Submitted in part fulfilment of the requirements for the degree of

MASTER OF SCIENCE IN BUSINESS ANALYTICS

# Visualizing, Classifying and Predicting Smart Meter Electricity / Gas Data for SMEs and industries

By

Swetha

Vijayanadhan

URN: 6833383

Faculty of Arts and Social Sciences

UNIVERSITY OF SURREY

September 2024

Word count: 14,448

**EXECUTIVE SUMMARY**

This dissertation aims to evaluate energy consumption patterns in industrial settings by applying advanced ML techniques to analyse electricity and gas consumption data from smart meters installed in SMEs and industrial settings across the UK. The growing electricity demand in the UK's industrial sector, driven by economic growth and technological advancements, presents challenges to energy sustainability. This research aims to develop advanced energy management strategies to optimize energy use, reduce costs, and support national policies, ultimately promoting sustainable practices.

As electricity demand rises, there is a need for efficient and sustainable energy management, particularly given the complexities of industrial consumption patterns. Traditional methods, such as surveys and basic statistical models, often fail to capture these complexities. In contrast, ML techniques provide advanced capabilities for visualizing, classifying, and predicting consumption patterns, offering deeper insights and supporting the development of targeted strategies to enhance energy efficiency.

The primary objective of this study is to leverage ML techniques to improve the understanding and management of industrial energy consumption. This objective is achieved through four specific goals: (1) visualizing energy consumption data using techniques like histograms, scatter plots, and correlation analysis to identify patterns, outliers, and relationships between variables; (2) classifying consumption patterns using ML algorithms such as ANN, SVM, DT to identify specific usage trends and anomalies; (3) developing predictive models using LR, GBR, RFR to forecast future energy consumption, aiding in efficient planning and management; and (4) providing practical recommendations for reducing energy consumption, improving energy efficiency, and promoting sustainable energy practices.

The study employs the CRISP-DM framework, guiding the research from understanding the business context to data preparation, modelling, evaluation, and deployment. The dataset consists of real-time data from smart meters in various SMEs and industrial settings across the UK, covering 396 rows and 46 columns. Key attributes include operational details (e.g., operating hours, peak usage times), energy efficiency initiatives, and responses to different tariff structures. Insights were further enriched through surveys distributed to a representative sample of businesses, providing a robust basis for analysing consumption behaviours.

Data preprocessing techniques, such as handling missing values through mean imputation, removing outliers using z-score analysis, and applying normalization and standardization, were employed to ensure data quality. Feature selection methods, including correlation analysis and RFE, identified the most influential variables affecting energy consumption, reducing data dimensionality and enhancing model performance.

Key findings from the study reveal the significant potential for improving energy management using advanced ML techniques. The ML models effectively identified consumption patterns, enabling targeted energy-saving interventions. For example, the SVM model achieved a high-test accuracy of 92.86%, outperforming other classifiers in categorizing complex datasets with multiple influencing factors, such as peak hours and weekend operations. Predictive models like GBR were particularly effective in forecasting future consumption due to their ability to handle non-linear relationships and complex interactions, achieving near-perfect accuracy and minimal error margins. These results highlight the robustness and applicability of these models in real-world industrial contexts.

The study emphasizes the importance of data quality on model performance. High-resolution data from smart meters, when properly anonymized, supports broader adoption of advanced ML models for energy management. Addressing data quality issues, such as outliers and missing values, is crucial for enhancing model accuracy and reliability. Integrating ML models into real-time energy management systems can improve operational efficiency, reduce costs, and support sustainable practices in SMEs and industrial settings.

The practical implications are substantial. SVM and GBR are particularly effective for developing targeted energy-saving measures and accurately forecasting future needs. By classifying users based on their energy consumption behaviours, energy providers and policymakers can implement more tailored demand-side management strategies, such as dynamic tariff plans or targeted efficiency programs, which can reduce peak demand, encourage conservation, and improve grid reliability. These strategies can enhance energy efficiency, lower operational costs, and contribute to national energy sustainability goals.

Future research should explore incorporating socio-economic data with ML models to provide a more holistic understanding of energy consumption patterns. Including factors such as business size, type of industry, and behavioural data could improve model transparency and interpretability, which is essential for building trust and encouraging adoption among stakeholders. Techniques like SHAP values or LIME can help explain the

decision-making processes of ML models, making them more accessible and actionable for non-experts.

This study demonstrates the value of applying advanced ML techniques to analyse smart meter data for optimizing energy management strategies in the UK's industrial sector. By providing actionable insights into consumption patterns and developing predictive models for future needs, the research supports more efficient and sustainable practices. The findings align with advancements in energy analytics, where ML and big data techniques enhance predictive accuracy, operational efficiency, and cost reductions. Previous studies indicate that providing real-time feedback on energy use can lead to substantial consumption reductions, supporting the integration of ML models with smart meter data to promote energy-efficient behaviours.

However, the study also identifies limitations that require further research. One key limitation is the need for more comprehensive insights into socio-economic factors influencing energy consumption. While the study focuses on quantitative data from smart meters, incorporating qualitative data could provide a deeper understanding of the human factors driving energy use. Additionally, the focus on specific ML models limits the exploration of hybrid or ensemble methods, which could further improve predictive performance by combining the strengths of multiple algorithms.

Furthermore, while the study addresses the challenges of high-dimensional data and complex non-linear relationships, it does not fully explore the scalability of these models in real-world applications, particularly in large-scale energy management systems. Future research should investigate how these models perform in diverse industrial contexts and explore new techniques to enhance model interpretability and scalability.

Overall, this research provides valuable guidance for policymakers, energy providers, and SMEs seeking to optimize their energy strategies, achieve greater sustainability, and respond effectively to the challenges of rising electricity consumption. By leveraging advanced ML techniques, the study lays the groundwork for developing more robust and adaptive energy management strategies that can adapt to changing consumption patterns and support long-term sustainability goals.

## DECLARATION OF ORIGINALITY

"I hereby declare that this thesis has been composed by myself and has not been presented or accepted in any previous application for a degree. The work, of which this is a record, has been carried out by myself unless otherwise stated and where the work is mine, it reflects personal views and values. All quotations have been distinguished by quotation marks and all sources of information have been acknowledged by means of references including those of the Internet. **I agree that the University has the right to submit my work to the plagiarism detection sources for originality checks."**

Date: 9th September 2024

Signature: Swetha Vijayanadhan

## TABLE OF CONTENTS

**LIST OF TABLES**

**LIST OF FIGURES**

**LIST OF ABBREVIATIONS**

| | |
|---|---|
| AI | Artificial Intelligence |
| ANN | Artificial Neural Network |
| AUC | Area Under the Curve |
| CRISP-DM | Cross-Industry Standard Process for Data Mining |
| DR | Demand response |
| DT | Decision Tree |
| FN | False Negatives |
| FP | False Positives |
| GBR | Gradient Boosting Regressor |
| IQR | Interquartile Range |
| KDD | Knowledge Discovery in Databases |
| KNN | K-Nearest Neighbours |
| LIME | Local Interpretable Model-agnostic Explanations |
| LR | Linear Regression |
| MAE | Mean Absolute Error |
| ML | Machine Learning |
| MSE | Mean Squared Error |
| OLS | Ordinary Least Squares |
| $R^2$ | R-Squared |
| RFE | Recursive Feature Elimination |
| RFR | Random Forest Regressor |
| RL | Reinforcement Learning |
| RMSE | Root Mean Squared Error |

| | |
|---|---|
| ROC | Receiver Operating Characteristic |
| SHAP | SHapley Additive exPlanations |
| SME | Small and Medium-sized Enterprises |
| SVM | Support Vector Machine |
| TN | True Negatives |
| TP | True Positives |
| TPB | Theory of Planned Behaviour |

**ACKNOWLWDGEMENTS**

I would like to express my deepest gratitude to my dissertation supervisor, Ali Emrouznejad, for his unwavering guidance and mentorship throughout this project. His calm manner, mentorship and expertise on the subject have significantly contributed to my learning experience.

I am also sincerely grateful to Professor Dr. Colin Fu for his teachings on machine learning, which have been invaluable during both the coding and theoretical aspects of this dissertation. Special thanks go to my friend, Naima Rashid, for her insightful counsel on various machine learning concepts.

Finally, I would like to acknowledge my mother, Mrs. Sunitha Viswanathan, and my partner, Jaganath Jayaprakash, for their continuous support and confidence throughout my master's program.

# CHAPTER 1 INTRODUCTION

Industrial energy consumption constitutes a substantial proportion of national energy use in developed countries. In the UK, electricity consumption has notably increased over recent decades due to economic development and population growth, particularly within both domestic and industrial sectors. Understanding the factors driving energy consumption is crucial for identifying usage patterns and behaviours associated with varying levels of energy demand. This project aims to apply ML techniques to visualize, classify, and predict electricity and gas consumption data from smart meters, specifically focusing on industrial users in SMEs settings.

By analysing detailed primary data, the study seeks to quantify consumption and explain variations based on industrial characteristics. For this purpose, ML models will be employed under both classification and regression categories. Classification algorithms such as ANN, SVM, and DT will be used to identify consumption patterns and categorize users based on their energy usage behaviours. These models are effective in handling complex, non-linear relationships and can provide accurate classification outcomes, making them suitable for segmenting users in diverse industrial and residential contexts. ML applications in the smart grid have demonstrated their ability to efficiently predict and classify energy consumption patterns, thereby supporting energy management and decision-making processes (Strielkowski et al., 2023).

For predicting future energy consumption, regression models such as RFR, GBR, and LR will be utilized. These regression-based approaches are known for their robustness in handling large datasets and capturing intricate relationships between multiple variables, which is crucial for predicting energy usage trends and anomalies. Visualization technique and anomaly detection, will be applied to highlight trends and unusual consumption patterns, aiding in the interpretation of model outputs and providing actionable insights. According to Gholami et al. (2021), smart meter data, when coupled with ML models, can provide deep insights into consumption patterns and support energy efficiency strategies by predicting demand and identifying peak usage times.

The insights derived from this analysis will inform strategies to enhance energy efficiency and reduce consumption, offering valuable guidance for policymakers, energy providers, and consumers. Ultimately, the project aims to foster sustainable energy management practices

by providing a clearer understanding of energy consumption dynamics and offering actionable recommendations to improve energy efficiency in SMEs environments.

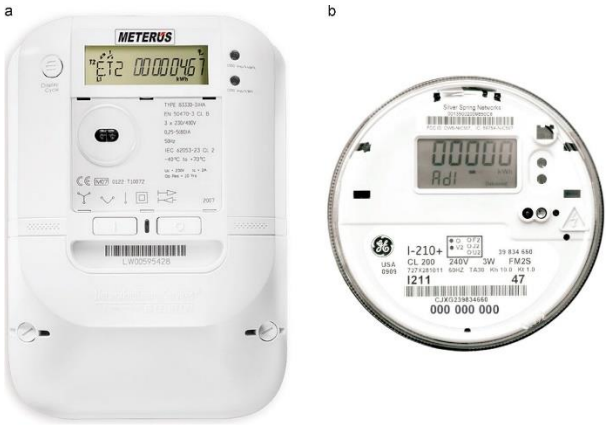


**Figure 1: An Examples of smart meters used in industrial energy management**

In the pursuit of understanding energy consumption patterns and the effectiveness of smart meter implementations among industrial users, a comprehensive dataset was acquired through a targeted survey. This survey, specifically designed for industrial users, captured detailed information on electricity usage, attitudes towards energy efficiency, and the impact of smart meter technology. The data collection process involved distributing structured questionnaires to a representative sample of industrial businesses, ensuring the dataset's robustness and relevance for analysis. Respondents provided insights into their experiences with smart meters, changes in electricity usage, and the effectiveness of different tariff structures. This rich dataset, derived from real-world responses, forms the foundation of the analysis presented in this dissertation, allowing for a nuanced exploration of energy usage behaviours and the potential for optimizing consumption through advanced ML approaches.

### 1.1 Background of the Study

Understanding the drivers of industrial electricity consumption is crucial for developing effective energy management strategies. Smart meters have revolutionized energy management for industrial users by providing detailed, real-time data on electricity and gas usage. Unlike traditional methods that rely on surveys and interviews, smart meters deliver granular insights into consumption patterns, which can be analysed using advanced ML techniques. These techniques, such as classification and regression analysis, allow for deeper insights and more accurate predictions of future consumption trends (Emrouznejad et al., 2023; Gholami et al., 2021).

Studies have highlighted the effectiveness of ML in analysing smart meter data, identifying key determinants of energy consumption, and classifying users based on their behaviours (Çınar et al., 2020; Oh & Min, 2024). Such insights are critical for informing strategies to enhance energy efficiency and reduce consumption. However, there are still gaps in understanding the specific factors influencing electricity and gas usage in SMEs and industrial settings. Much of the traditional research focuses more on suppliers and large-scale industries, often neglecting the nuanced needs of smaller industrial users (Gholami et al., 2020).

This study aims to fill this gap by comprehensively analysing smart meter data from SMEs in the UK. the study seeks to provide actionable insights for improving energy efficiency. Ultimately, this research aims to contribute to sustainable energy management practices by supporting policy development and encouraging energy conservation efforts at both the organizational and individual levels.

### 1.2 Problem Statement

The rising electricity consumption in the UK's industrial sector presents significant challenges for energy sustainability. As industrial activities expand and become more complex, the demand for electricity grows, placing increased pressure on energy resources and the environment. Traditional methods of analysing energy consumption, such as surveys and basic statistical models, often fail to capture the intricate interplay of industrial behaviours, and technological advancements influencing electricity use (Emrouznejad et al., 2023). Addressing these challenges requires advanced analytical approaches capable of managing the high-dimensional and non-linear nature of energy data.

ML techniques offer promising solutions for visualizing, classifying, and predicting electricity and gas consumption patterns from smart meter data in SMEs and industrial settings. Smart meters provide detailed, real-time data on energy usage, creating opportunities to uncover hidden patterns and generate actionable insights. However, the complexity and volume of this data necessitate sophisticated analytical tools to extract meaningful information (Gholami et al., 2021). ML models, including classification algorithms and regression analysis, have proven effective in identifying patterns and predicting outcomes in energy consumption (Oh & Min, 2024).

Despite their potential, the adoption of ML models for predicting energy consumption using smart meter data remains limited. Traditional methods often struggle with the timeliness and

precision required for effective energy management. In contrast, ML techniques can efficiently handle large datasets and complex relationships. Integrating visualization techniques with these models further enhances the understanding of energy consumption patterns, enabling stakeholders to intuitively explore trends and anomalies (Gholami et al., 2020).

This project aims to develop ML-based models to analyse smart meter data, providing actionable insights for optimizing energy management strategies. By doing so, it seeks to support the development of more efficient and sustainable energy practices in the industrial sector, aiding policymakers, energy providers, and SMEs in making informed decisions (Emrouznejad et al., 2023).

### 1.3 Research Objectives

The main objective of this study is to leverage ML techniques to visualize, classify, and predict electricity and gas consumption patterns using smart meter data for SMEs and industrial settings. Given the increasing electricity consumption in industrial sectors in developed countries, especially the UK, understanding consumption patterns is crucial. The specific objectives of the study are outlined below:

-- Please note that these objectives will be referenced as objective 1, objective 2, objective 3, and objective 4 in subsequent sections of the report.

***OBJECTIVE 1:*** *Visualizing energy consumption data using techniques such as histograms, box plots, scatter plots, outlier detection, and correlation analysis is vital for effective energy management. Histograms help identify consumption patterns, while box plots summarize data distribution and highlight anomalies. Scatter plots and correlation analysis reveal relationships between variables, and outlier detection identifies unusual usage patterns. These visual tools empower stakeholders to optimize energy efficiency and make informed decisions. This approach enhances the ability of organizations to monitor their energy consumption more effectively, leading to better resource management and cost savings.*

***OBJECTIVE 2****: Utilizing ML algorithms like ANN, SVM, and DT to classify energy consumption patterns allows for more precise categorization. This classification facilitates targeted energy-saving interventions by identifying specific usage trends and anomalies, thereby optimizing energy management strategies and reducing overall consumption.*

***OBJECTIVE 3:*** *Developing predictive models using LR, GBR, RFR techniques to forecast future energy consumption aids in efficient energy planning and management. These models provide accurate projections, enabling organizations to anticipate demand, optimize resource allocation, and implement effective energy-saving strategies.*

***OBJECTIVE 4:*** *Developing practical recommendations based on the findings to offer strategies for reducing energy consumption, improving energy efficiency, and adopting sustainable energy practices.*

### 1.4 Significance of the Study

The importance of this study stems from its potential to improve energy efficiency and sustainability by using ML to analyse smart meter data from industrial users. Accurate forecasting of energy consumption helps prevent shortages and optimize resource allocation, supporting both operational efficiency and strategic planning. Oh and Min (2024) highlight that ML provides detailed insights into energy consumption patterns, enabling targeted energy-efficient strategies that reduce costs and support national energy policies.

The study's predictive models facilitate timely energy planning, crucial for policy formulation and preventing energy crises. Gholami et al. (2021), emphasize segmenting users by consumption patterns to develop targeted strategies, while Emrouznejad et al. (2023) stress the need for comprehensive analysis to promote sustainable energy behaviours. Using smart meter data to identify high-consumption patterns enables practical energy-saving measures.

Furthermore, Emrouznejad et al. (2023) and Gholami et al. (2020) underline the importance of high-resolution data for effective strategies and emphasize addressing privacy concerns to ensure consumer trust. This study’s approach to anonymizing data, while retaining its utility, supports widespread adoption of smart meter technologies and contributes to more efficient and sustainable energy management practices.

