INC 362-Final Technical Report

Industrial Heat Demand Project

Professor: Dr. Teema Leangarun

Project Team (Group 2)

Data Analyst: Petdarat Vagost 66070504011

Data Engineer: Kongpob Suranan 66070504002

Data Engineer: Phurinut Soayyala 6670504012

Data Scientist: Sirawit Hengroong 66070504017

Data Scientist: Napatsorn Amadmuntree 66070504007

Introduction

The following project addresses an environmental concern about the high carbon emissions of the facility, which are recorded in the operational records. Although underreported previously, the data on emissions show levels that require immediate attention. These figures highlight an urgent requirement for emission reduction strategies that would reduce the contribution of the facility to global climate change.

The scope of the given project involves three main tasks. First, it incorporates a critical analysis of the presently existing energy-related data for developing a background of the present emission levels. Second, the project identifies and categorizes the principal sources of carbon emissions within facility operations. Third, it analyzes the share of different types of fuels and operational units in the total amount of emissions. This study also discusses possible directions for future monitoring and reduction of emissions.

The report is structured in a way that takes a comprehensive look at the current environmental status of the facility. It describes the methodology implemented for cleaning and analysing data, which enables transparency and reproducibility of the results. Further, it reports preliminary findings on the emission patterns and trends observed in the given dataset. The results of this analysis are expected to provide a basis for making informed decisions to help strategic planning in further endeavors and assist the facility in moving toward more energy-efficient and environmentally friendly operations.

Data Management & Pipeline

○ **Detail the steps and supporting reasons**

- **Load the file safely.** Apply multiple encoding detection strategies to ensure robust file parsing across varied character sets and prevent data loss from encoding mismatches.
- **Fix facility IDs.** I clean and standardize the facility identifier and create a placeholder ID if it’s missing.
- **Turn messy numbers into real numbers.** Remove commas, currency symbols, and parentheses so numeric columns become numeric.
- **Find and combine fuel columns.** Look for any columns that mean coal, gas, diesel, etc., and add them up into standard fuel columns.

- **Group rows into units.** Combine rows by facility and unit name so each row represents a physical unit.
- **Standardise unit types.** Map many different text labels (like “CP-1” or “steam boiler”) to a small set of canonical types, such as boiler, heater, CHP.
- **Score each unit for electrification.** For each unit, assign a simple 0–1 score indicating how easy it is to electrify, using unit type and temperature if available.
- **Compute electrifiable energy.** Multiply each unit’s score by its heat demand, then sum those values per facility to get the facility’s potential electrifiable energy.
- **Build the facility table.** I aggregate totals, fuel shares, emissions, and concentration metrics and merge the electrification results into a single facility summary file.
- **Clean up helper columns.** I remove intermediate columns because these columns are needed only for the machine to understand, not the readers.

○ **Decisions with supporting reasons**

Load the file safely

Reason: Multiple encodings prevent parse failures and ensure all data is captured despite source inconsistencies.

Normalise column names

Reason: Standardising to lowercase snake_case eliminates lookup errors and improves code reliability.

Fix facility identifiers

Reason: Clean, consistent IDs prevent splitting facilities across rows and enable proper aggregation.

Turn messy numbers into real numbers

Reason: Removing formatting characters (commas, currency, parentheses) ensures accurate calculations.

Find and combine fuel columns

Reason: Detecting variants and summing into canonical fields (coal, gas, diesel, other) produces consistent fuel metrics.

Group rows into units

Reason: Aggregating by facility and unit name eliminates duplicates and ensures one-row-per-unit accuracy.

Standardise unit types

Reason: Mapping noisy variants to canonical types (boiler, heater, chp) enables reliable rules and prevents misclassification.

Score each unit for electrification

Reason: The 0–1 elec_score quantifies technical feasibility per unit, enabling granular, traceable decisions.

Compute the electrifiable energy per unit and facility

Reason: Multiplying score by demand yields elec_potential_kwh; summing gives facility-level potential_energy_kwh and potential_fraction for energy-weighted decisions.

Build the facility summary table

Reason: Aggregating totals, fuel shares, emissions, and electrification metrics into one table simplifies prioritisation and reporting.

