Energy Usage Analysis and Model Comparison

1. Introduction

This report details the development of a predictive model for maximum daily energy usage, covering data cleaning methods, model construction, effectiveness evaluation, insights gained, and limitations.

For the creation and evaluation of this model, we utilized datasets from 'weather.csv' and 'price_and_demand.csv’

2. Objective of the Analysis

The objective of the analysis is to develop a predictive model for maximum daily energy use based on historical data, with a focus on understanding and leveraging the relationship between energy consumption and temperature. The ultimate goal is to provide energy companies with a tool that aids in planning and managing future energy usage more effectively.

3. Pre-processing and Data Cleaning

Data Sources Overview

We used 'weather.csv' and 'price_and_demand.csv’ for this Data Analysis. In the 'weather.csv' file, we have recorded daily weather information spanning from November 19, 2022, to April 24, 2023. Additionally, the 'price_and_demand.csv’ file contains data on demand and price at 48 distinct time points each day, covering the same period from November 19, 2022, to April 24, 2023.

In preparation for creating a ML model, we conducted data pre-processing using Open Refine and Python. The steps involved in this process are outlined below.

3.1. Remove Redundant and Unnecessary Columns

Empty column: Evaporation (mm), Sunshine (hours), 3pm cloud amount (oktas) columns are removed.

Irrelevant Features: Location, REGION, PERIODTYPE, RRP are removed.

3.2. Fill Missing data

Method: Imputing missing data by replacing them with the mean of the related column using the Pandas library.

For the 'Direction of maximum wind gust' column, Null values were replaced with 'SSW,' the most frequently occurring data with 63 repetitions.

3.3. Detecting Anomalous Data

We identified and removed inconsistencies in certain instances of the data, resulting in the removal of five rows. We chose to remove inconsistent data as we did not have correct values to replace them.

Anomalous Data is included:

- In 28/12/2022: 3pm temperature is less than Minimum Temperature
- In 18/01/2023: 3pm temperature is less than Minimum Temperature
- In 28/02/2023: 3pm temperature is less than Minimum Temperature
- In 11/11/2022: 3pm temperature is less than Minimum Temperature
- In 02/01/2023: 3pm temperature is less than Minimum Temperature

We replaced 'calm' in in 9am wind speed (km/h), with Zero (0).

3.4. Split merged columns

We split the 'SETTLEMENTDATE' column in 'price_and_demand.csv' into two separate columns: 'Date' and 'Time'. We use from panda library.

3.5. Outlier Detection

For outlier detection, we employed the IQR method in Python and utilized scatter plots. While we identified some anomalous data points, as explained above, no significant outliers were recognized.

3.6. Duplicate Detection

We did not find any duplicate data points.

3.7. Merge CSV files to create a dataset

Firstly, we read two CSV files into separate Data Frames (using panda read csv), after cleaning and removing irrelated columns

Then, aggregated 'TotalDemand' to determine the maximum TOTALDEMAND for each day. We utilized the pandas groupby function with Max Operation.

Finally, we performed an inner join to merge two Data Frames based on the 'Date' column, creating a new Data Frame.

4. Model Selection and Evaluation

4.1. Overview

To determine the optimal model for predicting maximum daily energy usage, we built several models, encompassing linear regression, decision tree classification, and K-Nearest Neighbours classification.
Initially assuming a linear relationship, we tested linear regression, but the results proved unsatisfactory.
Consequently, we pivoted to a demand classification approach, exploring various classifier models. This report provides an overview of diverse results obtained through the evaluation of these models. Our ultimate aim is to introduce a model with higher performance and accuracy for predicting maximum daily energy usage.

4.2. Feature Selection

We applied different Feature Selection for every model in order to find best combination of feature for our model.

We test our linear regression model with tow Feature Selection Method separately. Initially, we used correlation analysis using Pearson correlation coefficient and a Greedy approach to identify the best combination of features. And finally, we test our model with PCA method.