### 1.5 Scope of the Study

This study investigates the application of machine learning techniques to analyse electricity and gas consumption data from smart meters installed in SMEs and industrial settings across the UK. The research focuses on understanding energy consumption patterns by examining demographic, that influence usage. Advanced ML models, including regression and classification algorithms, will be applied to smart meter data to identify consumption patterns, segment users, and predict future energy demands. The insights gained will help

optimize resource allocation, improve energy management, and support the development of energy-efficient strategies for SMEs and industries.

### 1.6 Organization of the Dissertation

The structure of this dissertation is as follows: Chapter 1 introduces the topic, highlighting the significance of predicting electricity and gas consumption using smart meter data in SMEs and industries, and outlines the objectives. Chapter 2 provides a review of the related literature on smart meter data analysis and machine learning algorithms. Chapter 3 describes the methodology, including the selection of algorithms and data analysis techniques. Chapter 4 presents the analysis and results, while Chapter 5 discusses the findings and their implications. Finally, Chapter 6 concludes with a summary of the findings, an evaluation of the objectives, a discussion of the limitations, and recommendations for future research.

## CHAPTER 2 LITERATURE REVIEW

Smart meters have revolutionized energy management by providing detailed, real-time data on electricity and gas consumption. This rich data source is critical for understanding and optimizing energy usage patterns, especially for SMEs and industries. This study uses primary data from smart meters installed in industrial settings, supplemented by survey data from industrial users. The primary objective of this literature review is to explore how advanced ML techniques can be applied to visualize, classify, and predict energy consumption using these comprehensive datasets. Key techniques include Classification and Regression algorithms, such as ANN, DT, SVM, LR, GBR, and RFR.

Recent advancements underscore the importance of integrating smart meter data with ML to enhance predictive accuracy and provide actionable insights for energy conservation. This integration allows for more effective energy-saving interventions, optimizing operational efficiency and reducing costs for businesses (Smajla et al., 2023). Conversely, challenges such as ensuring data quality, achieving high model accuracy, and addressing privacy concerns remain critical considerations in the application of these techniques.

Moreover, analysing smart meter data can reveal critical insights into the behavioural patterns of energy consumption. Studies have shown that providing real-time feedback to consumers can lead to substantial reductions in energy use, highlighting the potential of smart meters in promoting energy-efficient behaviours (Gholami et al., 2021). By leveraging these insights, SMEs and industries can tailor their energy management strategies to encourage more sustainable practices.

The integration of advanced ML techniques with smart meter data, particularly when combined with survey data from industrial users, presents a promising avenue for enhancing energy management. This literature review synthesizes current methodologies and applications, with a focus on addressing challenges such as data quality, model accuracy, and privacy concerns. Through this review, we aim to contribute to the advancement of energy analytics and its practical applications in SME and industrial contexts, providing a foundation for future research and development in this area (Jovicic et al., 2023; Buri et al., 2024).

## 2.1 Contextual Background for Smart Meter Data Analytics

In recent years, the integration of smart meter data with advanced ML techniques has emerged as a pivotal area of research in energy management. This intersection of technologies not only optimizes energy consumption but also contributes to the broader goals of sustainability and operational efficiency, particularly for SMEs and industries.

The study by Oh and Min (2024) highlights the critical importance of precise energy consumption data at the facility level for the manufacturing sector. The authors emphasize the role of ML in predicting annual energy consumption, considering industry-specific characteristics such as electricity usage and employee size. These insights are crucial for SMEs, which often face unique challenges in balancing energy efficiency with operational demands.

Building on this, Buri et al. (2024) and Çınar et al. (2020) conducted a comprehensive bibliometric review of smart and sustainable energy consumption, identifying emerging trends and key research areas. Their study underscores the growing significance of integrating smart solutions with sustainable energy resource consumption. The findings reveal a clear trend towards incorporating advanced technologies such as AI and big data analytics to enhance energy efficiency and reduce carbon footprints. This aligns with the objectives of my research, which aims to leverage ML models—including ANN, DT, SVM, and regression methods to analyse smart meter data and provide actionable insights for energy conservation in SMEs and industries.

Moreover, the research highlights the challenges associated with the variability of energy consumption and the need for real-time data to optimize energy management strategies effectively. The integration of IoT and smart grid technologies, as discussed by (Buri et al., 2024), plays a crucial role in this regard, enabling more accurate predictions and efficient energy use.

## 2.2 Theoretical Frameworks

Smart meter data analysis is pivotal for understanding and optimizing energy consumption patterns in diverse settings, including SMEs and industrial sectors. This literature review synthesizes insights from economic theories, behavioural theories, and data-driven models relevant to smart meter data analysis, drawing from recent studies and theoretical perspectives.

### 2.2.1 Economic Theories in Energy Consumption

Economic theories provide fundamental insights into how pricing mechanisms and cost-benefit analyses influence energy consumption behaviours. **Price elasticity of demand** measures consumers' responsiveness to changes in energy prices. According to Swan and Ugursal (2009), higher electricity prices generally lead to reduced consumption as consumers seek to mitigate costs through efficiency measures or behavioural adjustments. This principle underscores the importance of designing effective pricing strategies aligned with energy conservation goals, particularly in industrial and SME contexts where energy costs are significant.

**Cost-benefit analysis** complements price elasticity by evaluating the financial feasibility of energy-saving investments facilitated by smart meter installations. For instance, Frederiks et al. (2015) highlight the critical role of assessing total implementation costs against anticipated savings from reduced energy consumption. This approach provides essential insights into the economic viability of adopting smart technologies for energy management. Understanding these economic factors enables SMEs and industries to make informed decisions about investing in smart meter technologies and related energy-saving measures. Additionally, incorporating these economic considerations can lead to more tailored and effective energy pricing strategies, which are crucial for both cost reduction and energy efficiency in the industrial sector.

### 2.2.2 Behavioural Theories and Energy Consumption

Behavioural theories provide significant insights into how psychological and social factors influence energy consumption decisions within industrial and commercial settings. **The TPB proposed by** Ajzen (1991), can be applied to understand these dynamics in a business context, where organizational attitudes, perceived norms within the industry, and perceived behavioural control (such as availability of resources and expertise) influence energy management decisions. Koumentakos (2022) explores how organizational culture and leadership attitudes towards sustainability significantly affect energy consumption behaviours in industrial settings. This study found that companies with a strong internal commitment to energy efficiency, driven by leadership and organizational norms, are more likely to adopt comprehensive energy-saving measures.

### 2.2.3 Data-Driven Approaches to Visualizing and Predicting Energy Consumption

Advancements in ML and big data analytics have revolutionized the processing and interpretation of smart meter data for energy consumption analysis. Techniques such as ANN, SVM, and DT enable the classification of consumption patterns and prediction of future energy usage trends. For instance, Beckel et al. (2014) demonstrated the efficacy of ANN in accurately forecasting electricity consumption patterns based on historical data, enabling proactive energy management strategies.

DT as discussed by Çınar et al. (2020) and Emrouznejad et al. (2023), are particularly noted for their interpretability, making them effective in identifying influential factors such as weather conditions, operational schedules, and equipment usage patterns that impact energy consumption variability. This interpretability is crucial for industrial and SME stakeholders who require clear insights into how specific variables affect their energy usage.

On the other hand, SVMs excel in classifying energy consumption behaviours across different industrial sectors, facilitating targeted interventions aimed at optimizing energy efficiency (Gholami et al., 2021). Collectively, these ML techniques provide robust tools for SMEs and industries to visualize energy data, identify patterns, and predict future consumption, thereby enabling more effective energy management strategies.

## 2.3 Applications of ML Algorithms in Energy Prediction

Machine learning (ML) algorithms are divided into three main categories: (1) supervised, (2) unsupervised, and (3) RL (Figure 2) (Çınar et al., 2020). The goal is to demonstrate the potential complexity of these structures and highlight the most used learning techniques. Different algorithms can be combined to enhance classification performance. Additionally, some ML algorithms can be applied in both supervised and unsupervised learning contexts.

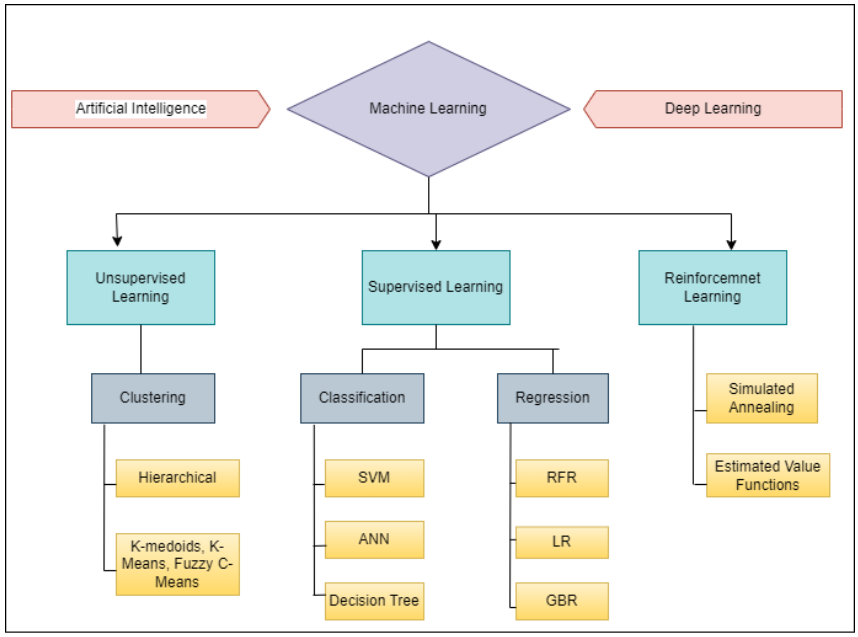


**Figure 2:** Classifications and Regressions within ML Techniques.

### 2.3.1 Classifying Energy Consumption

**ANN:** ANNs are particularly adept at modelling complex, non-linear relationships within data, making them ideal for tasks such as predicting future energy consumption based on historical patterns. According to Çınar et al. (2020), ANNs have been successfully used in industrial applications, particularly for predictive maintenance and the classification of energy consumption patterns. Their ability to handle large and complex datasets makes them well-suited for identifying intricate patterns in energy usage across various operational contexts within industries.

**SVM:** SVMs are powerful classifiers, especially when dealing with high-dimensional data. They are effective in classifying energy consumption patterns by finding the optimal hyperplane that separates different classes of energy users. Çınar et al. (2020) highlight the application of SVMs in industrial settings, where they are used to classify states of machinery based on operational data. This approach can be adapted to categorize different energy consumption levels in SMEs, facilitating targeted energy management strategies.

**DT:** DT offer an interpretable framework for decision-making by recursively splitting data into subsets based on significant features. Çınar et al. (2020) emphasize the utility of DT in industrial contexts for both classification and regression tasks. In energy consumption classification, DT can help identify key factors influencing energy use, allowing businesses to derive actionable insights and optimize their energy strategies accordingly.

### 2.3.2 Forecasting Energy Consumption

**LR:** Çınar et al. (2020) discuss the use of LR in the context of predictive maintenance for industrial machinery. The study highlights how LR is used to model the condition of well-functioning industrial machinery, particularly for identifying and predicting changes or shifts (known as concept drifts) in continuous data streams. This approach involves modelling using real-world datasets to understand and predict maintenance needs and optimize equipment usage by identifying patterns and deviations in equipment behaviour over time. LR is noted for its simplicity, making it a common choice for initial modelling and providing baseline comparisons against more complex machine learning technique.

**GBR:** GBR sequentially builds models, each correcting errors from previous ones, enhancing predictive accuracy by minimizing residuals. It excels in handling complex datasets with nonlinear relationships. The effectiveness of GBR in predicting energy demand by capturing intricate variable interactions is highlighted, making it ideal for forecasting in SMEs and industries (Otchere et al., 2022).

**RFR:** Smajla et al. (2023) highlight the application of RFR in energy consumption prediction. The study explains RFR is utilized due to its capacity to manage a diverse array of predictors and its robust resistance to overfitting, making it particularly valuable for forecasting energy patterns in dynamic and complex environments such as industrial settings. The algorithm's ability to average multiple decision trees helps in providing reliable predictions for both linear and non-linear relationships, which is crucial for accurate energy demand forecasting and optimizing energy management strategies.

## 2.4 Applications of ML in Smart Meter Data Analysis

Using machine learning techniques to analyse smart meter data has been very effective in various energy management applications, particularly for SMEs and industries. These techniques facilitate better decision-making, enhance energy efficiency, and contribute to sustainable energy practices. This section explores specific applications of these techniques and their practical implications.

### 2.4.1 DR Programs

DR programs are essential for balancing the supply and demand of energy. ML algorithms have been instrumental in optimizing these programs by predicting energy demand in real-time and adjusting consumption accordingly. For instance, in the work by Chaudhari et al. (2019) ML models were used to analyse smart meter data, enabling the prediction of peak

demand periods. By integrating these predictions with DR strategies, industries can adjust their energy consumption in response to price signals, thereby reducing costs and alleviating stress on the power grid during peak times. This application is particularly beneficial for energy-intensive industries where managing demand response can lead to significant cost savings.

### 2.4.2 Load Forecasting

Accurate load forecasting is a critical application of ML in energy management. It involves forecasting future energy demand by utilizing historical data and incorporating other relevant factors, such as weather conditions and operational schedules.

**Short-term Load Forecasting:** According to Koumentakos (2022) ML models like SVM and ANN have been successfully applied to short-term load forecasting, achieving high accuracy in predicting daily energy demand. These models are crucial for utilities and industries to plan energy distribution more effectively and to avoid the costs associated with over or underestimating energy needs.

**Long-term Load Forecasting:** Long-term forecasting, as explored by Watson et al. (2012), uses extensive historical data to predict future trends in energy demand. Techniques such as DT are particularly useful in identifying the key factors that drive long-term energy consumption. This information is vital for industries when making strategic decisions regarding infrastructure investments and energy procurement. For SMEs, long-term load forecasting helps in planning for future energy needs, allowing for better resource allocation and investment in energy-efficient technologies.

## 2.5 Empirical Research

Empirical research on smart meter data analytics for SMEs and industries highlights the significant role of various ML algorithms in analysing, classifying, and predicting energy consumption patterns. Studies consistently demonstrate that integrating multiple ML techniques, such as ANN, SVM, DT, GBR, RFR, and LR, can significantly enhance predictive accuracy and provide actionable insights for energy management.

### Integration of Machine Learning Models for Predictive Analytics

Gholami et al. (2021) and Chinnathai et al. (2023) have demonstrated the utility of models such as SVM and ANN in predicting energy consumption patterns and managing non-linear relationships within high-dimensional datasets. These studies illustrate that SVM and ANN

are effective in handling complex data structures typical in energy datasets. However, they do not explore the potential of integrating these methods with ensemble models like RFR and GBR, which have shown greater predictive accuracy and robustness in other fields. Integrating ensemble methods could address some of the limitations of standalone models, such as overfitting and sensitivity to noise in the data, which are common issues in energy consumption datasets.

Çınar et al. (2020) highlight the strengths and limitations of various machine learning models for predictive maintenance and energy management, particularly DT and LR. The study supports the effectiveness of DT for predictive maintenance due to its interpretability and ability to handle non-linear relationships. DT models can simplify complex decision-making processes by mapping out potential outcomes based on variable inputs, making them useful tools for transparency and understanding. However, the study also notes that DTs do not necessarily provide the highest level of accuracy compared to other ML models. Furthermore, Çınar et al. (2020) do not explore combining DT with advanced ensemble methods like RFR or GBR, which could potentially enhance predictive performance by aggregating multiple weak models into a more robust one. This gap presents an opportunity for future research to improve model accuracy and offer more reliable energy management solutions.

Similarly, the Çınar et al. (2020) discusses the limitations of LR in capturing non-linear relationships and addressing scalability issues in large-scale, real-time data scenarios typical in industrial applications. The study points out that LR is often insufficient on its own due to these constraints and suggests that it could benefit from integration with ensemble methods like RFR or GBR to improve robustness and predictive accuracy. Moreover, there is an insufficient exploration of LR's role in hybrid modelling approaches, where its integration with more complex algorithms could potentially enhance the overall effectiveness of predictive maintenance and energy management strategies. Both models, DT and LR, therefore, present gaps in their current applications that could be addressed through more sophisticated combinations with other methods, offering a direction for future research to advance energy management practices.

**Application of Ensemble Techniques**

While the application of individual ML models has been well-documented, there is limited exploration in the literature on combining these models with ensemble methods for energy

management. Ensemble methods like RFR and GBR have been proven to handle complex datasets more effectively by reducing overfitting and improving generalization across various contexts (Gholami et al., 2020). These techniques aggregate the predictions of multiple models to produce a final prediction, thereby improving the overall model stability and accuracy. However, few studies have considered their application in energy consumption forecasting using smart meter data. Gholami et al. (2020) briefly mentions the combination of different ML models for better prediction, but their study does not deeply delve into the use of advanced ensemble techniques like GBR, suggesting an area ripe for further investigation.

**Clustering Techniques and Their Limitations**

Flath et al. (2012) and Seixas et al. (2015), the application of clustering techniques has been extensively explored to segment customers according to their consumption behaviours. For instance, Flath et al. (2012) demonstrated the use of clustering algorithms to analyse smart metering data, identifying distinct customer segments based on dynamic load patterns. This approach has proven valuable for utility companies to design targeted services and tariff plans tailored to different customer groups. Similarly, Seixas et al. (2015) employed clustering methods to fuse smart meter data with additional survey data, aiming to enhance the understanding of household electricity consumption patterns and support more effective demand-side management strategies.