Clean up helper columns before saving

Reason: Removing intermediate columns keeps outputs focused on actionable fields while preserving key diagnostic data.

○ **Describe any new features you engineered and why they are useful.**

- elec_score: per-unit technical feasibility (0–1) for electrification; enables granular engineering judgment.
- elec_potential_kwh and potential_energy_kwh: quantify electrifiable energy in kWh for prioritisation.
- potential_fraction: energy-weighted share used to map to Yes/Partial/No; avoids misleading facility labels.
- facility_total_energy_kwh: canonical facility energy for consistent denominators.
- fossil_share: total fuel that is based on fossil fuels.
- electrification_reason: compact diagnostic for traceability.
- concentration_hhi: indicate the concentration of unit type; higher = more concentration.
- percent_top1_unit_type: share of energy from the most dominant unit type

○ **Exploratory Data Analysis (EDA) & Feature Engineering**

In-depth analysis of data distribution, missing values, correlations, and outliers

-Data distribution

-Annual Heat Demand (kWh): Highly skewed; majority of units consume <10,000 kWh, while a small number exceed 1M kWh, creating a long right tail.

-Emissions ($MMTCO_2e$): Strongly correlated with demand, most units are near zero, but a few facilities dominate totals.

-Electrification Score : Discrete distribution clustered at rule-based values (0.2, 0.4, 0.6, 0.8, 1.0).

-Fuel Mix: Coal and natural gas dominate while diesel and LPG/other fuels appear sporadically.

-Missing values

-Unit Names: Occasionally absent; filled with row indexes to maintain uniqueness.

-Fuel Columns: Sparse; many facilities report only one or two fuels, others filled with 0.

-Correlations

-Heat Demand with Electrification Potential: Near-linear relationship, scaled by electrical score.

-Fossil Share with Electrification Feasibility: Negative correlation; facilities with high fossil share tend to have lower feasibility labels.

-Outliners

-Mega-facilities: A handful of facilities account for >50% of total demand and emissions.

-Fuel Mix Extremes: Some facilities report 100% coal or 100% “other,” skewing fossil share statistics.

-Separated 2 data files

This was decided in discussions between the data engineers and data scientists. We decided to separate the process so that each data scientist will find it easier to work with. Thus, the data engineers produce outputs fully meeting the requirements of the data scientists.

-Final data set

As described above, the first dataset simply summarizes each facility. For each ID, we use aggregation so that multiple records with the same facility ID and end_use are combined into a single record; we then remove duplicates so that each facility appears in the file only once. In this dataset, we also generate the electrification score and the electrification potential. These numbers indicate how much of the facility's energy use could theoretically be transitioned away from fossil fuels onto electricity, considering types of processes in use such as furnaces, heaters, and boilers. For the second dataset, it is possible for the same facility ID to occur more than once. In this file we do not aggregate the processes into one row. That lets us retain more detail about each process within the same facility.

○ **Data Visualization Explanation**

Reason: To visualize the data into the graph that is easy to read and understand the information and see the differences on what the cleaned data provided.

We have four graphs to explain what each data is about.

-The 1st graph is showing Top 5 Facilities by Total Annual CO2 Emission.

We use the bar graph to show the data because the bar graph is the easiest to compare when it comes to showing the top 5 of facilities ranging from highest to lowest.

-The 2nd graph is showing CO2 by Fuel Type and Year (stacked by each year)

For this data we decided to use the stack bar graph. We decided to use this kind of graph because we wanted to see the change of each year starting from 2010 to 2015 and decided to show only the top 3 of facilities because the fourth and fifth place bar is too small to show.

-The 3rd graph shows the Top 5 unit types by Total Annual CO2.

This one we use a horizontal bar graph to show the data. This graph shows what type of machinery consumes fuel the most per year.

-The 4th graph is show Top 5 Facilities by Total annual cost

We use the bar graph to show the data of the top 5 total cost of the facilities ranging from most to lowest

Creation and selection of features used for modelling

○ **Feature Selection**

Decision tree

During feature analysis, several columns were identified as redundant or irrelevant to the predictive objective. Multiple CO_2-related variables existed, many describing similar information. To preserve model interpretability while avoiding multicollinearity, only **annual_co2_tonnes** was retained because:

- It directly represents facility emissions.
- It serves as the most interpretable CO_2-related feature.
- It captures the overall carbon footprint critical to electrification decisions.