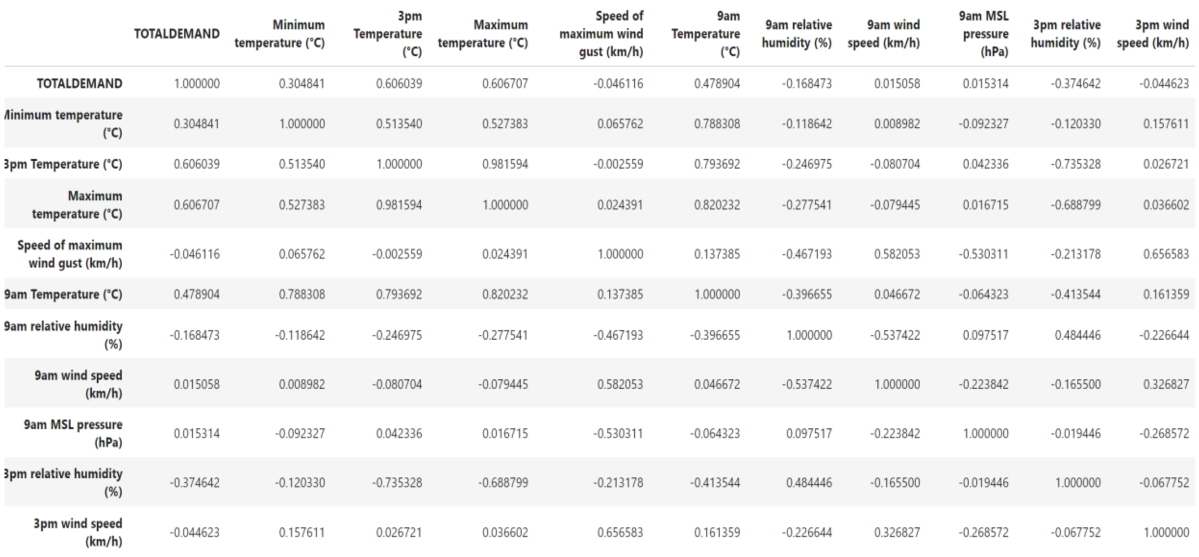
For the Decision Tree and KNN models, we employed the Chi-squared (Chi2) algorithm for feature selection. This approach yielded higher accuracy, contributing to improved model performance.

4.3. Efficient Model

We employed various models to identify the optimal one with high performance. In this section, we compare the results of three models.

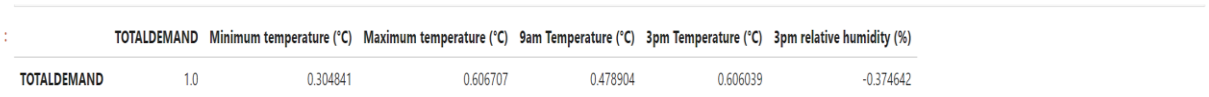
4.3.1 Linear Regression:

With Pearson correction we tried to find correlated futures.



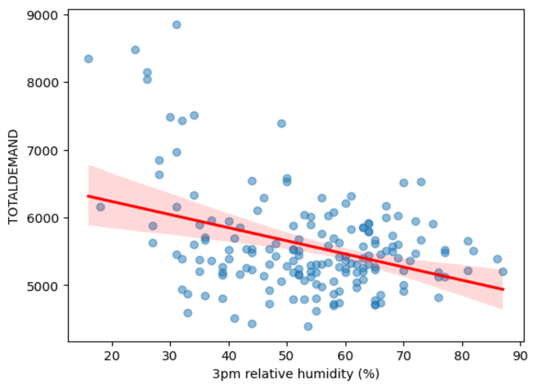
	TOTALDEMAND	Minimum temperature (°C)	3pm Temperature (°C)	Maximum temperature (°C)	Speed of maximum wind gust (km/h)	9am Temperature (°C)	9am relative humidity (%)	9am wind speed (km/h)	9am MSL pressure (hPa)	3pm relative humidity (%)	3pm wind speed (km/h)
TOTALDEMAND	1.000000	0.304841	0.606039	0.606707	-0.046116	0.478904	-0.168473	0.015058	0.015314	-0.374642	-0.044623
Minimum temperature (°C)	0.304841	1.000000	0.513540	0.527383	0.065762	0.788308	-0.118642	0.008982	-0.092327	-0.120330	0.157611
3pm Temperature (°C)	0.606039	0.513540	1.000000	0.981594	-0.002559	0.793692	-0.246975	-0.080704	0.042336	-0.735328	0.026721
Maximum temperature (°C)	0.606707	0.527383	0.981594	1.000000	0.024391	0.820232	-0.277541	-0.079445	0.016715	-0.688799	0.036602
Speed of maximum wind gust (km/h)	-0.046116	0.065762	-0.002559	0.024391	1.000000	0.137385	-0.467193	0.582053	-0.530311	-0.213178	0.656583
9am Temperature (°C)	0.478904	0.788308	0.793692	0.820232	0.137385	1.000000	-0.396655	0.046672	-0.064323	-0.413544	0.161359
9am relative humidity (%)	-0.168473	-0.118642	-0.246975	-0.277541	-0.467193	-0.396655	1.000000	-0.537422	0.097517	0.484446	-0.226644
9am wind speed (km/h)	0.015058	0.008982	-0.080704	-0.079445	0.582053	0.046672	-0.537422	1.000000	-0.223842	-0.165500	0.326827
9am MSL pressure (hPa)	0.015314	-0.092327	0.042336	0.016715	-0.530311	-0.064323	0.097517	-0.223842	1.000000	-0.019446	-0.268572
3pm relative humidity (%)	-0.374642	-0.120330	-0.735328	-0.688799	-0.213178	-0.413544	0.484446	-0.165500	-0.019446	1.000000	-0.067752
3pm wind speed (km/h)	-0.044623	0.157611	0.026721	0.036602	0.656583	0.161359	-0.226644	0.326827	-0.268572	-0.067752	1.000000