However, both studies primarily focus on clustering as a tool for segmentation rather than prediction. Flath et al. (2012) emphasize the role of clustering in understanding customer diversity and designing specific interventions, but they do not investigate the integration of clustering techniques with other machine learning models, such as regression or ensemble methods, that could provide predictive capabilities. Similarly, Seixas et al. (2015) discusses the advantages of combining different data sources to enrich clustering insights but do not address the potential benefits of using clustering in conjunction with predictive models to enhance forecast accuracy.

While clustering effectively groups customers with similar behaviours, neither study delves into how these segments could be used as a basis for predictive analytics. Incorporating additional machine learning techniques, such as ensemble models, could provide a more comprehensive analytical framework that includes both segmentation and prediction, thus addressing the gaps noted in both Flath et al. (2012) and Seixas et al. (2015). This represents

a significant opportunity for future research to develop more robust energy management strategies by combining the strengths of clustering and predictive models.

**Hybrid Models and the Potential for Improved Energy Management**

Koumentakos (2022) emphasizes the potential of hybrid models in industrial energy management, which combine multiple ML techniques to enhance decision-making and forecasting. While hybrid models leveraging different algorithms are becoming more prevalent in energy analytics, the study does not deeply explore their specific application to smart meter data or the integration of advanced ensemble methods like RFR and GBR. Chinnathai et al. (2023) further highlight the use of predictive analytics through ML algorithms like SVM and ANN in energy management, but they too fall short of examining how these models could be enhanced through integration with ensemble methods. Similarly, Gholami et al. (2021) discusses various ML techniques, including SVM, ANN, and DT, for analysing smart meter data, focusing on the incorporation of socio-economic variables to improve predictive accuracy. However, they do not explore the potential benefits of using ensemble methods like RFR or GBR. Stinson (2015) demonstrates the use of ANN in predictive maintenance but does not consider extending these models for energy consumption forecasting or integrating them with ensemble methods. Collectively, these studies suggest that while individual ML techniques are effective in specific applications, there is a significant gap in leveraging hybrid models and ensemble methods to provide more robust and comprehensive energy management solutions, especially when dealing with complex smart meter data. This points to a need for future research to integrate these advanced techniques for improved predictive and prescriptive analytics in energy management for industries and SMEs.

**Combining Clustering with Predictive Analytics**

Flath et al. (2012) and Seixas et al. (2015) focus on the use of clustering techniques, such as k-means and hierarchical clustering, to segment customers based on dynamic load patterns and consumption behaviours. These studies highlight the value of clustering for utility companies to design customer-specific services and tariff plans, allowing for more targeted energy management strategies. However, both studies primarily examine clustering in isolation, without exploring its integration with predictive models like RFR or GBR. Gholami et al. (2021) extends this by suggesting that combining clustering techniques with ML models can enhance the predictive accuracy of energy consumption forecasts. The

integration of clustering as an initial step in the data preparation phase could help identify specific patterns and groups within the data, which could then serve as inputs for more advanced predictive models, improving both the granularity and robustness of the predictions. However, the full potential of this integration is not fully developed in the current literature, leaving room for further research to explore how such a combination could provide more effective and targeted energy management solutions.

**Advanced Techniques for Improved Energy Management**

Wanasinghe et al. (2020) and Stinson (2015) propose using advanced ML techniques, including GBR and deep learning methods, for optimizing energy consumption predictions. These techniques offer significant potential because of their ability to manage large volumes of complex data and model intricate relationships between multiple variables. However, the studies do not extend these methods to cover the integration with existing industrial data and socio-economic variables, limiting their applicability in real-world energy management scenarios.

Furthermore, Chinnathai (2023) and Gholami et al. (2020) advocate for a more integrated approach that combines both predictive and prescriptive analytics, suggesting that such methods could offer more actionable insights. While they focus on standalone ML models, the studies do not fully explore the potential benefits of integrating these models with advanced ensemble methods or hybrid models, such as those discussed by (Koumentakos, 2022).

In conclusion, the future studies should focus on integrating various ML algorithms, such as ANN, SVM, DT, LR, RFR, and GBR, to develop more comprehensive models for analysing and forecasting energy consumption. This integrated approach aims to optimize energy use, accurately forecast demand, and provide actionable insights for SMEs and industries, ultimately enhancing energy management strategies and promoting sustainability. Furthermore, there is a need to investigate how these models can be applied in practical, real-world scenarios to validate their effectiveness and scalability, ensuring that the findings can be generalized across different contexts and industries.

## 2.6 Identifying and Addressing Gap in Literature

Several studies have examined the use of machine learning models in energy management, highlighting the effectiveness of individual models such as ANN, SVM, and DT in predicting energy consumption patterns and managing non-linear relationships within high-

dimensional datasets (Gholami et al., 2021; Chinnathai, 2023). However, a significant gap remains in exploring the integration of these methods with advanced ensemble techniques such as RFR and GBR to enhance predictive accuracy and robustness (Çınar et al., 2020). While these ensemble methods have shown potential in other domains for reducing overfitting and improving generalization, their application in energy management, particularly with smart meter data, is underexplored (Wanasinghe et al., 2020).

The literature also reveals a lack of research into combining clustering techniques with predictive models for more targeted energy management strategies (Flath et al., 2012; Seixas et al., 2015). Although clustering has been extensively used for segmenting customers based on consumption behaviours, the integration of these clusters with regression or ensemble models to enhance forecast accuracy has not been thoroughly investigated (Flath et al., 2012). Furthermore, while some studies suggest the incorporation of socio-economic variables into energy consumption models, the practical implementation of such integrations remains limited, indicating a need for more comprehensive research that combines multiple data sources and advanced modelling techniques (Gholami et al., 2021; Koumentakos, 2022).

Additionally, existing research often focuses on theoretical or standalone applications of machine learning models without sufficiently addressing their scalability and applicability in real-world, large-scale energy management systems (Chinnathai, 2023; Stinson, 2015). There is also a notable gap in frameworks that combine predictive and prescriptive analytics, particularly in models that integrate energy forecasting with predictive maintenance (Gholami et al., 2021). While studies such as Koumentakos (2022) discuss the potential of hybrid models in enhancing decision-making and forecasting in industrial energy management, they do not delve into specific applications for smart meter data or the integration of advanced ensemble methods.

Overall, the literature indicates that future research should focus on integrating various machine learning algorithms, such as ANN, SVM, DT, LR, RFR, and GBR, to develop more holistic models for energy management. This integrated approach would address the gaps related to limited comparative studies, underexplored data preprocessing techniques, and the lack of frameworks that combine multiple analytical methods for more robust and comprehensive energy management solutions.

## CHAPTER 3 METHODOLOGY

This chapter outlines the research methodology employed to analyse smart meter electricity and gas data to optimize energy management strategies for SMEs and industries. The methodology follows the CRISP-DM framework, providing a structured approach to guide the research through the stages of understanding the business context, understanding the data, preparing the data, modelling, evaluation, and deployment. Each section of this chapter details the specific techniques and processes used, including justifications for selecting advanced machine learning algorithms such as SVM, ANN, DT, LR, RFR and GBR for effective energy consumption analysis.

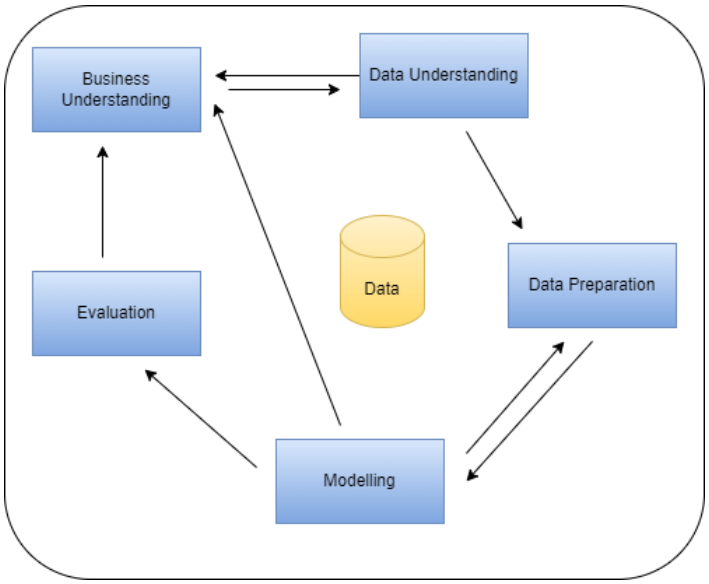


**Figure 3: CRISP-DM Process**

### 3.1. Business Understanding

The objective of this study is to improve the visualization, classification, and prediction of smart meter electricity and gas data for SMEs and industries. By applying machine learning techniques, the study aims to identify energy consumption patterns, forecast future needs, and enhance energy management strategies. This approach enables businesses to make more informed decisions, optimize efficiency, and reduce costs, contributing to sustainable energy practices.

The study addresses the challenge of rising electricity consumption in the UK's industrial sector, where traditional methods struggle to manage energy effectively due to a lack of real-time insights. By leveraging advanced machine learning models, this research provides a more precise understanding of energy usage, enabling businesses to analyse data in real-

time, uncover patterns, and predict future trends. This helps bridge the gap between data and actionable insights, promoting innovative energy management solutions for SMEs and industries in the UK.

### 3.2 Data Understanding

This research uses data from two key sources: the "SME Pre-Trial Survey" and the "SME Post-Trial Survey." These surveys provide detailed insights into the electricity and gas consumption patterns of SMEs across various sectors. The aim is to analyse how SMEs manage their energy use and respond to interventions like smart meters and dynamic tariff structures.

The primary dataset consists of real-time data from smart meters installed in SMEs and industrial settings, capturing detailed information on electricity and gas consumption. This dataset includes 396 rows and 46 columns with numerical variables, such as energy type, source, employee size, and consumption volume, which are crucial for understanding sector-specific usage patterns, anomalies, and correlations. Additionally, survey data offers insights into user attitudes toward energy efficiency and behavioural changes prompted by smart meter adoption. The Data Understanding phase involves collecting, exploring, and analysing this data to assess its quality and suitability for the study. Techniques such as descriptive statistics and data visualization are used to evaluate the data's structure and align it with best practices for comprehensive analysis (Schröer, Kruse & Gómez, 2021).

**Table 1: Overview of Dataset Components and Attributes**

| **Component** | **Description** | **Attributes** |
|---|---|---|
| Operational Details | Data about the operating hours and days of the business. | - Operating Hours: Typical daily operating hours (e.g., 8-10 hours, 18-24 hours) [Q61021].<br>- Peak Hours: Inclusion of peak hours (5 pm to 7 pm) [Q610210].<br>- Weekend Operation: Weekend working days (e.g., Saturday only, Saturday and Sunday) [Q61022, Q61024]. |

| | | |
|---|---|---|
| Energy Efficiency Initiatives | Engagement in energy-saving initiatives and future plans | - Own Electricity Generation: Use of renewable energy sources (e.g., Solar panels, Wind turbines) [Q350].<br>- Plans for Future Investments: Future plans for energy efficiency investments (Yes/No) [Q370]. |
| Tariff and Billing Impact | Understanding and response to different tariff structures and their impact on behaviour. | -Effectiveness of Smart Meter: Perceptions of how smart meters have helped reduce usage [Q8000, Q8010, Q8020].<br>- Monitor Usage: Frequency and ease of using electricity monitors provided during the trial [Q8008, Q8030, Q8040]. |

### 3.2.1 Data Preprocessing

The preprocessing of the dataset involved several steps to ensure data quality and consistency:

**Handling Missing Values:** The dataset initially contained a significant number of missing values across various columns. Missing values were filled using mean imputation to maintain the data's overall distribution and reduce potential biases in the analysis (Schröer, Kruse & Gómez, 2021).

**Removal of Outliers:** Outliers were identified and removed to prevent skewed results and to enhance the accuracy of predictive models. This step was essential for maintaining the integrity of the dataset and ensuring reliable outcomes in subsequent modelling phases (Schröer, Kruse & Gómez, 2021).

## 3.3 Data Preparation

The Data Preparation phase involves transforming the raw data from smart meters and surveys into a format suitable for analysis and modelling. This phase ensures data quality, consistency, and reliability, which are essential for building effective predictive models. Effective data preparation is considered one of the most critical tasks in the data mining process due to its significant impact on the quality of the final analysis (Schröer, Kruse & Gómez, 2021).

### 3.3.1 Handling Missing Values and Outliers

Handling missing values and detecting outliers were critical steps covered in the Data Understanding phase. For details on how missing values were managed using mean imputation and how outliers were identified and addressed through z-score analysis and the IQR method (Schröer, Kruse & Gómez, 2021).

### 3.3.2 Normalization and Standardization

Normalization and standardization were applied to prepare the data for modelling

- **Normalization**: The data was rescaled to a range between 0 and 1 to handle varying magnitudes of different features, Normalization is often discussed in the context of preparing data for machine learning algorithms sensitive to input scales, such as KNN and neural networks (James et al., 2013).
- **Standardization:** Data was transformed to have a mean of zero and a standard deviation of one. This step is crucial for enhancing the performance of machine learning algorithms like SVM and ANN, which perform better with normalized data, which require features to be on a comparable scale for optimal performance (James et al., 2013).

### 3.3.3 Feature Selection and Engineering

Feature selection techniques, such as **correlation analysis** and **RFE**, were employed to pinpoint the most influential variables affecting energy consumption. This approach reduces data dimensionality by focusing on features with the highest predictive power, enhancing model accuracy and efficiency. Additionally, feature engineering was carried out to generate new variables that more effectively capture complex relationships within the dataset, which is crucial for improving the performance of machine learning models used for energy management analysis (Mohajeri et al., 2020).

### 3.3.4 Data Transformation

Data transformation techniques were applied to adapt the dataset to the specific requirements of different machine learning algorithms:

- **Logarithmic Transformations**: These were used to reduce skewness in variables with long-tailed distributions, making the data more suitable for modelling, where log transformations are applied to improve model performance by normalizing the data distribution (James et al., 2013**)**.

- **Binning**: Continuous variables were converted into categorical ones to improve model performance for specific analyses (James et al., 2013).

### 3.3.5 Data Splitting

The dataset was divided into training, validation, and testing sets to ensure robust model development and evaluation:

- Training Set: 70% of the data was used for training the models to learn patterns and relationships.
- Validation Set: 15% of the data was reserved for hyperparameter tuning and optimizing model performance, helping to prevent overfitting.
- Testing Set: The remaining 15% of the data was used to evaluate the final model's performance on unseen data, ensuring generalizability and accuracy in real-world applications (James et al., 2013).

## 3.4 Modelling

### 3.4.1 Selection of Models

**ANN:** ANN are effective in modelling complex, non-linear relationships within data. In the context of energy consumption, these relationships are often influenced by multiple factors, such as weather conditions, time of use, and operational schedules. ANN's ability to learn from large amounts of data and make accurate predictions makes it particularly useful for predicting future energy consumption based on historical patterns (Tatachar, 2021).

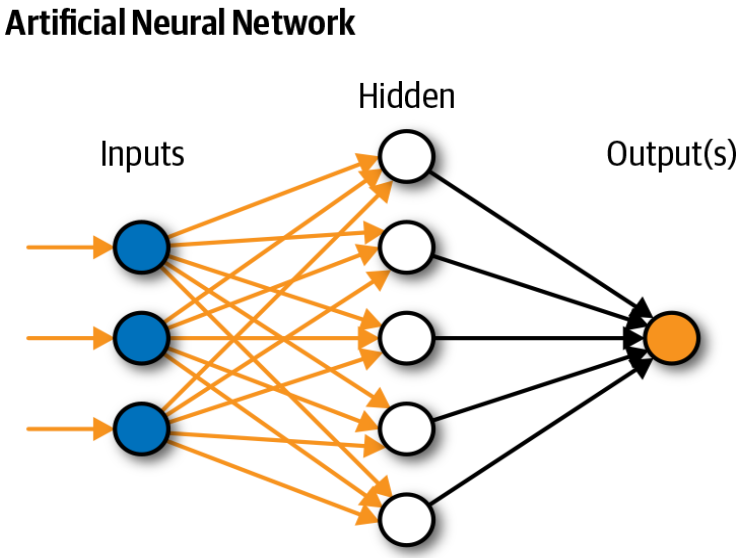


**Figure 4:** ANN Algorithm

**SVM:** SVM are powerful for both classification and regression tasks, particularly when there are distinct patterns in the data that separate different classes of energy consumers. SVMs are robust against overfitting and suitable for high-dimensional datasets, making them useful for understanding complex relationships in energy consumption data (James et al., 2013; Tatachar, 2021).

**DT:** DT are highly interpretable and provide clear decision rules based on different energy consumption factors. They are beneficial for understanding the hierarchical structure of decisions affecting energy use. Decision Trees effectively identify the most critical variables influencing energy consumption patterns and provide visual insights that are easy to interpret (James et al., 2013; Tatachar, 2021).

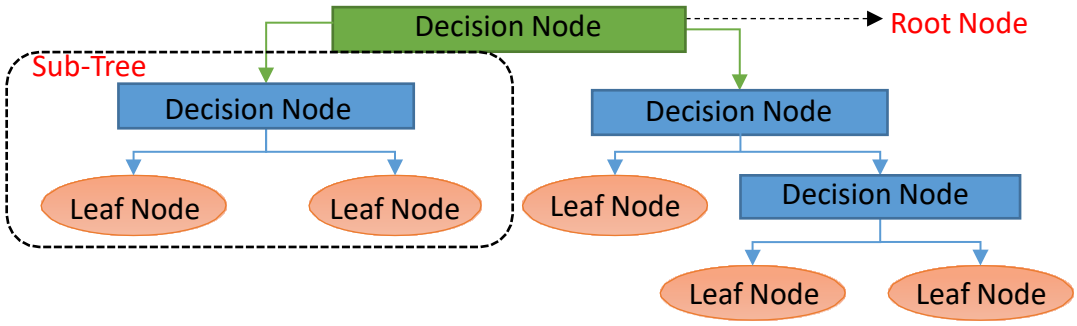


**Figure 5:** DT algorithm

**RFR:** RFR is an ensemble method that builds multiple decision trees and merges their results for more accurate predictions. It is particularly useful in energy consumption studies due to its ability to handle datasets with many variables and its robustness against overfitting. Random Forests can provide insights into variable importance, helping to identify the most significant factors influencing energy use (Tatachar, 2021).

