunshine (hours)                 float64
Direction of maximum wind gust    object
Speed of maximum wind gust (km/h) float64
Time of maximum wind gust         object
9am Temperature (°C)             float64
9am relative humidity (%)          int64
9am cloud amount (oktas)         float64
9am wind direction                object
9am wind speed (km/h)             object
9am MSL pressure (hPa)           float64
3pm Temperature (°C)             float64
3pm relative humidity (%)        float64
3pm cloud amount (oktas)         float64
3pm wind direction                object
3pm wind speed (km/h)            float64
3pm MSL pressure (hPa)           float64
dtype: object
```

g. Conversion of Bearings

```
#Replacing all wind directions with True Bearing quantities
weather_bearings = weather_nocalm.replace(['N','NNE','NE','ENE','E','ESE','SE','SSE','S','SSW','SW','WSW','W','WNW','NW','NNW'],
                                          [0,22.5,45,67.5,90,112.5,135,157.5,180,202.5,225,247.5,270,292.5,315,337.5])
```

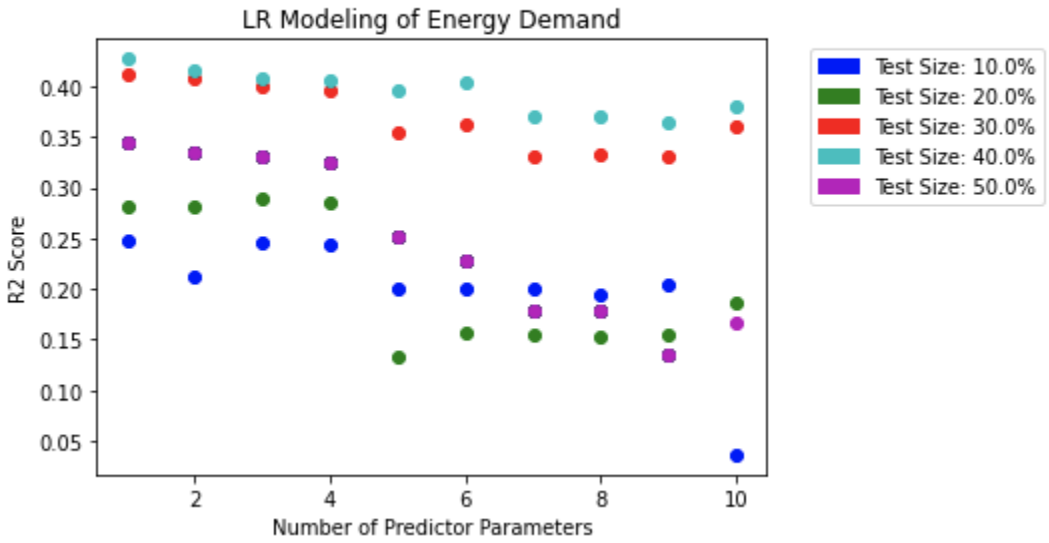
h. Utilizing Dates as Indices

```
# Convert "SETTLEMENTDATE" column to datetime
price_and_demand["SETTLEMENTDATE"] = pd.to_datetime(price_and_demand["SETTLEMENTDATE"], format="%d/%m/%Y %H:%M")
```

```
# Extract date from "SETTLEMENTDATE" and create a new 'Date' column
price_and_demand['Date'] = price_and_demand['SETTLEMENTDATE'].dt.date
```

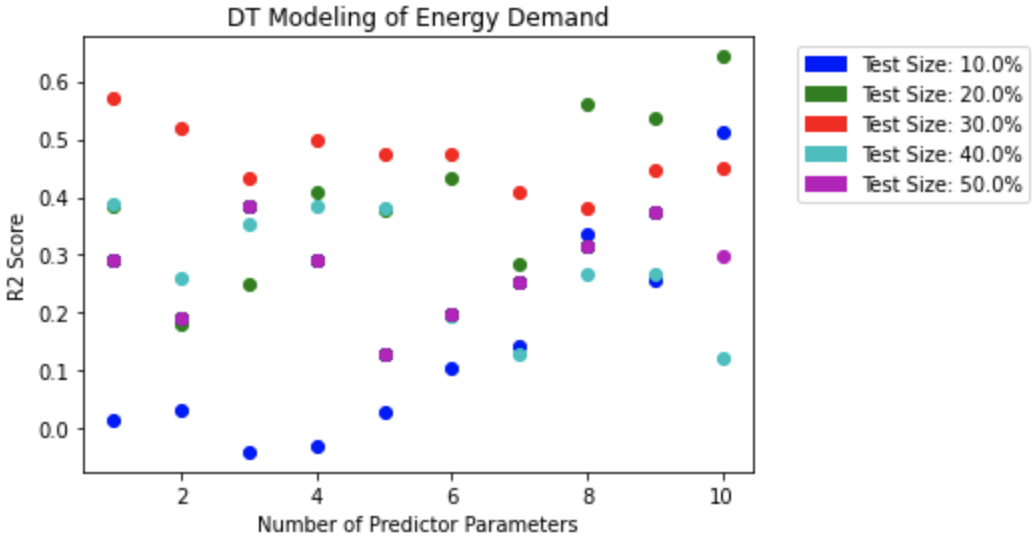
i. Linear Regression Modelling of Predictor Parameter Qty and Test Sizes