The following features were intentionally removed:

- facility_id—Only serves as an identifier
- unit_name—Highly specific to individual equipment
- city—not generalizable to new regions or contexts
- year—Temporal information not essential to the decision framework

This decision aligns with the goal of creating a general-purpose model.

columns_to_drop = ['facility_id','unit_name', 'city', 'mmtco2e_raw','mmtco2e_by_unit_type_mmt','elec_score','overall_mmtco2e_by_unit_type_mmt', 'mmtco2e_by_unit_type_tonnes','overall_mmtco2e_by_unit_type_tonnes','year',

'fuel_cost_usd', 'electricity_cost_usd' , 'mean_fuel_price_per_unit', 'est_other_cost']

df_clean = df.drop(columns=columns_to_drop, errors='ignore')

○ **Encoding Categorical Data**

Decision tree

The selected model (Decision Tree) does not require feature scaling because it uses **threshold-based decision boundaries**, not distance-based metrics. However, categorical variables need to be transformed into numeric features.

X_encoded = pd.get_dummies(X, columns=categorical_cols, drop_first=True)

This ensures all features are machine-readable while keeping the dataset interpretable.

Modeling Methodology (CLO2)

○ **Experimental Setup**

Decision tree

This model aims to develop an interpretable and production-ready machine learning pipeline that predicts whether a facility should transition from fuel consumption to electrification. The modeling methodology involves the full lifecycle from data ingestion and cleaning, through feature engineering, to model selection, evaluation, and comparison.

The analysis was conducted using Python 3.12 with essential scientific libraries including **Pandas**, **Scikit-Learn** and **Matplotlib**. All experiments were run in Jupyter Notebook (see model.ipynb). This section details the experimental environment, data preprocessing procedures, and modeling approach.

○ **Required Libraries and Initial Setup**

Decision tree

The following Python libraries were imported to support data loading, preprocessing, visualization, and model training:

```
import pandas as pd

import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.tree import DecisionTreeClassifier, plot_tree

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

import matplotlib.pyplot as plt
```

A Pandas DataFrame was created to load the dataset:

```
df = pd.read_csv("facility_emissions_cleaned.csv")

df.head()

df.info()
```

The .info() output confirmed the dataset contained a mixture of numerical and categorical features with no missing values in the target column.

To verify the target distribution:

```
df['target'].value_counts()
```

This step revealed that the dataset is **highly imbalanced**, with class 1 dominating significantly and class 0 having a much smaller representation. This imbalance impacted the modeling strategy, particularly the choice of test size and the need for careful evaluation using precision and recall metrics.

○ **Train-Test Split Consideration**

Decision tree

Given that class 0 contains very few samples, using a smaller test size (e.g., 10%) would risk having **no samples of class 0 in the test set**, making evaluation unreliable. Therefore, a **20% test size** was selected to guarantee both classes appear in both splits.

```
target_column = 'target'

X = df_clean.drop(columns=[target_column])

y = df_clean[target_column]

X_train, X_test, y_train, y_test = train_test_split(

    X_encoded, y, test_size=0.2, random_state=42)
```

○ **Model Selection**

Decision tree (Test 3 classification model)

1. **Decision Tree Classifier**
2. **Random Forest Classifier**
3. **SMOTE + Random Forest Classifier**

The **Decision Tree** was chosen as the primary model because of the project's emphasis on **interpretability**. While ensemble models may yield marginally higher accuracy, they are less transparent, making them unsuitable for explaining electrification decisions to stakeholders.

The Decision Tree provides:

- Clear rule-based logic
- Full visibility into split thresholds
- Direct interpretability for policy decision justification

Additionally, the model is robust to categorical features and does not require scaling.