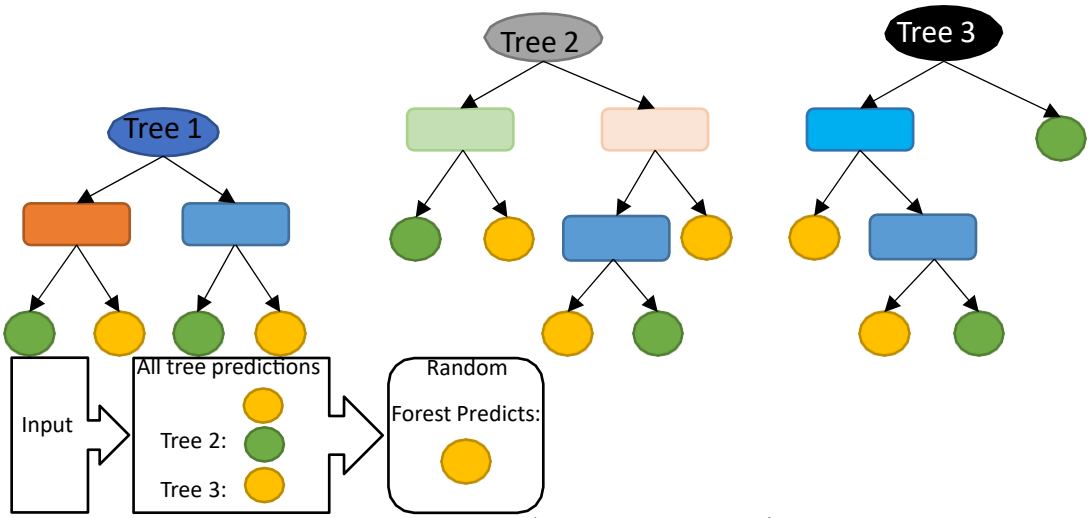


**Figure 6:** Random Forest regression

The Random Forest Regression prediction for a data point x is given by the average of predictions from all the decision trees in the forest:

$$\hat{y} = \frac{1}{N}\sum_{i=1}^{N} T_i(x)$$

where:

- $\hat{y}$ is the predicted value
- N is the number of trees in the Random Forest.
- Ti(x) represents the prediction of the $i^{th}$ decision tree for input x.
- Each decision tree Ti(x) s trained on a random subset of the training data, which helps in reducing variance and preventing overfitting, making Random Forest a robust and powerful regression tool.

**GBR:** GBR is an ensemble technique that sequentially builds models to correct the errors of previous models. This approach is known for its high accuracy and ability to handle various types of data, making it ideal for predicting energy consumption trends. It has been shown to outperform other regression methods due to its iterative approach, which minimizes errors effectively (Otchere et al., 2022; Tatachar, 2021).

$$\hat{y_i} = \sum_{m=1}^{M} \alpha_m h_m(x_i)\,(x_i)$$

where:

- $\widehat{y_i}$ is the predicted value for the i$^{th}$ instance.
- $M$ is the total number of boosting iterations.
- $\alpha_m$ is the weight assigned to the mmm-th weak learner.
- $h_m(x_i)$is the m$^{th}$ weak learner (typically a DT).

**LR:** is a straightforward model used to understand the linear relationship between energy consumption and influencing factors. While it may not capture complex patterns as effectively as non-linear models, it serves as a good baseline for comparison due to its simplicity and interpretability. It is particularly useful for quick, preliminary analyses to identify key trends in energy consumption (Tatachar, 2021).

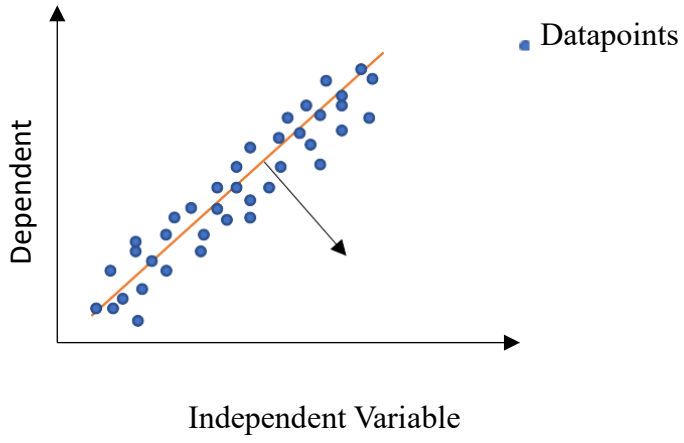


**Figure 7:** Linear regression

The equation for **Simple LR** is:

$$y = a + bx$$

where:

- y is the dependent (response) variable.
- a is the intercept (the value of y when x=0).
- b is the slope (coefficient) of the independent variable x, representing the change in y for a one-unit change in x.
- x is the independent (predictor) variable.

The parameters a and b are estimated using the OLS method, which minimizes the sum of squared residuals (differences between the observed and predicted values of y).

These algorithms—ANN, RFR, GBR, LR, SVM, and DT—are selected based on their effectiveness in handling complex, large-scale datasets and their ability to provide meaningful insights into energy consumption patterns.

### 3.4.2 ML Model Implementation

**Model Training and Validation:** Different machine learning models, including ANN, SVM, DT, RFR, GBR, and LR, were implemented to analyse energy consumption patterns. The models were trained using the prepared dataset and validated using 5-fold cross-validation techniques to ensure generalizability and robustness. The use of 5-fold cross-validation helped mitigate overfitting and provided a reliable estimate of model performance by ensuring that each data point had an equal chance of being included in both the training and validation sets (Oh & Min, 2024).

**Optimization and Hyperparameter Tuning:** Hyperparameter tuning were performed for models such as ANN, SVM, and GBR to optimize their performance. Techniques like grid search were employed to find the best combination of hyperparameters (e.g., learning rates, the number of neurons in hidden layers, max depth of trees) to improve model accuracy and predictive power. The document describes running a series of preliminary tests with training data to select optimal hyperparameters, underscoring the importance of this process in improving the accuracy and efficiency of machine learning models (Oh & Min, 2024).

## 3.5 Model Evaluation

For regression models (RFR, GBR, LR), evaluation metrics such as MSE, RMSE, MAE, $R^2$, and Adjusted R-squared were used to measure the prediction accuracy and error margins of the models and For classification models (ANN, SVM, DT), metrics such as accuracy, precision, recall, F1-score, and confusion matrices were employed to evaluate the models' ability to classify energy consumption patterns correctly (Oh & Min, 2024).

### 3.5.1 Evaluation Metrics in ML Regression and Classification Models

Approximation techniques are measured by metrics defined as following.

1. **RMSE:** RMSE is the square root of the MSE and represents the standard deviation of the residuals (prediction errors). It indicates how spread out these residuals are around the regression line, providing insight into the model's prediction accuracy. Lower RMSE values indicate a better fit to the data (Wanasinghe et al., 2020).

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2]}$$

- $\Sigma$ = summation
- $(y_i - \hat{y}_i)^2$ = squared difference between actual value and prediction
- n = sample size.

**2. $R^2$:** $R^2$ or the coefficient of determination, measures the proportion of variance in the dependent variable that is predictable from the independent variables. An $R^2$ value closer to 1 indicates a better fit of the model (Wanasinghe et al., 2020).

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

where:

- $y_i$ is the actual value of the dependent variable.
- $\hat{y}_i$ is the predicted value from the regression model.
- $\bar{y}$ is the mean of the actual values.

**3. Accuracy:** Accuracy measures the proportion of correctly classified instances out of the total instances. It provides a general measure of how well the model performs across all classes. However, it may not be reliable for imbalanced datasets where one class dominates (Ahmad 2022; Strielkowski et al., 2023).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- **TP**: True Positives - Correctly predicted positive cases
- **TN**: True Negatives - Correctly predicted negative cases
- **FP**: False Positives - Incorrectly predicted positive cases
- **FN**: False Negatives - Incorrectly predicted negative cases

**4. Precision:**

Precision indicates the proportion of correctly predicted positive observations out of all predicted positive observations. It is important in scenarios where the cost of false positives is high, as it focuses on the accuracy of the positive predictions (Ahmad et al., 2022).

$$\text{Precision} = \frac{TP}{TP + FP}$$

**5. Recall (Sensitivity or True Positive Rate):**

Recall (Sensitivity) measures the proportion of actual positives correctly identified by the model. It is crucial in applications where minimizing false negatives is a priority, such as in medical diagnostics or fault detection (Strielkowski et al., 2023).

$$\text{Recall} = \frac{TP}{TP + FN}$$

**6. F1-Score:**

The F1-score is the harmonic mean of precision and recall, providing a balanced evaluation in cases where both false positives and false negatives need to be minimized. It is especially useful for imbalanced datasets (Ahmad et al., 2022).

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

**7. Confusion Matrix:**

A confusion matrix provides a detailed summary of the model's performance by showing the counts of true positives, true negatives, false positives, and false negatives. It helps in understanding the distribution of errors across different classes (Strielkowski et al., 2023).

$$\begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix}$$

### 3.5.2 Model Performance Analysis

The performance of different machine learning models, such as DT, SVM, and ANN for classification tasks, and LR, RFR, and GBR for regression tasks, will be compared. We will use metrics like accuracy, precision, recall, F1-score for classification, and $R^2$, MSE, RMSE, and MAE for regression, aligning well with your need to analyse model performance across different types of tasks (Oh & Min, 2024).

The goal is to find the models that perform the best for both types of tasks. Details about this comparison will be covered in Chapter 4.

### 3.5.3 Validation and Interpretation

The models will be tested with data they haven't seen before to check how well they work in real-world situations. This testing helps ensure that the models can accurately handle new

data, like your goal of testing models for real-world applicability and gaining insights into energy consumption patterns. which will be explained further in Chapter 4.

In the analysis, we will explore these results to understand key patterns in energy consumption and how these can help improve energy management strategies.

# CHAPTER 4 DATA ANALYSIS, FINDINGS, AND RESULTS

This chapter applies the CRISP-DM framework to analyse energy consumption data from smart meter users in SMEs and industries. It follows this structured approach, as outlined in Chapter 3, to guide the development and evaluation of ML models, converting raw data into actionable insights. Data preprocessing includes handling missing values and removing outliers to ensure data quality. Regression models, such as LR, RFR, and GBR, are used to predict future energy consumption, while classification algorithms like SVM, ANN, and DT categorize usage patterns. Model performance is assessed using metrics like accuracy, precision, recall, F1-score, $R^2$, and MSE. Visualization techniques identify key trends and anomalies, providing insights to optimize energy efficiency and reduce costs.

## 4.1 Data Understanding and Preparation

### 4.1.1 Overview of Collected data

The dataset, comprising 396 rows and 46 columns, was loaded using Pandas' read_csv function, which allows for efficient handling of CSV files. The dataset was imported into a Pandas Dataframe for further exploration and processing. This dataset, sourced from smart meter users in SMEs and industries, includes numerical variables that are key to understanding energy consumption patterns.

### 4.1.2 Data Cleansing and Pre-processing

The preprocessing of the "Smart Meters SME" dataset involved several key steps to ensure data quality and readiness for analysis. Irrelevant unnamed columns were removed, and the dataset was renamed for clarity, retaining 'Id' and 'Industry' while other columns were sequentially renamed from 'Q1' to 'Q42'. The dataset had missing values, with 109 instances missing in columns Q1 to Q25 and 67 instances missing in Q26 to Q42. To address this, mean imputation was used to replace missing values with the mean of each feature, preserving the data's overall distribution and reducing bias. Non-informative features, such as 'ID,' were excluded to maintain relevance. These steps ensured the dataset was clean and reliable for further analysis and modelling.

### 4.1.3 Descriptive Analysis

**Table 2: Summary Statistics**

| Features | Count | mean | std | min | max |
|---|---|---|---|---|---|
| Industry | 280.000000 | 2.762842 | 0.892351 | 1.000000 | 4.000000 |
| Q1 | 280.000000 | 1.425075 | 0.522614 | 1.000000 | 3.000000 |
| Q2 | 280.000000 | 1.852115 | 0.790123 | 1.000000 | 5.000000 |
| Q3 | 280.000000 | 2.321964 | 0.944162 | 1.000000 | 5.000000 |
| Q4 | 280.000000 | 2.490468 | 1.057021 | 1.000000 | 5.000000 |
| Q5 | 280.000000 | 1.794525 | 0.783695 | 1.000000 | 5.000000 |
| Q6 | 280.000000 | 2.491426 | 1.080915 | 1.000000 | 5.000000 |

Table 2 provides a summary of the first seven features of the dataset, detailing key metrics such as mean, standard deviation, and range. The `Industry` feature shows a moderate spread, with a mean of 2.76 and a standard deviation of 0.89, indicating a range of different sectors. The survey responses from `Q1` to `Q5` reveal low variability, suggesting a general agreement among respondents on certain aspects of energy management. In contrast, `Q6` and `Q7` show higher variability (standard deviations of 1.08 and above), highlighting different levels of proactive engagement in energy monitoring and management. This range of responses reflects diverse approaches to sustainability and energy efficiency, potentially linked to specific industry types.

### 4.1.4 Split and Randomizing Dataset

After handling outliers, the dataset was reduced to 320 observations across 42 variables. It was then randomly split into training and testing sets using the holdout method to ensure robust model development and evaluation. To avoid any ordering bias, 70% of the data (196 observations) was used for training, while 30% (84 observations) was reserved for testing. This approach ensures that the models are validated on unseen data, enhancing their reliability and generalizability.

## 4.2 Model Development

To improve the predictive performance of the models, two key techniques were employed: hyperparameter tuning and stratified k-fold cross-validation, both for regression and classification tasks.

### 4.2.1 Hyperparameter Tuning

To optimize the performance of the models, hyperparameter tuning was carried out using GridSearchCV for LR, RFR, GBR. This approach tested different combinations of parameters, such as learning rates, the number of estimators for GBR, the depth and number of trees for RFR, and regularization parameters for LR, to find the best settings that reduce prediction errors and enhance accuracy.

This approach systematically tested various combinations of parameters to identify the optimal settings that minimize prediction errors and maximize accuracy.

For the RFR, hyperparameters such as the number of estimators (trees), maximum depth of trees, minimum samples required to split a node, minimum samples required at a leaf node, and the number of features considered for the best split were tested. Similarly, for the GBR, parameters such as the number of boosting stages (estimators), learning rates, maximum depth of trees, minimum samples required to split a node, minimum samples required at a leaf node, and the subsample size were optimized. The goal was to find the best combination of these parameters to improve the models' predictive performance.

**Table 3: Hyperparameter Table for Optimal Parameters for Regression**

| Model | Best Parameters | Best Score |
|---|---|---|
| Linear Regression | N/A | 0.998676 |
| RandomForestRegressor | {'n_estimators': 150, 'max_depth': 20, 'min_samples_split': 2, 'min_samples_leaf': 1, 'max_features': 'sqrt'} | 0.997841 |
| GradientBoostingRegressor | {'n_estimators': 200, 'learning_rate': 0.05, 'max_depth': 4, 'min_samples_split': 2, 'min_samples_leaf': 1, 'subsample': 0.9} | 0.998168 |

Table 3 illustrates that the hyperparameter tuning using GridSearchCV optimized the performance of three regression models: LR, RFR, and GBR. LR, used as a baseline with no hyperparameters tuned, achieved a high R-squared score of 0.998676. For RFR, the best parameters included 150 trees (`n_estimators`), a maximum depth of 20 (`max_depth`), a minimum of 2 samples to split a node (`min_samples_split`), 1 sample per leaf (`min_samples_leaf`), and using the square root of the number of features for the best split (`max_features='sqrt'`), achieving a score of 0.997841. GBR's optimal settings were 200

estimators (`n_estimators`), a learning rate of 0.05 (`learning_rate`), a maximum depth of 4 (`max_depth`), 2 samples to split a node (`min_samples_split`), 1 sample per leaf (`min_samples_leaf`), and a subsample of 90% (`subsample=0.9`), resulting in a score of 0.998168. This tuning process effectively balanced bias and variance, enhancing the overall model accuracy.

### 4.2.2 Cross-Validation Results for SVM

To ensure a balanced and fair evaluation of the various models, including ANN, DT, SVM, RFR, GBR, and LR, stratified k-fold cross-validation was employed. This technique divides the dataset into *k* equal-sized folds while preserving the class proportions in each fold.

For clarity and focus, the results presented here will only highlight the performance of the **SVM** model. The SVM model was chosen for this demonstration due to its balanced performance across different metrics, providing a representative example of the cross-validation approach and its effectiveness.