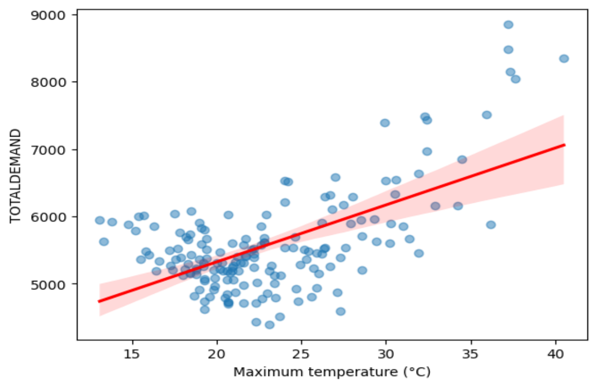
The above table shows multicollinearity between Maximum temperature and 3pm temperature. Two independent variables with a 98% correlation might carry similar information.
This multicollinearity can present challenges in regression analysis. Implementing PCA to transform the correlated variables into a set of linearly uncorrelated variables can help retain the essential information while mitigating multicollinearity.

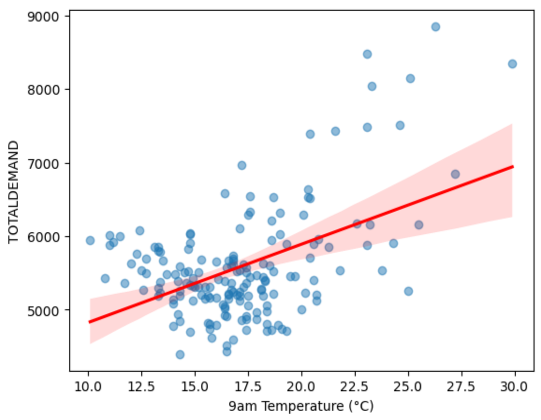


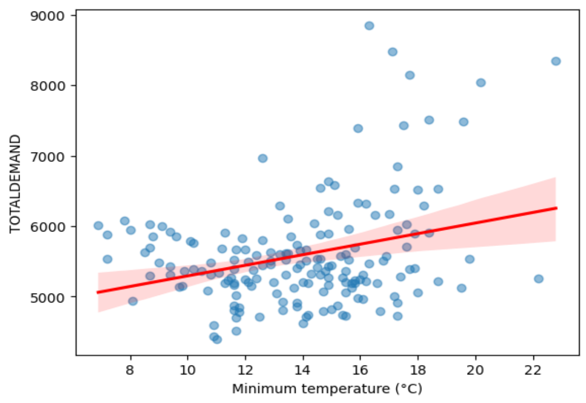
	TOTALDEMAND	Minimum temperature (°C)	Maximum temperature (°C)	9am Temperature (°C)	3pm Temperature (°C)	3pm relative humidity (%)
TOTALDEMAND	1.0	0.304841	0.606707	0.478904	0.606039	-0.374642

in visualisation we created scatter plots with regression lines to visualize the relationship between the ‘TOTALDEMAND’ and various weather-related features given in the weather csv file. This step is crucial in data analysis for exploring potential correlations and helpful in understanding how the other features influence the total demand.