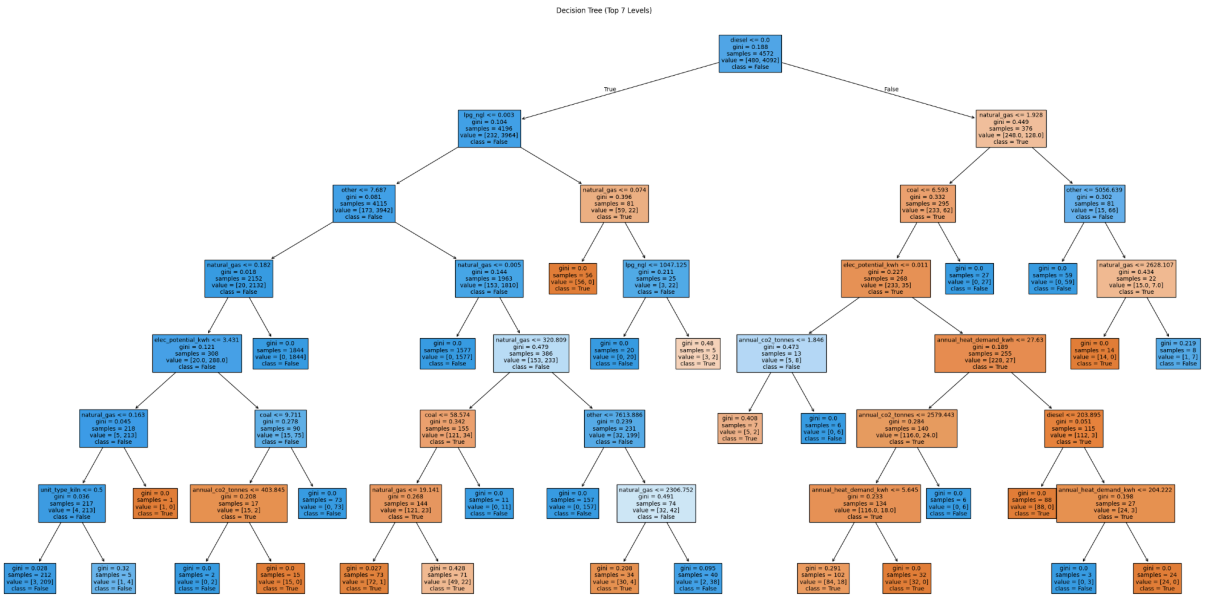
A sample of the training code:

```
tree = DecisionTreeClassifier(

    max_depth=7,  # limit depth to prevent overfitting

    min_samples_split=10,  # need at least 10 samples to split

    random_state=42)
```

```
tree.fit(X_train, y_train)
```

A visualization was generated using below code snippet:

```
# Visualize the tree (simplified version)
plt.figure(figsize=(30, 15))
plot_tree(tree,
          feature_names=X_encoded.columns,
          class_names=['True', 'False'],
          filled=True,
          max_depth=7,  # only show top 3 levels
          fontsize=10)
plt.title('Decision Tree (Top 7 Levels)')
plt.tight_layout()
plt.show()
```



This visualization enables domain experts to understand why the model decides that a facility should or should not electrify.

Results and Evaluation (CLO2)

This section summarizes the classification model performance across accuracy, precision, recall, and confusion matrices. These metrics were computed using the Scikit-Learn evaluation suite.

Model	Accuracy	Precision	Recall	Confusion Matrix
Decision tree	0.9738	0.99	0.98	[[997 24] [6 117]]
Random Forest	0.9537	0.99	0.96	[[984 37] [13 110]]
SMOTE + Random Forest	0.9231	1.00	0.92	[[936 85] [3 120]]

○ **Decision Tree Performance Analysis**

The Decision Tree achieved the **best balanced performance**, with strong metrics across accuracy, precision, and recall. The optimal parameters:

- max_depth = 7
- min_samples_split = 10

The confusion matrix indicates only a small number of misclassifications, and the high recall score shows the model rarely misses facilities that should electrify.

The tree structure further revealed decision rules involving CO_2 emissions, operational parameters, and cost-related features. These rules provide significant insight into actionable conditions that justify electrification.

○ **Random Forest Performance Analysis**

Even with optimized hyperparameters:

- n_estimators = 300
- max_depth = 7

- min_samples_split = 5
- class_weight = {0:10, 1:1}

The Random Forest performed worse than the Decision Tree. Although precision was high, the model struggled with recall and failed to capture several class-0 samples.