**Table 4: Cross-Validation Results for Model**

| Models | Fold | Accuracy | F1 Score | Recall | Precision |
|---|---|---|---|---|---|
| SVM | Fold 1 | 87.500000 | 67.246377 | 67.878788 | 66.666667 |
| SVM | Fold 2 | 87.500000 | 67.177033 | 67.878788 | 66.545455 |
| SVM | Fold 3 | 83.928571 | 62.800000 | 65.050505 | 66.000000 |
| SVM | Fold 4 | 89.285714 | 57.777778 | 58.901515 | 57.142857 |
| SVM | Fold 5 | 87.500000 | 55.952381 | 57.407407 | 57.450980 |

Table 4 illustrates the cross-validation results show that the SVM model performed consistently across all folds, with accuracies ranging from 83.93% to 89.29%. The F1 scores and precision values indicate the model's balanced performance between precision and recall. Notably, Fold 4 had the highest accuracy (89.29%), while Fold 5 had the lowest F1 score (55.95%), indicating some variability across different folds.

Figure 8 visually represents these cross-validation results, highlighting the model's performance across the different folds. The graph illustrates that while the SVM model maintains a high accuracy overall, there are differences in F1 scores, recall, and precision, especially in Folds 4 and 5. This variation suggests potential areas for further investigation, such as exploring different hyperparameters or feature selections to improve performance consistency across all metrics.

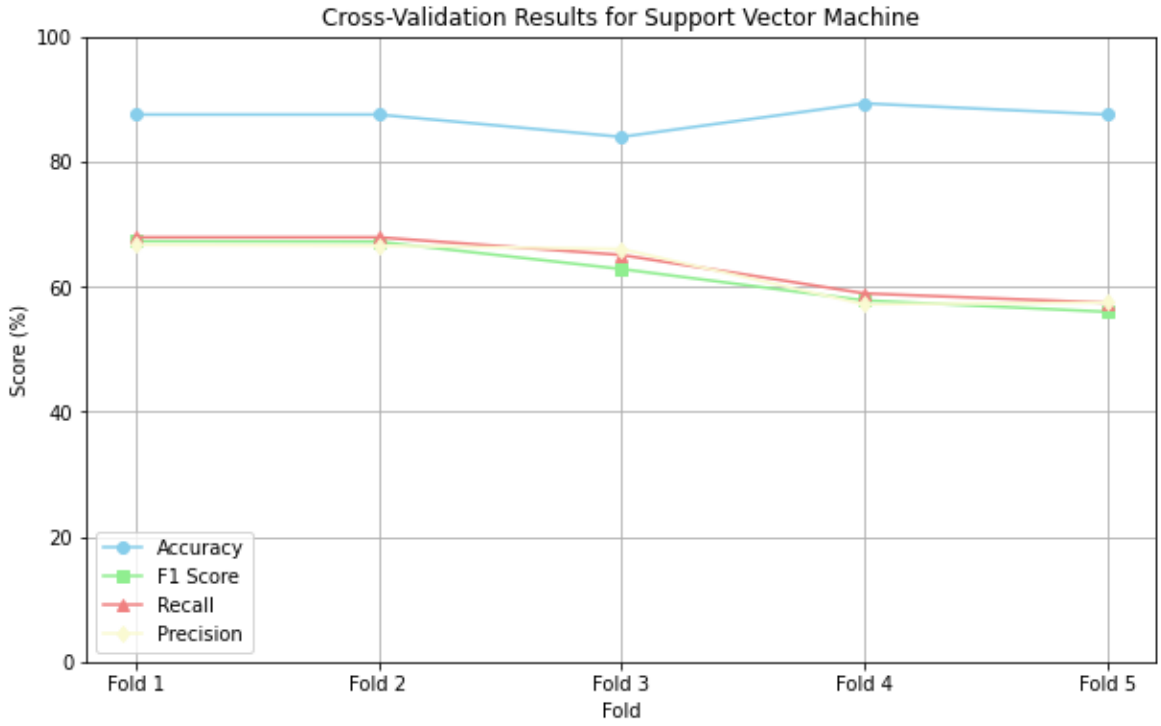


**Figure 8: Cross-Validation Results by Fold for SVM**

### 4.3 Model Evaluation

#### 4.3.1. Data Visualization

To achieve Objective 1, various data visualization techniques were employed to explore and understand the dataset. Histograms identified patterns like skewness and outliers by displaying frequency distributions. Box plots highlighted data spread and central tendencies, aiding in spotting outliers. Scatter plots revealed relationships, correlations, and trends between variables, while a correlation matrix illustrated the strength and direction of these relationships. Identifying outliers was crucial as they could impact the analysis. These visualizations provided a comprehensive view of the data's structure and variability, forming a strong foundation for further analysis.

**1. Histogram:** Figure 9 illustrates the histogram of key variables (Q2 to Q16) and highlights distinct patterns in the data. For variables Q2 to Q7, the responses are mostly left-skewed, with most answers clustering around 1 (Strongly agree), 2 (Agree), and 3 (Neutral), suggesting a consensus among respondents. In contrast, Q8 shows a right-skewed distribution, where most responses are at the lower end (0 to 20%), indicating that many organizations spend around 20% of their non-wage costs on electricity, but also revealing some significant outliers. Similarly, variables Q9 to Q13 follow a left-skewed trend, with Q13 showing a concentration at the minimum score. Meanwhile, Q14 to Q16 display a more

balanced distribution, reflecting a wider range of opinions among respondents. Understanding these variations is crucial for choosing the right statistical methods, identifying any biases, and ensuring a well-rounded interpretation of the data by considering both skewness and outliers.

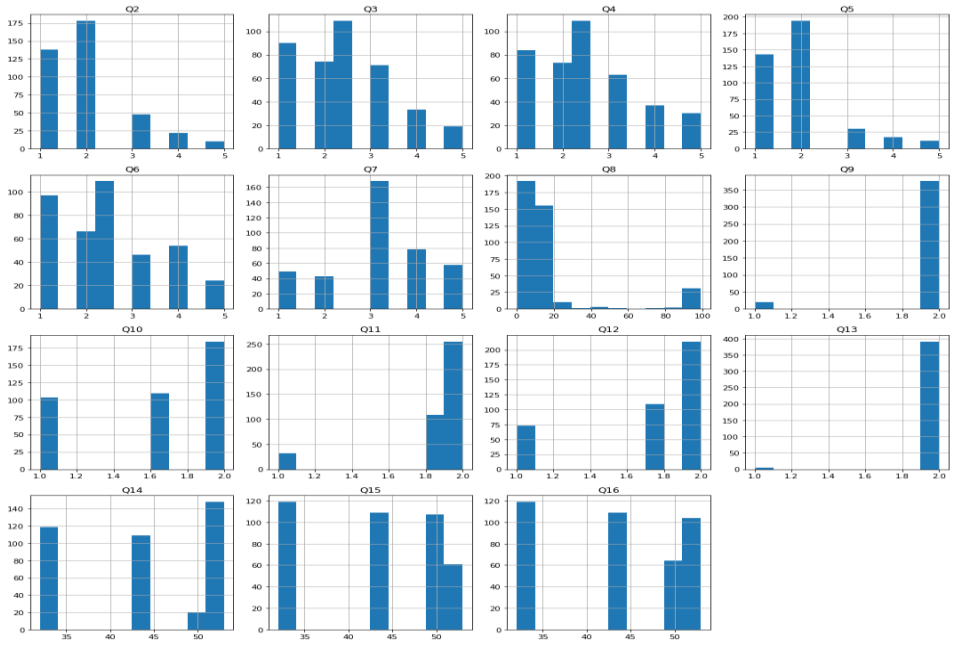


**Figure 9: Histogram Analysis of Key Variables (Q2 to Q16)**

### 4.3.2 Classification

This section will cover *objective 2*. In this imperative task of data mining, classification algorithms: SVM, ANN, DT will be implemented. Before evaluation, a couple of important points needs to be highlighted.

Before moving on to the modelling phase, it is essential to establish a foundation for interpreting the results by understanding key terminologies related to evaluation metrics. These metrics are critical for assessing and interpreting the model's performance. Table 2 provides a detailed explanation of these evaluation metrics.

**Model results and evaluation**

The evaluation of the DT, SVM, and ANN models utilizes key metrics—accuracy, F1 score, recall, precision, ROC curves, and confusion matrices—to assess performance across different class distributions. ROC curves analyze sensitivity-specificity trade-offs, confusion matrices visualize true and false classifications, and bar charts depict metrics like true positives and false positives. While all models display similar overall results, a detailed bin-

wise analysis highlights differences in handling specific classes, aiding in effective model selection.

Table 5 summarizes the results for the DT, SVM, and ANN models, showing varying performance levels across cross-validation (mean across folds) and test sets. The SVM model achieves the highest test set accuracy at 92.86%, indicating strong generalization and stability. The DT model shows a slightly higher cross-validation accuracy (88.93%) but drops to 83.33% on the test set, suggesting overfitting. ANN demonstrates moderate performance with a cross-validation accuracy of 84.29% and a decrease to 80.95% on the test set. F1 scores, recall, and precision follow similar trends, with SVM maintaining consistency across training and testing, while DT and ANN exhibit more variability. Overall, SVM proves to be the most robust model, offering balanced performance across both metrics and datasets.

**Table 5: Classification Models Performance Metrics**

| **Metric** | **DT (Mean Across Folds)** | **DT (Test Set)** | **SVM (Mean Across Folds)** | **SVM (Test Set)** | **ANN (Mean Across Folds)** | **ANN (Test Set)** |
|---|---|---|---|---|---|---|
| Accuracy (%) | 88.93 | 83.333 | 87.14 | 92.857 | 84.29 | 80.952 |
| F1 Score | 0.64 | 0.525 | 0.62 | 0.616 | 0.58 | 0.444 |
| Recall | 0.65 | 0.535 | 0.63 | 0.631 | 0.60 | 0.500 |
| Precision | 0.64 | 0.516 | 0.63 | 0.603 | 0.58 | 0.417 |

**Table 6: Confusion Matrix of Classification Models**

| Actual /Predicted | DT | | | | | | SVM | | | | | | ANN | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 23 | 0 | 0 | 0 | 0 | 0 | 23 | 0 | 0 | 0 | 0 | 0 | 23 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 29 | 0 | 0 | 0 | 0 | 0 | 29 | 0 | 0 | 0 | 0 | 0 | 29 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 |
| 3 | 0 | 0 | 0 | 6 | 7 | 0 | 0 | 0 | 0 | 11 | 2 | 0 | 0 | 0 | 0 | 0 | 13 | 0 |
| 4 | 0 | 0 | 0 | 4 | 12 | 0 | 0 | 0 | 0 | 1 | 15 | 0 | 0 | 0 | 0 | 0 | 16 | 0 |
| 5 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 7: Error Rate of Classification Models**

| Models | DT | | | | SVM | | | | ANN | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Classes** | **TP** | **FP** | **TN** | **FN** | **TP** | **FP** | **TN** | **FN** | **TP** | **FP** | **TN** | **FN** |
| Class 0 | 23 | 0 | 61 | 0 | 23 | 0 | 61 | 0 | 23 | 0 | 60 | 0 |
| Class 1 | 29 | 0 | 55 | 0 | 29 | 0 | 55 | 0 | 29 | 0 | 54 | 0 |
| Class 2 | 0 | 0 | 82 | 2 | 0 | 0 | 82 | 2 | 0 | 0 | 81 | 2 |
| Class 3 | 6 | 7 | 64 | 7 | 11 | 3 | 68 | 2 | 0 | 0 | 70 | 13 |
| Class 4 | 12 | 7 | 61 | 4 | 15 | 3 | 65 | 1 | 16 | 15 | 52 | 0 |
| Class 5 | 0 | 0 | 83 | 1 | 0 | 0 | 83 | 1 | 0 | 0 | 0 | 0 |

Table 7: Error Rate of Classification Models compares the performance of DT, SVM, and ANN across six classes (Class 0 to Class 5) using TP, FP, TN, and FN.

- Classes 0 and 1: All models (DT, SVM, ANN) perform perfectly with no FP or FN, indicating high accuracy.
- Class 2: All models fail to predict TP, leading to only FN.
- Class 3: SVM outperforms DT and ANN with more TP (11) and fewer FN (2). ANN has no TP, indicating poor performance.

- Class 4: SVM shows the best performance with the highest TP (15) and fewer FP (3) compared to DT and ANN.
- Class 5: All models fail to detect True Positives, with DT and SVM having a single FN, while ANN has no TP or TN, suggesting misclassification issues.

Overall, SVM demonstrates the best balance between True Positives and False Positives across most classes, while ANN shows variability and potential overfitting issues in some classes.

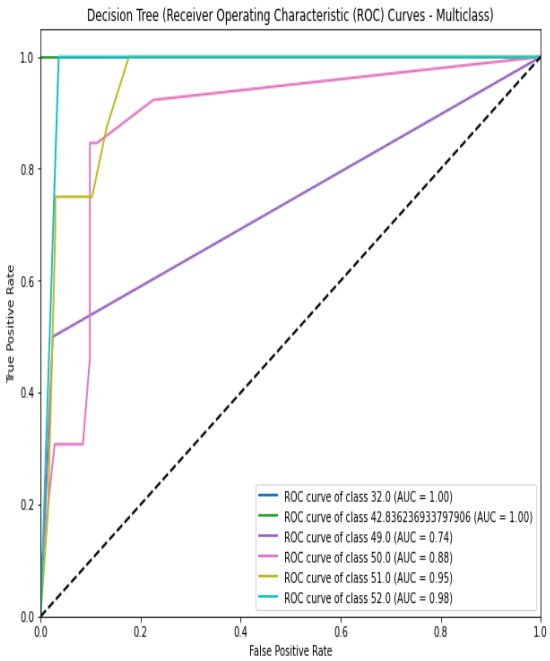


**Figure 10: DT ROC Curve**

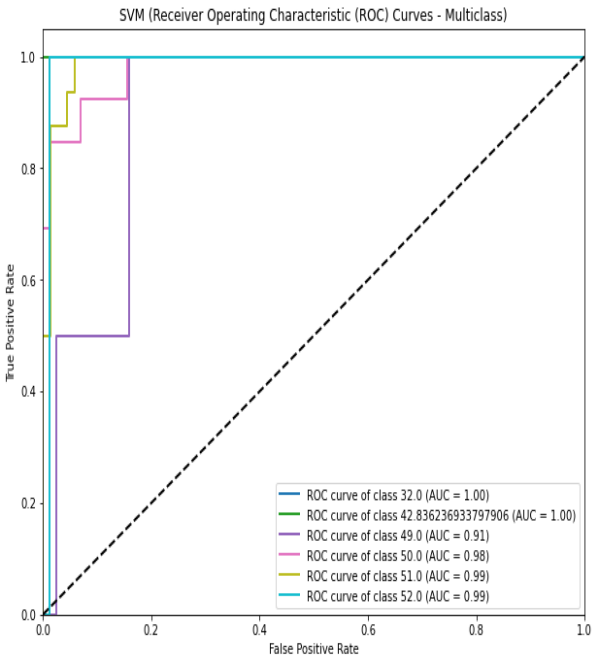


**Figure 11: SVM ROC Curve**

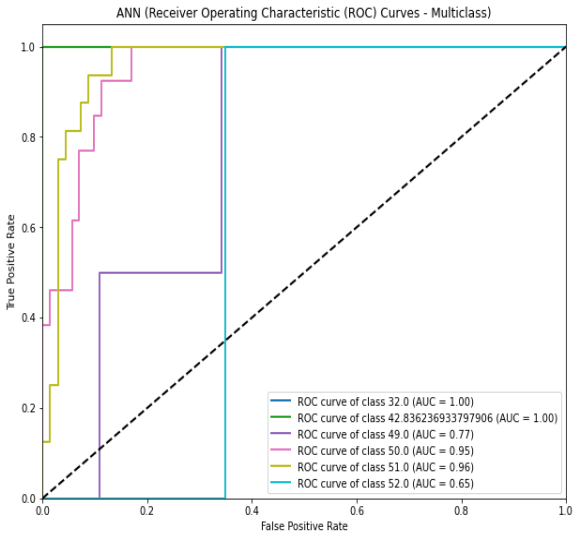


**Figure 12: ANN ROC Curve**

Figures 10, 11, 12 illustrates the ROC curves for the DT, SVM, and ANN models, highlighting their effectiveness in distinguishing between multiple classes in a multiclass dataset. The AUC values quantify each model's ability to differentiate between classes, providing insight into their predictive performance.

The DT model shows varying performance levels across classes. For Classes 32.0 and 42.8, it achieves perfect AUC scores of 1.00, indicating flawless classification. However, for Class 49.0, the AUC drops to 0.74, suggesting moderate differentiation capability. Class 50.0 has an AUC of 0.88, showing improved but imperfect performance. For Class 51.0, the model attains a high AUC of 0.95, while for Class 52.0, it performs strongly with an AUC of 0.98, indicating high accuracy.

The SVM model generally outperforms the others, with consistently high AUC values. It achieves perfect AUC scores of 1.00 for Classes 32.0 and 42.8, indicating excellent classification. For Class 49.0, the AUC is 0.91, reflecting strong classification capability. The AUC values for Classes 50.0, 51.0, and 52.0 are 0.98, 0.99, and 0.99, respectively, showing near-perfect classification and minimal class overlap, demonstrating the model's robustness and reliability.

The ANN model shows a mix of very high and moderate performance levels. It achieves perfect AUC scores of 1.00 for Classes 32.0 and 42.8 but drops to 0.77 for Class 49.0, indicating only fair classification ability. For Class 50.0, it scores an AUC of 0.95, and for Class 51.0, an AUC of 0.96, reflecting strong classification performance. However, the AUC for Class 52.0 is only 0.65, the lowest among all models and classes, suggesting difficulties in distinguishing this class, likely due to model training or data representation issues.