For the Linear Regression model, using both Pearson correlation and the Greedy method, we identified 'Maximum temperature (°C)', '9am Temperature (°C)', and '3pm relative humidity (%)' as having a more favourable correlation matrix with Total Demand.

Model Training

we performed train and test splitting with the test size of 0.20 and trained our model with the 80% of the data. To predict for the 20% test set.

Hyperparameters used: Test size = 0.20

Evaluation of the Model

The evaluation metrics suggest that this combination is not sufficient for a linear regression model.

R-squared: 0.46 Approx.
mean_squared_error:439556.38 Approx.

In Conclusion With an R-squared value of 0.46, our linear regression model accounts for approximately 46% of the variability in predicting Total Demand. While this suggests a moderate degree of explanatory power, it also indicates that a significant portion of the variability remains unexplained by the model.
The Mean Squared Error of 423,557.06 reflects the average squared difference between predicted and actual Total Demand values. This result suggests that, based on our data, a linear regression model may not be the most efficient choice for predicting Total Dem and.
We will work on other models to compare the efficiency and accuracy of the model to decide which model works best for our dataset.

4.3.2 Decision Tree Model using K_Mean Classification

To improve the model's performance, we explored classifier models. Since the maximum Total Demand falls within the range of 4396 and 8851, it is suitable to consider classifier models. This approach enables the model to categorize and classify Total Demand into specific ranges (or clusters), providing a more accurate prediction for values within this interval. Classifier models are well-suited for scenarios where the target variable is discrete or falls into distinct categories, making them a valuable alternative for refining predictions in this context.

Feature filtering

We employed the Chi-squared (Chi2) algorithm for feature filtering, specifically with k=4. This method allowed us to select the most relevant features from our dataset based on their statistical significance.

Elbow Method as a clustering algorithm

For this the above model, we implemented the Elbow Method to determine the optimum number of Clusters(K) for a KMeans clustering algorithm based on the sum of squared distances(inertia). The Elbow Method helps in finding the optimal numbers of clusters by looking for an "elbow" point in the plot where the rate of decrease in inertia sharply changes. We also supressed any warning that arisen during the KMeans fitting.

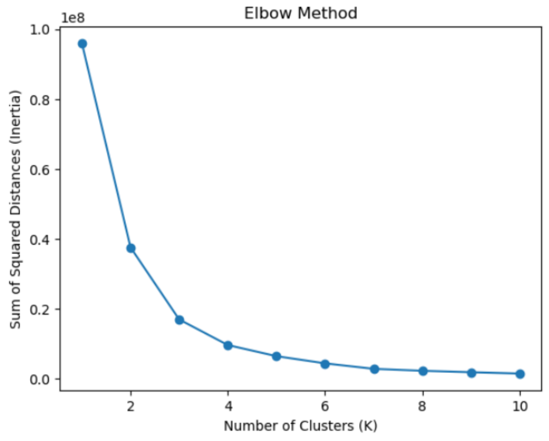


Figure 6: *The Visual shows the rate of decrease in inertia sharply changes between K values 2 and 4.*

With the K-means algorithm, we partitioned Total Demand into three distinct groups. To represent these categories, we introduced a new field named "MaxDemand," which serves as a class identifier for each data point. This approach enables the model to classify Total Demand into specific groups based on similarities, providing a structured representation of the data.

k-fold cross-validation as Splitting Strategies

We implemented the k-fold cross-validation technique (with n_splits=10) as our chosen splitting strategy in model development. This involved dividing the datas et into 10 subsets and iteratively using each subset for both training and validati on in a rotating fashion. Our performance metric was the average accuracy calcu lated across all folds. This approach ensured comprehensive training and validati on across the entire dataset.

Model Evaluation

The Decision Tree Classifier and accuracy_score from scikit-learn was used to ev aluate the performance of the model.

- **Average Accuracy :** 70.22875816993464
- **max depth** : 4 (based on Gride algorithm)
- **Clustering algorithm** : K_mean(k=3)
- **criterion :** Gini
- **Feature filtering :** Chi-square SelectKBest(k=4))
- **Selected features :** 'Maximum temperature (°C)', 'Minimum temperatur e (°C)','9am Temperature (°C)', '3pm relative humidity (%)'
- **Splitting Strategies :** KFold(n_splits=10, random_state=42)

In Conclusion, The Decision Tree model achieved an average accuracy of approx imately 70.23% with adjusted hyperparameters. While this performance signifie s a moderate level of predictive success, there is room for potential model refin ement, and exploring alternative algorithms may be considered to enhance over all predictive capabilities.