This poor performance demonstrates that ensembling did not provide additional value for this dataset.

○ **SMOTE + Random Forest Analysis**

SMOTE (Synthetic Minority Oversampling Technique) was used to correct the class imbalance, followed by training a second Random Forest model.

Optimized parameters:

- n_estimators = 200
- max_depth = 8
- min_samples_split = 10
- class_weight = {0:8, 1:1}

While this model achieved a **perfect precision of 1.00**, it suffered a major decline in accuracy and recall due to overfitting on synthetic samples.

This outcome suggests that SMOTE introduces patterns that do not generalize well.

System Architecture (CLO3)

Decision tree

The final system architecture supports integration with **Power BI** for visualization and decision reporting.

In the final dataset, two new columns were added:

1. **predict**—The model's predicted value for each facility
2. **BI**—A tri-state evaluation column defined as:
    - 1 if both target and predict = 1
    - 0 if both target and predict = 0
    - 2 if target ≠ predict

Before adding these columns, the model was used to predict values for **the entire dataset**, not just the test split.

```
# Predict on entire dataset

y_pred_full = tree.predict(X_encoded)

# Add predicted column

df['predicted'] = y_pred_full

# Create BI column

df['BI'] = 2  # default to mismatch

# Set to 1 where both are 1

df.loc[(df['target'] == 1) & (df['predicted'] == 1), 'BI'] = 1

# Set to 0 where both are 0

df.loc[(df['target'] == 0) & (df['predicted'] == 0), 'BI'] = 0

# Save to new CSV

df.to_csv('dataset_with_predictions.csv', index=False)
```

These columns allow Power BI to compute:

- Total number of recommended electrification candidates
- Accuracy of recommendations
- Counts of mismatches for risk analysis

target	predicted	BI
1	1	1
0	1	2
0	0	0
1	1	1

K-Means Clustering

Modeling Methodology

○ **Experimental Setup**

To analyze energy consumption, the data set consists of multiple energy related variables for industrial facilities. The features used for modeling include **total_annual_heat_demand and fuel ratio for coal, diesel, natural gas, lpg_ngl, and other.**

- **Required Libraries**

    1. Pandas - for data loading, cleaning, and handling table dataset
    2. Numpy - provides numerical operations and array
    3. Matplot - for creating base plot and visualize clustering results
    4. Seaborn - for enhanced and more aesthetic visualizations, including heatmaps.
    5. StandardScaler - Normalization features to equal scale for proper K-Means performance
    6. KMeans - The core clustering model for segmenting factories by behavior
    7. Silhouette_score - Evaluates cluster quality and help determine optimal K
    8. GaussianMixture - Alternative clustering method used for comparison
    9. PCA - Reduces dimensionality to visualize clusters in 2D.

- **Preprocessing Steps**

    - Handled NaN and infinite values from ratio units
    - Computed fuel ratios using; ratio = fuel usage by type/ total fuel
    - Select heat demand as the primary indicator of energy consumption
    - StandardScaler applied to normalize difference in feature scales
    - Used PCA (2 components) to visualize cluster boundaries