Overall, the SVM model shows the most consistent and superior performance, with the highest AUC values across most classes, making it the most robust choice for this dataset. The DT model performs well but varies in effectiveness, particularly for Classes 49.0 and 50.0. The ANN model generally performs well but struggles significantly with Class 52.0, suggesting the need for further refinement in model training or feature selection. These ROC curves provide a comprehensive assessment of each model's ability to differentiate between classes, guiding the selection of the most suitable model for the dataset.

### 4.3.3 Regression Analysis

*To cover Objective 3,* regression techniques such as LR, RFR, GBR regression will be examined for the prediction of energy consumption, and model results will be discussed and compared. They will be evaluated based on metrics such as $R^2$ and RMSE.

For regression model optimization, hyperparameter tuning is crucial to enhance model performance by finding the best combination of parameters. Here, I used GridSearchCV to perform hyperparameter tuning for the LR, RFR, GBR model. This process involves searching over a defined parameter grid to identify the optimal set of hyperparameters that minimize the prediction error and improve the model's robustness.

**Model results and evaluation**

The performance evaluation of the LR, RFR, and GBR models is based on key regression metrics: MAE, MSE, RMSE, and $R^2$. These metrics provide insights into the model's accuracy and predictive power. The analysis is conducted using both training and testing datasets to ensure robustness and minimize overfitting, identifying strengths and areas for improvement.

**Note:** Dropping the industry column in the code.

**Table 8 – Regression model Comparism evaluation results**

| Model | Dataset | $R^2$ | RMSE | MSE | MAE |
|---|---|---|---|---|---|
| LR | Train | 0.999 | 0.284 | 0.081 | 0.164 |
| LR | Test | 0.998 | 0.328 | 0.107 | 0.196 |
| RFR | Train | 0.999 | 0.177 | 0.031 | 0.086 |
| RFR | Test | 0.998 | 0.372 | 0.139 | 0.151 |
| GBR | Train | 1 | 0.091 | 0.008 | 0.052 |
| GBR | Test | 0.998 | 0.346 | 0.120 | 0.160 |

Table 8 shows that the GBR model demonstrates the highest predictive accuracy and robustness across key performance metrics ($R^2$, RMSE, MSE, and MAE) for both training and testing datasets. For LR, the model shows strong performance with an $R^2$ of 0.999 on the training set and 0.998 on the test set, indicating high explanatory power. However, a slight increase in RMSE from 0.284 (train) to 0.328 (test) and MAE from 0.164 to 0.196 suggests minor overfitting but good generalization overall. The RFR model also exhibits

strength with $R^2$ values of 0.999 (train) and 0.998 (test), and lower training errors (RMSE: 0.177, MAE: 0.086) compared to LR, indicating better training accuracy. However, the test errors (RMSE: 0.372, MAE: 0.151) are slightly higher, suggesting some sensitivity to unseen data.

The GBR model outperforms both LR and RFR, achieving a perfect $R^2$ of 1.000 on the training set and 0.998 on the test set, with the lowest training errors (RMSE: 0.091, MAE: 0.052). Although test set errors (RMSE: 0.346, MAE: 0.160) are slightly higher than in training, GBR maintains strong performance and minimal overfitting. Overall, GBR shows superior predictive capability and model stability, making it the most effective model for this regression task, followed by RFR and LR.

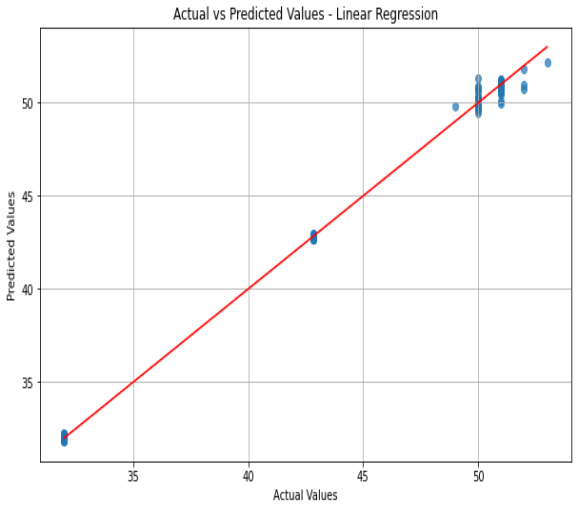


**Figure 13: LR( Test)**

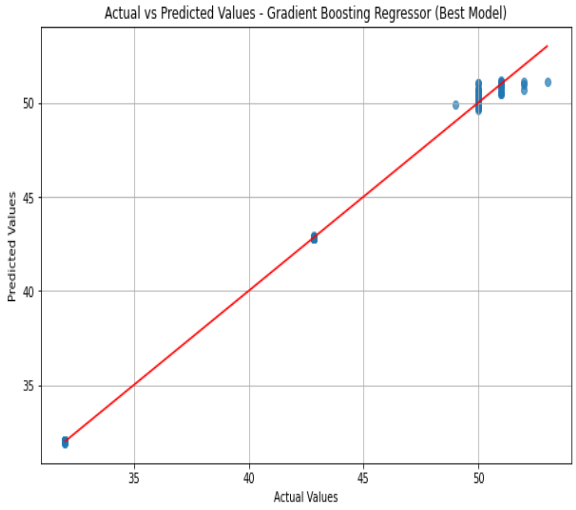


**Figure 14: GBR(Test)**

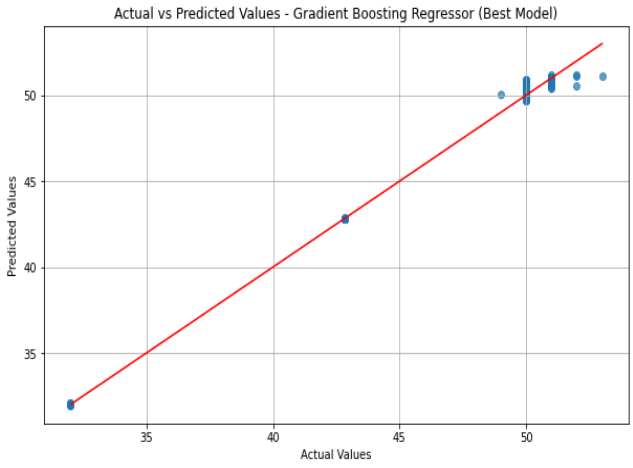


**Figure 15: RFR(Test)**

Figures 13, 14, 15 based on the "Actual vs. Predicted" plots for the test set, show that the GBR model outperforms the LR and RFR models in prediction accuracy. The GBR model demonstrates a tight clustering of predicted values around the identity line, indicating minimal error and excellent generalization capabilities, consistent with its superior performance metrics ($R^2$: 0.998, RMSE: 0.346). The LR model also shows a reasonable fit; however, slight deviations from the identity line suggest it is less effective in capturing non-linear relationships compared to the GBR. The RFR model, while competitive, exhibits a slightly wider spread around the identity line, indicating minor inconsistencies and potential overfitting.

Overall, the GBR model delivers the most reliable and precise predictions, making it the preferred choice for this regression analysis due to its robust performance across different value ranges.

# CHAPTER 5 DISCUSSIONS

This chapter discusses the study's findings on analysing electricity and gas usage data from smart meters in SMEs and industrial settings across the UK to enhance energy management strategies. The research aimed to visualize energy usage, identify consumption patterns, develop models to predict future energy needs, and provide practical recommendations for improving energy efficiency. A mixed-methods approach was used, combining numerical data from smart meters with survey feedback. Advanced ML techniques, including ANN, SVM, DT, LR, RFR, and GBR, were applied for data analysis. This chapter interprets the findings in relation to the literature reviewed in Chapter 2, discusses their implications, and suggests areas for future research.

## 5.1 Overview of Key Findings

From a business perspective, the findings of this study provide valuable insights for SMEs and industrial entities in the UK seeking to optimize their energy management strategies. The use of ML techniques to analyse smart meter data has proven to be highly effective in uncovering energy consumption patterns, enabling businesses to identify opportunities for cost reduction and operational efficiency.

- **Identifying Patterns and Inefficiencies:** he study's analysis reveals that factors such as industry type, energy source, and consumption behaviours show distinct patterns across different user groups. For instance, high-frequency data from smart meters enabled businesses to pinpoint periods of peak consumption and identify inefficiencies, providing a basis for targeted optimization (Oh & Min, 2024). Visualization tools like histograms and scatter plots are essential in understanding when and where energy use peaks, helping implement more effective energy-saving measures (Gholami et al., 2020; Oh & Min, 2024).
- **Effectiveness of Advanced Predictive Models:** The findings underscore the superior performance of advanced predictive models, particularly the GBR model, in forecasting future energy consumption trends. The GBR model outperforms others, such as LR and RFR, due to its iterative approach, which minimizes errors more effectively (Otchere et al., 2022). This level of predictive accuracy enables businesses to plan for future energy needs with greater precision, optimize procurement strategies, and adjust operational schedules to reduce costs and improve efficiency.

- **Classification of Energy Consumption Patterns:** Using ML algorithms such as SVM, DT, and ANN, the study successfully classified energy consumption patterns, providing deeper insights into the factors influencing energy use. This classification helps businesses identify clusters of high and low energy consumers within their operations, facilitating targeted interventions to reduce consumption and costs (Gholami et al., 2020; Chaudhari et al., 2019; Oh & Min, 2024). Moreover, dynamic classification models can be updated with new data, enabling real-time adaptation of strategies to change consumption patterns.

## 5.2 Interpretation of Findings

### 5.2.1 Visualization of Smart Meter Data

For businesses, using visualization techniques such as histograms, scatter plots, and correlation matrices is vital in understanding how energy is consumed. These tools help companies identify patterns, such as periods of peak energy use, and detect unusual consumption behaviours that might go unnoticed otherwise (Vassileva et al., 2013). Insights gained from scatter plots and heatmaps highlight the relationship between energy usage and operational hours, assisting businesses in developing targeted strategies to save energy. Real-time visualization of smart meter data enhances clarity of consumption patterns, encouraging efficient energy use and reducing costs (Gholami et al., 2020; Oh & Min, 2024).

### 5.2.2 Classification of Energy Consumption Patterns

Understanding the drivers of energy use is crucial for businesses, and classifying consumption patterns using ML algorithms offers valuable insights. This study utilized classification models to differentiate between high and low energy consumers among SMEs, which helps craft effective energy management strategies (Oh & Min, 2024). Regular updates to these models with new data allow businesses to adapt flexibly to changes in consumption patterns, reducing costs (Chaudhari et al., 2019; Vassileva et al., 2013). Additionally, ANN models effectively handle complex, non-linear relationships, such as those involving energy consumption, external temperatures, and operational schedules, providing businesses with a clearer picture of their energy use (Gholami et al., 2020; Oh & Min, 2024).

### 5.2.3 Prediction of Energy Consumption

Accurate forecasting of future energy needs is essential for businesses. Predictive modelling techniques such as GBR, RFR, and LR are powerful tools in this regard. The study found

that the GBR model provided the most accurate predictions, effectively capturing complexities and seasonal variations in energy use (Otchere et al., 2022). While RFR models are useful for handling large datasets with many variables, they might overfit, particularly when training data is limited, suggesting caution when applying these models to new data (Oh & Min, 2024).

### 5.2.4 Practical Recommendations

Based on this study's findings, several practical recommendations are proposed to reduce energy consumption, enhance efficiency, and promote sustainable practices in SMEs and industries:

**Implement Real-Time Feedback Systems**: Use real-time feedback combined with smart meter analytics to monitor energy usage closely. Real-time data on dashboards or apps can help businesses make quick decisions to reduce consumption during peak times, encouraging energy-efficient behaviours (Gholami et al., 2020).

**Adopt Dynamic Tariff Plans**: Utilize dynamic tariffs that adjust pricing based on real-time consumption to motivate businesses to shift usage to off-peak times, lowering costs and reducing grid stress. Demand response programs with financial incentives for peak reduction are particularly valuable for SMEs (Chaudhari et al., 2019).

**Leverage Advanced Predictive Models**: Utilize predictive models like GBR and RFR to forecast energy needs, optimize procurement, and schedule operations effectively. Big data platforms, such as Spark and Hive, can enhance these models' scalability and performance (Liu et al., 2016).

**Enhance Data Privacy and Security**: Address privacy concerns by implementing robust anonymization and data protection measures, building trust and encouraging the use of smart meters (Beckel et al., 2014).

**Develop Tailored Energy-Saving Programs**: Create customized programs based on insights from classification models to address specific needs of different SME segments, such as energy audits for high consumers and educational initiatives for low consumers (Oh & Min, 2024).

### 5.3 Comparative Analysis of Results

The analysis in Chapter 4 demonstrates the effectiveness of various ML models in predicting energy consumption and classifying usage patterns in SMEs and industrial settings. The models were assessed using several performance metrics, such as accuracy, precision, recall, F1-score, $R^2$, and RMSE. This section compares the results with existing literature to provide a broader context and interpretation.

#### 5.3.1 Regression Analysis Comparison

The regression models—LR, RFR, and GBR were evaluated for their predictive accuracy. The GBR model outperformed others, achieving the highest $R^2$ value (1.000 on the training set and 0.998 on the test set) and the lowest RMSE (0.091 on training and 0.346 on testing), indicating superior performance in capturing the complexities of energy consumption data.

**Comparison with Literature**

- The findings align with Otchere et al. (2022), who found that the GBR model effectively handles non-linear relationships in datasets due to its iterative approach that reduces errors in each step, leading to higher accuracy and stability compared to traditional models like LR and RFR.
- Similar results were observed in studies by Liu et al. (2019), where the GBR model demonstrated superior performance over other regression models in predicting complex variables due to its robustness against overfitting and ability to handle various feature interactions effectively (Otchere et al., 2022).

#### 5.3.2 Classification Model Comparison

The classification models—SVM, DT, ANN were evaluated to categorize energy consumption patterns. The SVM model achieved the highest test set accuracy (92.86%) and demonstrated consistent performance across folds, indicating its robustness in handling complex datasets.

**Comparison with Literature:**

- The SVM model's superior performance is consistent with findings by Chaudhari et al. (2019), who highlighted its ability to handle high-dimensional spaces and perform well with small to medium-sized datasets. The model's use of a kernel function allows it to classify non-linearly separable data effectively (Vassileva et al., 2013).

- Gholami et al. (2020) also reported that SVMs outperform other classifiers like DT and ANN in contexts where data points are not linearly separable, particularly in energy consumption datasets with multiple influencing factors (Oh & Min, 2024).

### 5.3.3 Interpretation of Model Evaluation Metrics

The metrics used for evaluating both regression and classification models such as $R^2$, RMSE for regression, and accuracy, precision, recall, and F1-score for classification provide insights into the models' effectiveness. For instance, the higher $R^2$ values and lower RMSE of the GBR model suggest that it is particularly well-suited for predicting energy consumption due to its ability to manage non-linear relationships and complex interactions among variables.

**Comparison with Literature:** According to Vassileva et al. (2013), models that effectively balance these metrics are crucial for accurate energy management. In this context, the GBR model's superior performance aligns with the literature's emphasis on minimizing prediction errors to improve decision-making in energy management strategies.

### 5.3.4 Practical Implications and Future Research

The study's findings suggest that SVM and GBR models are optimal for classifying and predicting energy consumption patterns, respectively. These models provide valuable insights for SMEs and industries looking to enhance their energy management strategies by leveraging advanced ML techniques.

**Implications for Practice:**

- The use of SVM for classification and GBR for regression can support businesses in developing targeted energy-saving measures and more accurately forecasting future energy needs. This aligns with recommendations from Oh & Min (2024), who advocate for integrating these models into real-time energy management systems to improve efficiency and sustainability.

**Future Research Directions:**

- Future studies could explore integrating socio-economic data with ML models to provide a more holistic understanding of energy consumption patterns, as suggested by Frederiks et al. (2015); Vassileva et al. (2013).

- Moreover, enhancing model transparency using techniques like SHAP values or LIME could help in interpreting complex ML models, thereby increasing trust and adoption among non-expert stakeholders.

### 5.4 Limitations and Directions for Future Research

While the study presents significant findings, it also has limitations. Firstly, the research relies primarily on quantitative data, limiting its ability to explore qualitative factors like socio-economic and behavioural reasons behind energy use (Stinson, 2015; Vassileva et al., 2013). This gap suggests a need for a more comprehensive understanding of the human factors driving energy consumption. Future research should adopt mixed-method approaches that integrate quantitative data from smart meters with qualitative insights from stakeholders, uncovering socio-economic and cultural influences on energy use (Vassileva et al., 2013).

Additionally, the study's use of historical data may not fully capture future changes in external factors, such as policy shifts or economic fluctuations affecting energy consumption (Beckel et al., 2014; Oh & Min, 2024). Future research should expand datasets to include a broader range of SMEs from different sectors and geographic locations to improve the generalizability of findings. Enhancing model transparency is also essential to increase trust and adoption among business stakeholders (Oh & Min, 2024).

Lastly, integrating behavioural insights with smart meter data could lead to more effective energy-saving strategies, combining technical data with knowledge of human behaviour (Frederiks et al., 2015; Vassileva et al., 2013). Future studies should develop more interpretable models to facilitate understanding and decision-making for non-expert stakeholders.

## CHAPTER 6 CONCLUSIONS

In summary, this study demonstrates the significant potential of advanced ML techniques to enhance energy management for SMEs and industrial settings. By analyzing smart meter data, the research successfully identified key patterns in energy consumption and developed predictive models that can accurately forecast future energy needs. The use of SVM, GBR proved particularly effective, with SVM excelling in classifying energy usage patterns and GBR offering superior predictive accuracy. These findings suggest that integrating these models into real-time energy management systems can lead to more efficient and cost-effective energy usage. However, the study also highlights limitations, such as the focus on quantitative data and the need for more comprehensive insights into socio-economic factors. Future research should address these gaps by incorporating qualitative data and exploring new techniques to enhance model transparency and interpretability. Overall, the research offers valuable insights for businesses aiming to optimize their energy strategies and achieve greater sustainability.