```
LR Ideal Paramaters: 1
Ideal Test Size: 0.4
R2: 0.42716155558021385
```

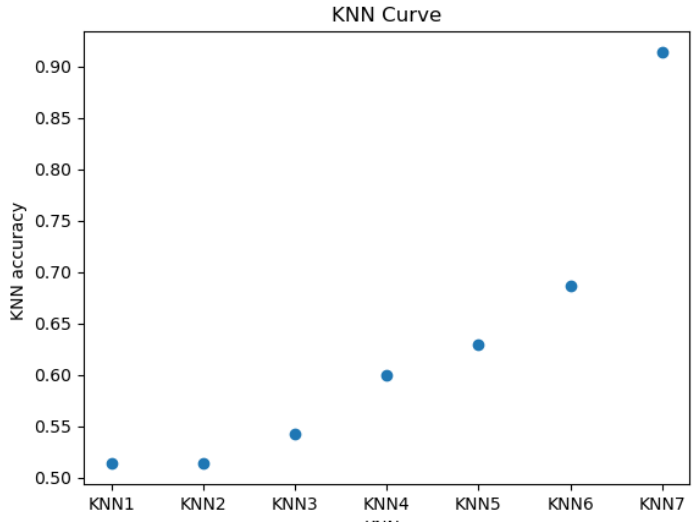


j. Decision Tree Regression Modelling of Predictor Parameter Qty and Test Sizes

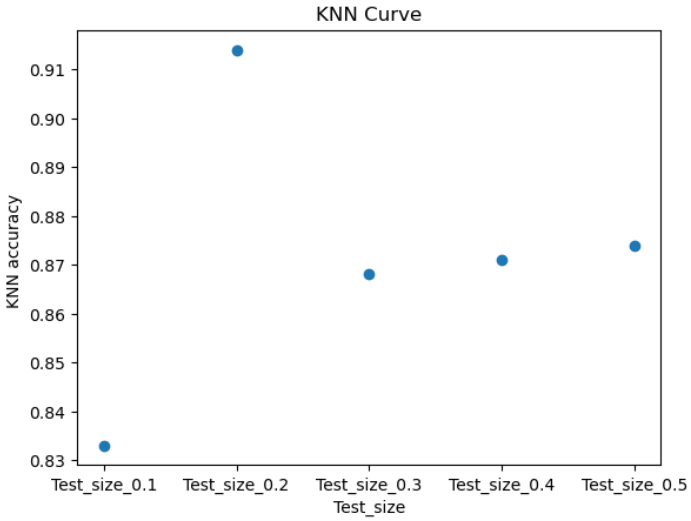
```
DT Ideal Paramaters: 10
Ideal Test Size: 0.2
R2: 0.6426249746342441
```



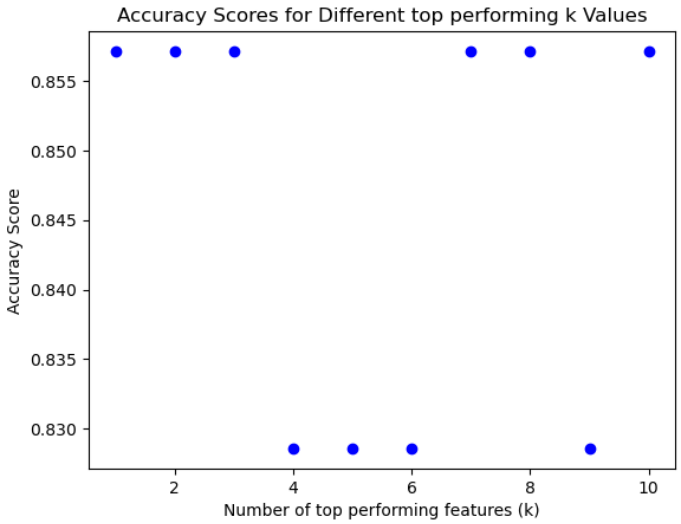
k. KNN modeling based on quantity of bins



l. KNN Test Size proportion modeling



m. KNN modeling of varying quantity of predictor parameters



* Main Content Word Count: 2482 words