---

○ **Model Selection**

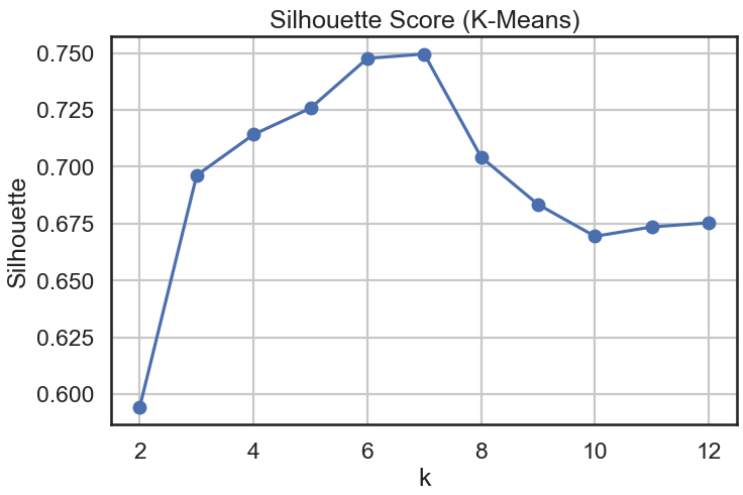
- Model Comparison

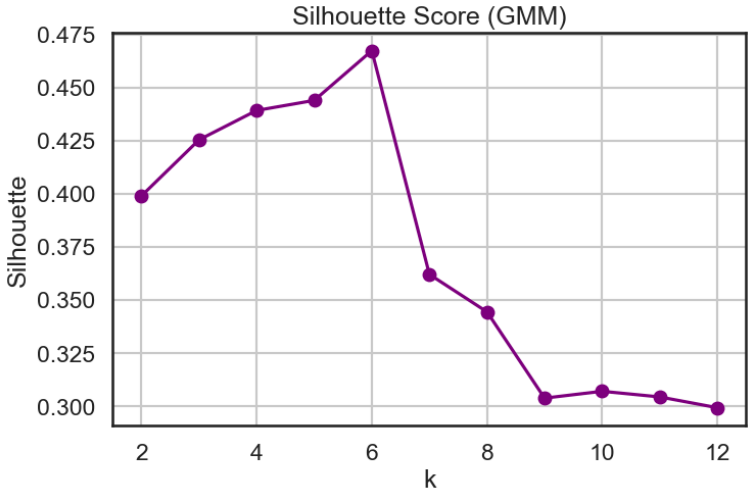
    We compare **K-Means and Gaussian Mixture Model** because they are the two most common unsupervised clustering methods.
    K-Means works well when clusters are or non label separated.
    GMM can handles overlapping or non spherical cluster

    How the best model was chosen

    - Test K = 2 to 12 on both model

- Evaluate Silhouette Score for each K
- Identified the best K for each model





K-Means Best K is K = 7
Silhouette Score = 0.74957

GMM Best K is K = 6
Silhouette Score = 0.46733

Comparison Result: K-Means's Silhouette score was higher than GMM, so we chose the K-Means model. After selecting K-Means as the preferred model, K comparison was performed to find the clearest segmentation.

How to choose number of K

- We select the number of clusters from the Silhouette Score calculation.

```
k=1, silhouette=nan
k=2, silhouette=0.5944564700364916
k=3, silhouette=0.6962462647065369
k=4, silhouette=0.7141940225449064
k=5, silhouette=0.7257728756546197
k=6, silhouette=0.7476685037316889
k=7, silhouette=0.7495723770215394
k=8, silhouette=0.7042098473337174
k=9, silhouette=0.6834330225923321
k=10, silhouette=0.669431861291369
k=11, silhouette=0.6735503379495269
k=12, silhouette=0.6753799391835661
```

```
Cluster size for k = 6
0    636
1    231
2    119
3     32
4      6
5      4
Name: count, dtype: int64

Cluster size for k = 7
0    230
1    636
2    119
3      1
4     32
5      4
6      6
Name: count, dtype: int64
```

The Silhouette Scores for **k = 6 and k = 7 were very close**, meaning the metric alone could not clearly distinguish which K was better. When Silhouette values are similar, evaluating the

internal structure of clusters becomes necessary. We decided to look at the number of members in each cluster.

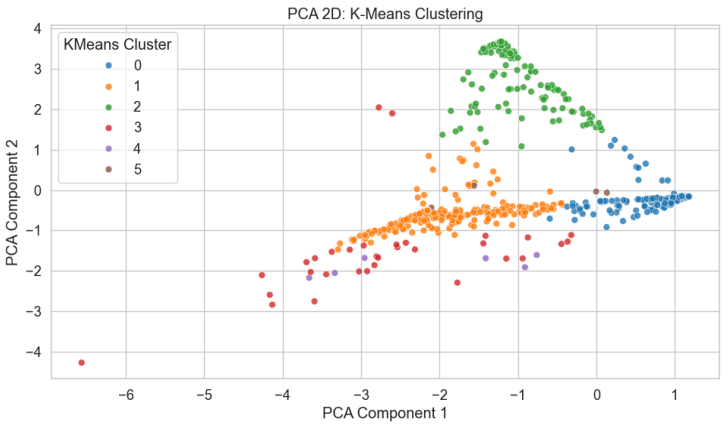
From the result: K = 6, all clusters have reasonable sizes, no cluster has 1 member.