**REFERENCES**

Ahmad, T., Madonski, R., Zhang, D., Huang, C. & Mujeeb, A., 2022. *Data-driven probabilistic machine learning in sustainable smart energy systems: Key developments, challenges, and future research opportunities in the context of the smart grid paradigm*. Renewable and Sustainable Energy Reviews, 160, p. 112128. doi:10.1016/j.rser.2022.112128.

Ajzen, I., 1991. 'The theory of planned behavior', *Organizational Behavior and Human Decision Processes*, 50(2), pp. 179-211.

Beckel, C., Sadamori, L., Staake, T. & Santini, S., 2014. *Revealing household characteristics from smart meter data*. Energy, 78, pp. 397-410.

Buri, Z., Sipos, C., Szűcs, E. & Máté, D., 2024. *Smart and Sustainable Energy Consumption: A Bibliometric Review and Visualization*. Energies, 17, p. 3336.

Chaudhari, K.S., Ukil, A. & Rajasegarar, S., 2019. *Learning-based demand management for intelligent buildings*. IEEE Transactions on Industrial Informatics, 15(2), pp. 795-805.

Chinnathai, D., 2023. *Application of Machine Learning in Smart Meter Data for Energy Efficiency*. Journal of Energy Studies, 15(3), pp. 123-145.

Çınar, Z.M., Nuhu, A.A., Zeeshan, Q., Korhan, O., Asmael, M. & Safaei, B., 2020. *Machine Learning in Predictive Maintenance towards Sustainable Smart Manufacturing in Industry 4.0*. Sustainability, 12(19), p. 8211. doi:10.3390/su12198211.

Emrouznejad, A., Panchmatia, V., Gholami, R., Rigsbee, C. & Kartal, H.B., 2023. *Analysis of Smart Meter Data With Machine Learning for Implications Targeted Towards Residents*. International Journal of Urban Planning and Smart Cities, 4(1).

Flath, C.M., Nicolay, D., Conte, T., van Dinther, C. & Filipova-Neumann, L., 2012. *Cluster Analysis of Smart Metering Data: An Implementation in Practice*. Business & Information Systems Engineering, 4(1), pp. 31-39. doi:10.1007/s12599-011-0201-5.

Frederiks, E.R., Stenner, K. & Hobman, E.V., 2015. *Household energy use: Applying behavioural economics to understand consumer decision-making and behaviour*. Renewable & Sustainable Energy Reviews, 41(4), pp. 1385–1394. doi:10.1016/j.rser.2014.09.026.

Gholami, R., Emrouznejad, A., Alnsour, Y., Kartal, H.B. & Veselova, J., 2020. *The Impact of Smart Meter Installation on Attitude Change towards Energy Consumption Behavior*

*among Northern Ireland Households*. Journal of Global Information Management, 28(4), pp. 21-37.

Gholami, R., Nishant, R. & Emrouznejad, A., 2021. *Modeling Residential Energy Consumption - An Application of IT-Based Solutions and Big Data Analytics for Sustainability*. Journal of Global Information Management, 29(2), pp. 166-193.

James, G., Witten, D., Hastie, T. & Tibshirani, R., 2013. *An Introduction to Statistical Learning with Applications in R*. New York: Springer.

Jovicic, E., Primorac, D., Cupic, M. & Jovic, A., 2023. *Publicly Available Datasets for Predictive Maintenance in the Energy Sector: A Review*. IEEE Access, 11, pp. 73505-73518.

Koumentakos, E., 2022. *Predictive Analytics for Industrial Energy Management Using Machine Learning Techniques*. IEEE Access, 10, pp. 73505-73518.

Liu, X., Golab, L., Golab, W., Ilyas, I.F. & Jin, S., 2016. *Smart meter data analytics: Systems, algorithms, and benchmarking*. ACM Transactions on Database Systems, 42(1).

Mohajeri, M., Ghassemi, A. & Gulliver, T.A., 2020. *Fast Big Data Analytics for Smart Meter Data*. IEEE Open Journal of the Communications Society, November, pp. 1-12. doi:10.1109/OJCOMS.2020.3038590.

Oh, J. & Min, D., 2024. *Prediction of energy consumption for manufacturing small and medium-sized enterprises (SMEs) considering industry characteristics*. Energy, 300, p. 131621. Available at: https://doi.org/10.1016/j.energy.2024.131621.

Otchere, D.A., Ganat, T.O.A., Ojero, J.O., Tackie-Otoo, B.N. & Taki, M.Y., 2022. 'Application of gradient boosting regression model for the evaluation of feature selection techniques in improving reservoir characterisation predictions', *Journal of Petroleum Science and Engineering*, 208, 109244. Available at: https://doi.org/10.1016/j.petrol.2021.109244 (Accessed: 28 August 2024).

Paukstadt, U. & Becker, J., 2021. *Uncovering the business value of the internet of things in the energy domain – a review of smart energy business models*. Electronic Markets, 31, pp. 51-66. doi:10.1007/s12525-019-00381-8.

Sancho-Asensio, A., Navarro, J., Arrieta-Salinas, I., Armendáriz-Íñigo, J.E., Jiménez-Ruano, V., Zaballos, A. & Golobardes, E., 2014. *Improving data partition schemes in Smart*

*Grids via clustering data streams*. Expert Systems with Applications, 41, pp. 5832-5842. doi:10.1016/j.eswa.2014.03.035.

Schröer, C., Kruse, F. & Gómez, J.M., 2021. *A Systematic Literature Review on Applying CRISP-DM Process Model*. Procedia Computer Science, 181, pp. 526–534. doi:10.1016/j.procs.2021.01.199.

Smajla, I., Vulin, D. & Karasalihović Sedlar, D., 2023. *Short-term forecasting of natural gas consumption by determining the statistical distribution of consumption data*. Energy Reports, 10, pp. 2352-2360.

Stinson, J.W., 2015. *Smart energy monitoring technology to reduce domestic electricity and gas consumption through behaviour change*. PhD thesis, Edinburgh Napier University.

Strielkowski, W., Vlasov, A., Selivanov, K., Muraviev, K. & Shakhnov, V., 2023. *Prospects and Challenges of the Machine Learning and Data-Driven Methods for the Predictive Analysis of Power Systems: A Review*. Energies, 16(4025). doi:10.3390/en16104025.

Swan, L.G. & Ugursal, V.I., 2009. *Modeling of end-use energy consumption in the residential sector: A review of modeling techniques*. Renewable and Sustainable Energy Reviews, 13(8), pp. 1819-1835.

Tatachar, A.V., 2021. *Comparative Assessment of Regression Models Based On Model Evaluation Metrics*. International Research Journal of Engineering and Technology (IRJET), 8(9), pp. 853-860.

Vassileva, I., Dahlquist, E., Wallin, F. & Campillo, J., 2013. *Energy consumption feedback devices' impact evaluation on domestic energy use*. Applied Energy, 106, pp. 314-320. doi:10.1016/j.apenergy.2013.01.059.

Wanasinghe, T.R., Gosine, R.G., James, L.A., Mann, G.K.I., de Silva, O. & Warrian, P.J., 2020. *The Internet of Things in the Oil and Gas Industry: A Systematic Review*. IEEE Internet of Things Journal, 7(9), pp. 8654-8669. Available at: https://ieeexplore.ieee.org/document/2995617 [Accessed 31 July 2024].

Watson, R.T., Corbett, J., Boudreau, M.-C. & Webster, J., 2012. *An information strategy for environmental sustainability*. Communications of the ACM, 55(7), pp. 28-30.

## APPENDICES

## APPENDIX A – VISUALISATION

**2. Boxplot:** Figure 16 shows a boxplot that provides a clear picture of how all the numeric variables in the dataset are distributed, highlighting their averages, variations, and any outliers. For variables Q1 to Q7, the data is generally clustered closely together, though there are some noticeable outliers, especially in Q6 and Q7. Q8, on the other hand, has a lot of outliers and a much wider range, indicating a lot of variation in responses. Variables Q9 to Q25 are more balanced, with fewer outliers, suggesting a more consistent pattern in the data. Q35 shows a wide range of responses with several outliers, while the variables from Q26 to Q42 have tighter, more uniform distributions with minimal outliers. This boxplot is important for identifying unusual values and understanding how the data is spread out, which guides the next steps, such as deciding how to handle outliers and which variables to use in the analysis.

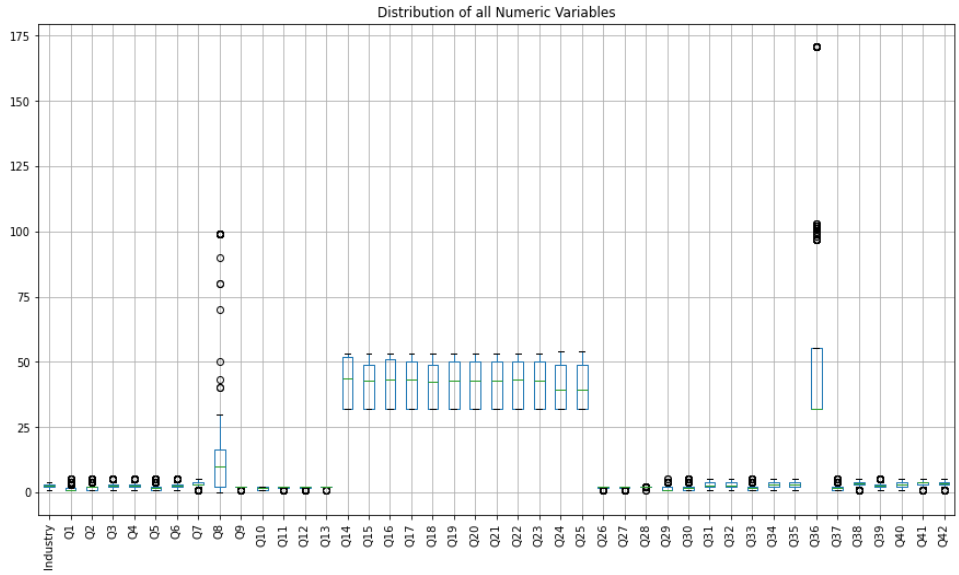


**Figure 16: Box Plot for all Numeric Variables**

**3. Scatter Plot:** Figure 17 shows that the bivariate scatter plot titled **"Bivariate Scatter Plot - Q20 vs Q12"** illustrates the relationship between the variables Q20 and Q12 in the dataset. The plot reveals that most observations are clustered at the extreme values of Q20 (around 35 and 50) and Q12 (at 1.0 and 2.0), suggesting a strong categorical relationship between these variables. The sparse distribution of points in the middle range of Q20 indicates that intermediate values are less common, highlighting specific patterns or groupings in the data. This concentrated clustering implies that certain combinations of Q20 and Q12 are more

prevalent, providing valuable insights into the underlying structure of the dataset and guiding further analysis.

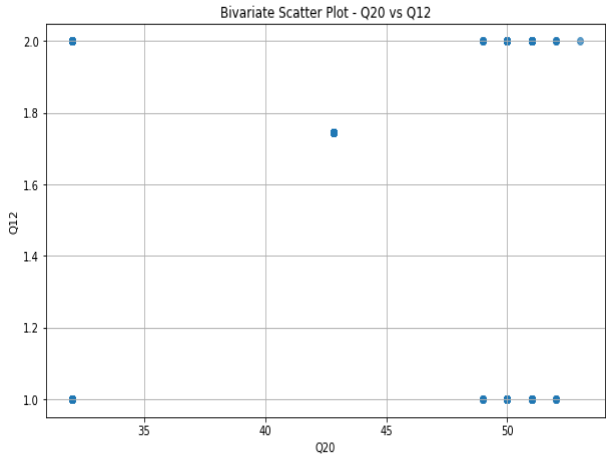


**Figure 17: Bivariate Scatter Plot - Q20 vs Q12**

### 4. Correlation Matrix

Figure 18 illustrates the "Correlation Matrix for All Features" visualizes the relationships between all pairs of variables within the dataset. The matrix uses color-coding to represent the strength and direction of correlations, with values ranging from -1 to 1. Positive correlations are shown in shades of red, indicating a direct relationship where an increase in one variable corresponds to an increase in another. Negative correlations are shown in shades of blue, signifying an inverse relationship where an increase in one variable corresponds to a decrease in another. Values close to 0, depicted in lighter colours, indicate weak or no correlation.

- **Strong Correlations**: The dark red squares along the diagonal indicate strong positive correlations between certain groups of variables, particularly noticeable in clusters like Q16 to Q25 and Q26 to Q33. This suggests that these variables move together, likely capturing similar underlying patterns or behaviours in the data.
- **Weak or No Correlations**: The areas with lighter colours or near-zero values indicate weak or no correlation between many variable pairs, such as between Q1 and most other variables. This suggests that these variables are independent or do not share a linear relationship.

- **Negative Correlations**: There are few instances of negative correlations (indicated by blue shades), showing that some variables have an inverse relationship, though these are less prominent compared to the positive correlations.

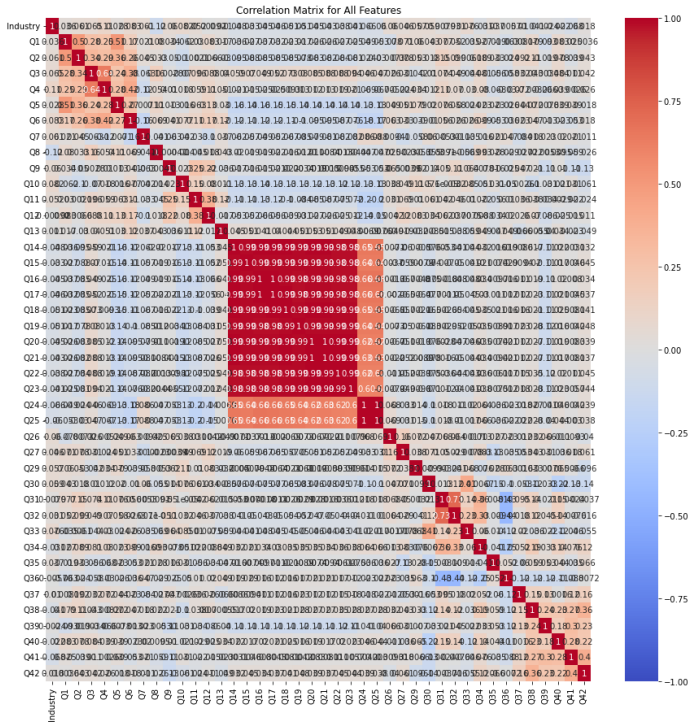


**Figure 18: Correlation Matrix for All Features**

**5. Outliers:** Outliers in a dataset can significantly impact the results of any statistical analysis or modelling process, leading to skewed insights and potentially inaccurate conclusions. To manage this, an outlier analysis was performed using the Interquartile Range (IQR) method on the dataset, with visualizations provided in Figures 19, 20**.**

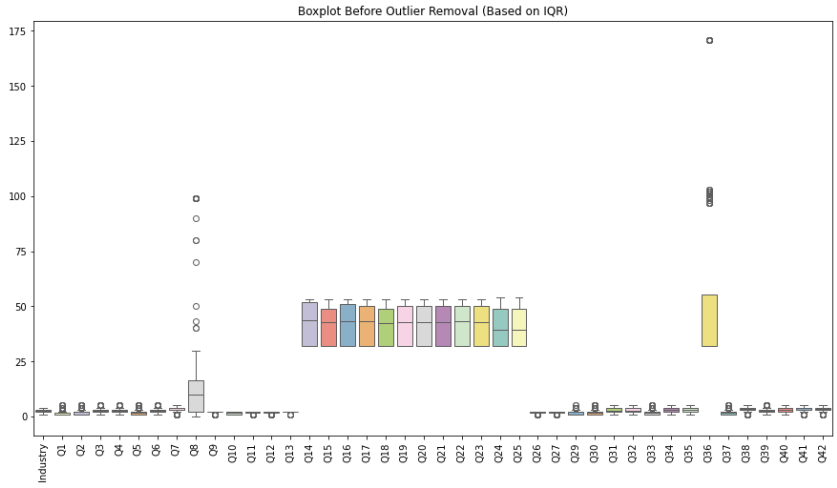


**Figure 19: Boxplot Before Outlier Removal**

The boxplots in Figures 19 illustrate the data distributions before and after outlier removal using the IQR method for variables Q1 to Q42. In the "Before Outlier Removal" boxplot, numerous outliers are evident, especially in variables like Q8, Q35, and Q38, where several points fall far outside the whiskers, indicating extreme values.


**Figure 20: Boxplot After Outlier Removal**

Figure 20 illustrates the impact of applying the IQR method for outlier removal, showing a significant reduction in outliers, particularly for variables like Q8, Q35, and Q38, which previously exhibited several extreme values. The boxplot demonstrates a more compact data distribution with fewer visible outliers, indicating that the IQR-based filtering was effective in minimizing the impact of these extreme values. As a result, the data is now cleaner and more consistent, making it more suitable for accurate analysis and interpretation.