4.3.3 KNeighbors Classifier Model using K_Mean Classification

Given the results obtained from the Decision Tree model, our next step involves transitioning to a different model—specifically, the K-Nearest Neighbours (KNN) algorithm.

The rationale behind this shift is to explore an alternative approach that might better capture the underlying patterns within the data.

KNN is particularly valuable in scenarios where the relationships between data points are non-linear or complex.

Our strategy entails training the KNN model, optimizing its hyperparameters, and rigorously evaluating its performance in comparison to the Decision Tree.

The KNN Classifier was initialized with "n_neighbors = 5", indicating that the model considers the five nearest neighbours for classification.

Feature filtering and a clustering algorithm

We employed the Chi-squared (Chi2) algorithm for feature filtering, specifically with k=4 and Elbow Method as a clustering algorithm.

K-fold cross-validation as Splitting Strategies

We implemented the k-fold cross-validation technique (with n_splits=10) as our chosen splitting strategy in model development.

Model Evaluation

The following displays the hyperparameters and results of the K-Nearest Neighbors (KNN) model:

- **avg_acc_scorec :** 0.7480392156862745
- **acc_score :** [0.8333333333333334, 0.7647058823529411, 0.8823529411764706, 0.5882352941176471, 0.6470588235294118, 0.7058823529411765, 0.7647058823529411, 0.8235294117647058, 0.7058823529411765, 0.7647058823529411]
- **Feature filtering :** Chi-square SelectKBest(k=4))
- **Selected features :**Maximum temperature , Minimum temperature, 9am Temperature , 3pm relative humidity
- **Clustering algorithm :** K_mean(k=3)
- **Splitting Strategies :** KFold(n_splits=10, random_state=42, shuffle=True)

In Conclusion, with an average accuracy score of 74.8%, the K-Nearest Neighbours (KNN) model demonstrates better predictive capabilities on the given task. This high accuracy suggests that the model effectively captures patterns and relationships within the dataset, making it a choice for the predictive task at hand.

5. Overview of results

5.1. Final model

The analysis is centred around the K-Nearest Neighbours (KNN) model, a machine learning algorithm used for classification tasks. KNN works by identifying the 'k' nearest data points to a given input, and the majority class among these neighbours is assigned to the input.

The KNN model is trained using specific features that are considered relevant for predicting the Max of daily energy usage. The selected features are Maximum Temperature, Minimum Temperature, 9am Temperature, and 3pm Relative Humidity based on Chi2 algorithm.

The model's performance is assessed using an accuracy metric. Accuracy measures the percentage of correctly predicted instances out of the total instances. An average accuracy score of 74.8% is achieved, indicating that the model accurately predicts the target variable for about 74.8% of the instances in the dataset.

5.2. Insights

The K-Nearest Neighbours (KNN) model, utilizing features such as Maximum and Minimum Temperature, 9 am Temperature, and 3 pm Relative Humidity, demonstrates a acceptable average accuracy of 74.8%.

This suggests that daily energy usage is notably influenced by varying weather conditions throughout the day.

The inclusion of temperature at specific times (9 am and 3 pm) highlights the significance of time-of-day fluctuations, indicating potential peaks or shifts in energy demand corresponding to these periods.

Moreover, the model's sensitivity to Relative Humidity underscores the impact of humidity levels on energy consumption patterns.

Overall, the KNN model provides valuable insights into the interplay between time, weather conditions (temperature and humidity), and daily energy usage, offering a foundation for informed energy management decisions.

5.3. Results limitations

We have 175 records spanning four months of the year. This subject may impact the generalizability and reliability of the results. The limitations that may impact our results include:

Four-Month Data Coverage

The dataset exclusively spans four months, particularly during hot weather, potentially limiting its ability to comprehensively capture seasonal variations. As energy consumption patterns distinctly differ across various seasons, the model's capacity to generalize to months or seasons beyond the specified timeframe may be constrained.

Small Dataset

the dataset may be considered small (175 record). The risk of overfitting increases with a smaller dataset, potentially leading the model to capture noise in the data as if it were a meaningful pattern. This may impact the model's performance on new, unseen data.

In a small dataset, the influence of occasional dates, such as holidays, may not be adequately discernible or representative.

GitHub Link: https://github.com/edalatnejad/Assessment2_MelUni