K = 7, one cluster had only 1 member, statistically weak and not meaningful

Therefore, K = 6 and K = 7 had similar Silhouette scores, we used cluster size to break the tie. K=7 produces a singleton cluster, so k=6 is the more appropriate and stable choice.

Results and Evaluation (CLO2)

This section presents the comprehensive results of the final clustering model, focusing on the characteristics and behavior of the selected K-Means solution. The analysis highlights the resulting cluster structures, energy heat demand patterns, fuel ratio, and visual such as PCA separation and heatmap interpretations. These outputs collectively describe how the model segments factories based on real energy use behavior.



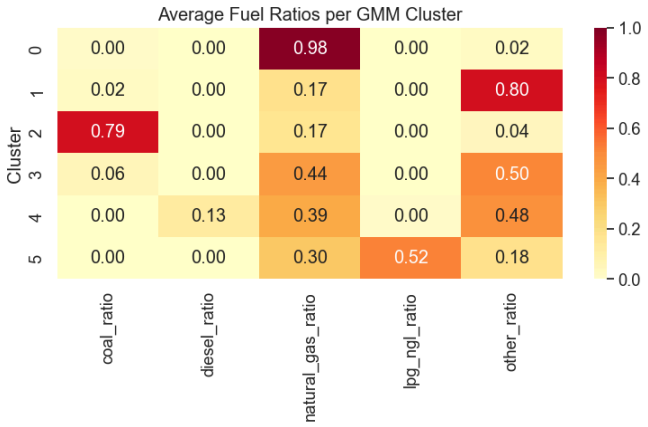
	PCA1	PCA2
total_annual_heat_demand_kwh	-0.408875	-0.295679
coal_ratio	-0.200356	0.846246
diesel_ratio	-0.113294	-0.122525
natural_gas_ratio	0.653225	-0.256194
lpg_ngl_ratio	-0.041986	-0.013863
other_ratio	-0.592771	-0.340003

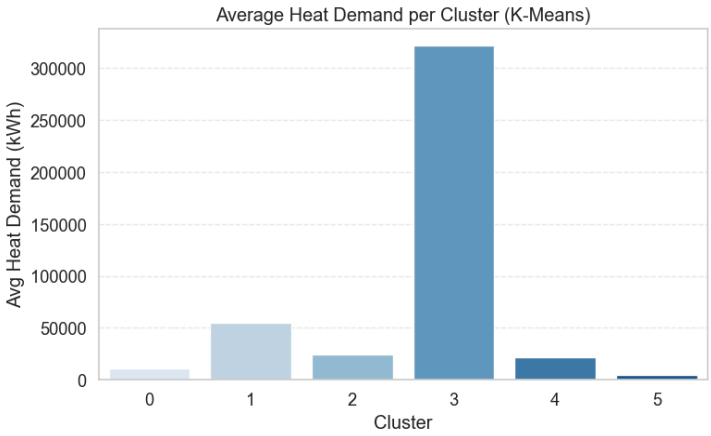
PCA helps visualize the underlying structure of the clustering model by transforming the original variables (heat demand + fuel ratios) into two major axes (PCA1 and PCA2) that explain most of the variance.

PCA1 (Horizontal Axis): Overall Energy Pattern
- PCA1 is dominated by natural_gas_ratio (positive)

- High PCA1: natural-gas-heavy factories
- Low PCA1: high heat demand or mixed-fuel factories

PCA2 (Vertical Axis): Fuel mix patter
- PCA2 is dominated by coal_ratio (positive)
- High PCA2 → coal-intensive facilities
- Low PCA2 → non-coal or mixed-fuel factories





○ Average Fuel Ratios per Cluster

- Cluster 0: Natural Gas Dominant (98%)
- Cluster 1: Other Fuel Dominant (80%)
- Cluster 2: Coal Dominant (79%)
- Cluster 3: Natural Gas (44%) + Other (50%) Mix
- Cluster 4: Diesel (13%) + Natural Gas (39%) + Other (48%) Mix
- Cluster 5: LPG/NGL Dominant (52%)

○ Average Heat Demand per Cluster

- Cluster 5 & Cluster 0 - Low Heat Demand
    - Clusters 5 and 0 show the lowest heat demand.
    - These clusters represent low-energy factories, even if they rely heavily on a single fuel type.
- Cluster 4 & Cluster 2 - Low to Medium Heat Demand
    - Clusters 4 and 2 fall within the Low–Medium heat demand range.
    - Even with dominant fuels (e.g., high coal ratio in Cluster 2), their total heat load remains moderate.
- Cluster 1 - Medium Heat Demand
    - Cluster 1 represents Medium heat demand.
    - The higher energy load aligns with a high share of “other” fuels.
- Cluster 3 - Highest Heat Demand

    - Cluster 3 has the highest heat demand by a large margin.
    - These factories have very high energy loads driven by a combination of multiple fuel types (mixed-fuel users).

Key Insight:

Cluster 3 is the highest-risk group for electrification, Cluster 3 combines multiple fuels and shows extremely high heat demand. These factories are likely large, energy intensive operations and represent the most impactful targets.

System Architecture

The K-Means clustering model is integrated into the same analytics pipeline and is designed to feed directly into Power BI for segmentation analysis and energy-planning dashboards.

After feature engineering and scaling, the entire dataset is passed through the PCA transformation and the K-Means model. The outputs are saved into a final analytical table that Power BI uses as its main source. In this table, several clustering-related columns are added:

- **PCA1, PCA2** – the two principal component coordinates for each facility, representing its overall energy pattern in a 2-D space (heat demand + fuel mix).
- **Cluster** – the numeric K-Means cluster ID assigned to each facility.
- **Cluster Name** – a human-readable description of the cluster (e.g., “Natural Gas Dominant).

This enriched clustering table is exported as CSV / stored in a database and then connected to Power BI. Power BI can then:

- Visualize the PCA scatter plot coloured by cluster_kmeans to show separation between factory segments.

Conclusion & Future Work

○ **Summary of Findings**

The decision tree emerged as the most effective model with strong interpretability and high predictive performance. It provides clear explanations of why certain facilities should electrify, making it suitable for policy and operational decision-making.

Strengths:

- Excellent recall and precision
- Transparent and interpretable
- Easy integration into downstream tools
- Minimal training time

Weaknesses:

- Sensitive to overfitting if hyperparameters are not carefully tuned
- Performance is slightly lower than ensemble models on perfectly balanced datasets
- Requires retraining if feature distributions shift significantly

K-Means can identify meaningful energy-use segments across industrial facilities. By combining the fuel proportion data, total energy consumption and PCA visualization, the six clusters have significantly diverged structures, making each cluster meaningful in terms of practical application, such as the high energy group, the mixed fuel group, and the gas-based group.

Strengths:

- Clearly segmented by fuel mix and heat demand
- Easy interpretation via PCA, Heatmap, and cluster summary.
- Fast processing, suitable for large amounts of factory data.

Weaknesses:

- Sensitive to the selection of k, requiring validation metrics by silhouette and cluster size
- Does not predict outcomes provides segmentation only, not recommendations
- Outliers can distort centroids if not properly scaled or filtered

---

○ **Future Work and Deployment Plan**

The following steps are recommended for future expansion of the classification model:

1. **Integrate with live data sources**
    - Automatic updating from emissions databases
    - Real-time monitoring dashboards in Power BI
2. **Improve model robustness**
    - Experiment with gradient boosting
    - Test cost-sensitive learning instead of SMOTE

The following steps are recommended to further improve and operationalize the K-Means clustering model:

1. **Integrate new factory data automatically**

   Enable routine updates of fuel ratios and heat demand from live data sources.
2. **Refresh clustering and PCA results on a schedule**

   Re-run PCA and K-Means monthly or quarterly to reflect changes in fuel behavior.
3. **Deploy to Power BI**

   Maintain an automated pipeline that sends updated PCA coordinates and cluster IDs to dashboards for real-time segmentation analysis.