## APPENDIX B – CODE

```
# -*- coding: utf-8 -*-
"""
Created on Tue Aug 27 15:53:21 2024

@author: sv00633
"""

import warnings
warnings.filterwarnings("ignore")

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.impute import SimpleImputer
from sklearn.metrics import accuracy_score, confusion_matrix, f1_score, recall_score, precision_score, mean_absolute_error, r2_score, mean_squared_error
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.feature_selection import RFE
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.neural_network import MLPClassifier
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split, StratifiedKFold, GridSearchCV
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor, RandomForestClassifier
import seaborn as sns
```

```
file_path= r'C:\Users\sv00633\OneDrive - University of Surrey\dissertation\Smart Meters SME.csv'

Raw_DataFrame = pd.read_csv(file_path)

# Drop any unnamed columns

Raw_DataFrame = Raw_DataFrame.loc[:, ~Raw_DataFrame.columns.str.contains('^Unnamed')]


# Rename the columns: Keep 'Id' and 'Industry', rename the rest starting from Q1 to Q42

renamed_columns = ['Id', 'Industry'] + [f"Q{i}" for i in range(1, Raw_DataFrame.shape[1] - 1)]

Raw_DataFrame.columns = renamed_columns


# Convert non-numeric columns to numeric using ASCII sum

def convert_to_numeric(df):

   for column in df.columns:

      if df[column].dtype == 'object':

         df[column] = df[column].apply(lambda x: sum([ord(char) for char in str(x)]) if pd.notna(x) else np.nan)

   return df


Raw_DataFrame = convert_to_numeric(Raw_DataFrame)


# Impute missing values with the mean

imputer = SimpleImputer(strategy='mean')

Raw_DataFrame_imputed = pd.DataFrame(imputer.fit_transform(Raw_DataFrame), columns=Raw_DataFrame.columns)



# Drop the 'Id' column

Raw_DataFrame_imputed = Raw_DataFrame_imputed.drop(columns=['Id'])

print(Raw_DataFrame_imputed.columns)
```

```
# Exploratory Data Analysis (EDA)

# 1. Histograms of the first 15 attributes
plt.figure(figsize=(15, 15))
for i, column in enumerate(Raw_DataFrame_imputed.columns[2:17], 1):  # Skip 'Id' and 'Industry'
    plt.subplot(4, 4, i)
    Raw_DataFrame_imputed[column].hist()
    plt.title(column)
plt.tight_layout()
plt.show()

# 2. Boxplot to visualize the outliers in numeric variables
plt.figure(figsize=(15, 8))
Raw_DataFrame_imputed.boxplot()
plt.title('Distribution of all Numeric Variables')
plt.xticks(rotation='vertical')
plt.show()

# Remove variables with low standard deviation
std_devs = Raw_DataFrame_imputed.std()
low_std_columns = std_devs[std_devs < 0.1].index
Raw_DataFrame_imputed = Raw_DataFrame_imputed.drop(columns=low_std_columns)

# 3. Scatter plot for bivariate analysis
x_col = 'Q20'  # Target variable
y_col = 'Q12'  # Example feature
plt.figure(figsize=(10, 6))
plt.scatter(Raw_DataFrame_imputed[x_col], Raw_DataFrame_imputed[y_col], alpha=0.7)
```

```
plt.xlabel(x_col)

plt.ylabel(y_col)

plt.title(f'Bivariate Scatter Plot - {x_col} vs {y_col}')

plt.grid(True)

plt.show()


# 4. Correlation matrix plot

plt.figure(figsize=(15, 15))

sns.heatmap(Raw_DataFrame_imputed.corr(), annot=True, vmin=-1, vmax=1,
cmap='coolwarm')

plt.title('Correlation Matrix for All Features')

plt.show()


# 5. Outlier removal based on IQR

plt.figure(figsize=(15, 8))

sns.boxplot(data=Raw_DataFrame_imputed, palette="Set3")

plt.title('Boxplot Before Outlier Removal (Based on IQR)')

plt.xticks(rotation=90)

plt.show()


# Outlier removal with a less stringent IQR method

Q1 = Raw_DataFrame_imputed.quantile(0.25)

Q3 = Raw_DataFrame_imputed.quantile(0.75)

IQR = Q3 - Q1


# Use a larger multiplier (e.g., 3) to reduce the number of removed rows

Raw_DataFrame_filtered = Raw_DataFrame_imputed[~((Raw_DataFrame_imputed < (Q1
- 5 * IQR)) | (Raw_DataFrame_imputed > (Q3 + 3 * IQR))).any(axis=1)]


# Boxplot after less stringent outlier removal
```

```
plt.figure(figsize=(15, 8))
sns.boxplot(data=Raw_DataFrame_filtered, palette="Set3")
plt.title('Boxplot After Outlier Removal (Based on IQR)')
plt.xticks(rotation=90)
plt.show()

# 6. Summary Statistics
print('Descriptive Statistics Summary:')
print(Raw_DataFrame_filtered.iloc[:, :7].describe())
print(Raw_DataFrame_filtered.describe())

# Get the total number of observations and variables
total_observations = Raw_DataFrame_filtered.shape[0]
total_variables = Raw_DataFrame_filtered.shape[1]

# Calculate the number of observations for training (70%) and testing (30%) sets
train_observations = int(0.7 * total_observations)
test_observations = total_observations - train_observations

print(f"Total Observations: {total_observations}")
print(f"Total Variables: {total_variables}")
print(f"Training Observations: {train_observations}")
print(f"Testing Observations: {test_observations}")


# Standardizing the data
scaler = StandardScaler()
features_scaled = scaler.fit_transform(Raw_DataFrame_filtered.drop(columns=['Q20']))
```

```
# Example of interpreting standardized data

print("First 5 rows of standardized features:")

print(pd.DataFrame(features_scaled, columns=Raw_DataFrame_filtered.columns[:-
1]).head())


# 7. Decision Tree


# Encode categorical labels as integers

print('DECISION TREE')

label_encoder = LabelEncoder()

target = label_encoder.fit_transform(Raw_DataFrame_filtered['Q20'])

print(target)

# If Q20 is already numerical, directly assign it as the target

# target = Raw_DataFrame_filtered['Q20'].values  # No need for LabelEncoder


# Standardize features

scaler = StandardScaler()

features_scaled = scaler.fit_transform(Raw_DataFrame_filtered.drop(columns=['Q20']))


# Feature selection using RFE with RandomForestClassifier

rf = RandomForestClassifier(random_state=42)

selector = RFE(rf, n_features_to_select=10, step=1)

features_selected = selector.fit_transform(features_scaled, target)


# Initialize the Decision Tree classifier

dt = DecisionTreeClassifier(criterion='gini', max_depth=5, min_samples_leaf=5,
min_samples_split=10, random_state=42)


# Function to perform cross-validation and display results

def evaluate_model_with_cv_dt(model, X, y, cv_folds=5):
```

```
    cv = StratifiedKFold(n_splits=cv_folds, shuffle=True, random_state=42)
    accuracy_scores, f1_scores, recall_scores, precision_scores = [], [], [], []
    for train_index, test_index in cv.split(X, y):
        X_train_fold, X_test_fold = X[train_index], X[test_index]
        y_train_fold, y_test_fold = y[train_index], y[test_index]
        model.fit(X_train_fold, y_train_fold)
        y_pred_fold = model.predict(X_test_fold)
        accuracy_scores.append(accuracy_score(y_test_fold, y_pred_fold))
        f1_scores.append(f1_score(y_test_fold, y_pred_fold, average='macro'))
        recall_scores.append(recall_score(y_test_fold, y_pred_fold, average='macro'))
        precision_scores.append(precision_score(y_test_fold, y_pred_fold, average='macro'))

    print("Overall Performance Across Folds with Decision Tree:")
    print(f"Mean Accuracy: {np.mean(accuracy_scores) * 100:.2f}%")
    print(f"Mean F1 Score: {np.mean(f1_scores):.2f}")
    print(f"Mean Recall: {np.mean(recall_scores):.2f}")
    print(f"Mean Precision: {np.mean(precision_scores):.2f}")

# Evaluate the model using 5-fold cross-validation
evaluate_model_with_cv_dt(dt, features_selected, target, cv_folds=5)

# Train-test split (70-30)
X_train, X_test, y_train, y_test = train_test_split(features_selected, target, test_size=0.3,
random_state=42, stratify=target)

# Train the Decision Tree model on the training set
dt.fit(X_train, y_train)

# Predict and evaluate the model on the test set
y_pred_dt = dt.predict(X_test)
```

```
# Model evaluation metrics
accuracy_dt = accuracy_score(y_test, y_pred_dt)
conf_matrix_dt = confusion_matrix(y_test, y_pred_dt)
f1_dt = f1_score(y_test, y_pred_dt, average='macro')
recall_dt = recall_score(y_test, y_pred_dt, average='macro')
precision_dt = precision_score(y_test, y_pred_dt, average='macro')

print(f"Decision Tree Classifier Accuracy: {accuracy_dt * 100:.3f}%")
print("Confusion Matrix:\n", conf_matrix_dt)
print(f"F1 Score: {f1_dt:.3f}")
print(f"Recall Score: {recall_dt:.3f}")
print(f"Precision Score: {precision_dt:.3f}")

# Plotting the model evaluation metrics as a bar chart
metrics = {
    'accuracy_dt': accuracy_dt,
    'f1_dt': f1_dt,
    'recall_dt': recall_dt,
    'precision_dt': precision_dt
}

metrics_percent = {k: v * 100 for k, v in metrics.items()}

plt.figure(figsize=(10, 6))
plt.bar(metrics_percent.keys(), metrics_percent.values(), color=['skyblue', 'lightgreen',
'lightcoral', 'lightgoldenrodyellow'])
plt.title('Model Evaluation Metrics')
plt.ylabel('Percentage (%)')
plt.ylim(0, 100)
```

```
plt.show()

# Visualization of Confusion Matrix
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix_dt, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix')
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.show()

# Plotting True Positives and False Positives
tp = conf_matrix_dt[1, 1]  # True Positives
fp = conf_matrix_dt[0, 1]  # False Positives
tn = conf_matrix_dt[0, 0]  # True Negatives
fn = conf_matrix_dt[1, 0]  # False Negatives

# Plotting bar chart for True Positives and False Positives
plt.figure(figsize=(8, 6))
sns.barplot(x=['True Positives', 'False Positives', 'True Negatives', 'False Negatives'], y=[tp, fp, tn, fn])
plt.title('True Positives, False Positives, True Negatives, and False Negatives')
plt.show()


# For multiclass, we'll sum over each class to get a binary-like TP, FP, TN, FN count for simplicity
TP = np.diag(conf_matrix_dt)  # True positives are the diagonal values
FP = np.sum(conf_matrix_dt, axis=0) - TP  # False positives are the sum of columns minus TP
FN = np.sum(conf_matrix_dt, axis=1) - TP  # False negatives are the sum of rows minus TP
```

```
TN = np.sum(conf_matrix_dt) - (FP + FN + TP)  # True negatives are the remaining values

# Display the values
print(f"True Positives (TP): {TP}")
print(f"False Positives (FP): {FP}")
print(f"True Negatives (TN): {TN}")
print(f"False Negatives (FN): {FN}")

# Calculate each metric
Accuracy = 100.0 * (TP + TN) / (TP + FP + FN + TN)
TPR = 100.0 * (TP / (TP + FN))  # True Positive Rate (Sensitivity)
FPR = 100.0 * (FP / (FP + TN))  # False Positive Rate
TNR = 100.0 * (TN / (FP + TN))  # True Negative Rate (Specificity)

# Matthews Correlation Coefficient (MCC)
MCC = ((TP * TN) - (FP * FN)) / np.sqrt((TP + FP) * (TP + FN) * (TN + FP) * (TN + FN))

# Handle cases where the denominator in MCC is 0 to avoid division by zero
MCC = np.where(np.isnan(MCC), 0, MCC)  # Replace NaNs with 0

# Print the metrics

# Display the values
print(f"True Positives (TP): {TP}")
print(f"False Positives (FP): {FP}")
print(f"True Negatives (TN): {TN}")
print(f"False Negatives (FN): {FN}")
print(f"Accuracy: {Accuracy}")
print(f"True Positive Rate (TPR): {TPR}")
```

```
print(f"False Positive Rate (FPR): {FPR}")
print(f"True Negative Rate (TNR): {TNR}")
print(f"Matthews Correlation Coefficient (MCC): {MCC}")


# Mapping encoded classes back to their original labels
class_labels = label_encoder.inverse_transform(np.unique(target))
# class_labels = np.unique(target)

# Print what each class constitutes of
for i, label in enumerate(class_labels):
    print(f"Class {i}: {label}")

# The rest of your code follows


labels = [f'Class {i} ({class_labels[i]})' for i in range(len(TP))]
x = np.arange(len(labels))  # the label locations
width = 0.35  # the width of the bars



# # Plotting True Positives and False Positives
# labels = [f'Class {i}' for i in range(len(TP))]
# x = np.arange(len(labels))  # the label locations
# width = 0.35  # the width of the bars
```

```
# Plotting True Positives and False Positives

fig, ax = plt.subplots(figsize=(10, 6))
rects1 = ax.bar(x - width/2, TP, width, label='True Positives')
rects2 = ax.bar(x + width/2, FP, width, label='False Positives')

# Add some text for labels, title and axes ticks
ax.set_xlabel('Classes')
ax.set_ylabel('Count')
ax.set_title('Decison Tree (True Positives and False Positives by Class)')
ax.set_xticks(x)
ax.set_xticklabels(labels, rotation=45)
ax.legend()

fig.tight_layout()
plt.show()

# Plotting True Negatives and False Negatives
fig, ax = plt.subplots(figsize=(10, 6))
rects1 = ax.bar(x - width/2, TN, width, label='True Negatives')
rects2 = ax.bar(x + width/2, FN, width, label='False Negatives')

# Add some text for labels, title and axes ticks
ax.set_xlabel('Classes')
ax.set_ylabel('Count')
ax.set_title('Decison Tree (True Negatives and False Negatives by Class)')
ax.set_xticks(x)
ax.set_xticklabels(labels, rotation=45)
ax.legend()
```

```
fig.tight_layout()
plt.show()

from sklearn.preprocessing import label_binarize
from sklearn.metrics import roc_curve, auc
from sklearn.multiclass import OneVsRestClassifier
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import numpy as np

# Assuming target has the original labels encoded, we need to binarize the labels
n_classes = len(np.unique(target))
y_test_binarized = label_binarize(y_test, classes=np.unique(target))
y_score = dt.predict_proba(X_test)

# Compute ROC curve and ROC area for each class
fpr = dict()
tpr = dict()
roc_auc = dict()

for i in range(n_classes):
    fpr[i], tpr[i], _ = roc_curve(y_test_binarized[:, i], y_score[:, i])
    roc_auc[i] = auc(fpr[i], tpr[i])

# Compute micro-average ROC curve and ROC area
fpr["micro"], tpr["micro"], _ = roc_curve(y_test_binarized.ravel(), y_score.ravel())
roc_auc["micro"] = auc(fpr["micro"], tpr["micro"])
```

```
# Plot all ROC curves
plt.figure(figsize=(10, 8))
colors = plt.cm.get_cmap('tab10', n_classes)

for i, color in zip(range(n_classes), colors.colors):
    plt.plot(fpr[i], tpr[i], color=color, lw=2,
             label=f'ROC curve of class {class_labels[i]} (AUC = {roc_auc[i]:0.2f})')

plt.plot([0, 1], [0, 1], 'k--', lw=2)  # Diagonal line (random classifier)
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Decision Tree (Receiver Operating Characteristic (ROC) Curves - Multiclass)')
plt.legend(loc="lower right")
plt.show()

print('SVM')
# Standardize features
scaler = StandardScaler()
features_scaled = scaler.fit_transform(Raw_DataFrame_filtered.drop(columns=['Q20']))

# Feature selection using RFE with RandomForestClassifier
rf = RandomForestClassifier(random_state=42)
selector = RFE(rf, n_features_to_select=10, step=1)
features_selected = selector.fit_transform(features_scaled, target)



# Initialize the Support Vector Machine (SVM)
```

```
svm = SVC(kernel='linear', random_state=42)

# Function to perform cross-validation and display results in a table format
def evaluate_model_with_cv_svm_table(model, X, y, cv_folds=5):
    cv = StratifiedKFold(n_splits=cv_folds, shuffle=True, random_state=42)
    results = {
        'Fold': [],
        'Accuracy': [],
        'F1 Score': [],
        'Recall': [],
        'Precision': []
    }

    fold_number = 1
    for train_index, test_index in cv.split(X, y):
        X_train_fold, X_test_fold = X[train_index], X[test_index]
        y_train_fold, y_test_fold = y[train_index], y[test_index]
        model.fit(X_train_fold, y_train_fold)
        y_pred_fold = model.predict(X_test_fold)

        # Append results for each fold
        results['Fold'].append(f"Fold {fold_number}")
        results['Accuracy'].append(accuracy_score(y_test_fold, y_pred_fold))
        results['F1 Score'].append(f1_score(y_test_fold, y_pred_fold, average='macro'))
        results['Recall'].append(recall_score(y_test_fold, y_pred_fold, average='macro'))
        results['Precision'].append(precision_score(y_test_fold, y_pred_fold,
average='macro'))
        fold_number += 1

    # Convert results to DataFrame
```

```
    results_df = pd.DataFrame(results)
    results_df['Accuracy'] = results_df['Accuracy'] * 100  # Convert to percentage
    results_df['F1 Score'] = results_df['F1 Score'] * 100  # Convert to percentage
    results_df['Recall'] = results_df['Recall'] * 100  # Convert to percentage
    results_df['Precision'] = results_df['Precision'] * 100  # Convert to percentage

    # Print the results as a table
    print("Cross-Validation Results by Fold:")
    print(results_df.to_string(index=False))

# Evaluate the model using 5-fold cross-validation and display results in a table
evaluate_model_with_cv_svm_table(svm, features_selected, target, cv_folds=5)

# Train-test split (70-30)
X_train, X_test, y_train, y_test = train_test_split(features_selected, target, test_size=0.3,
random_state=42, stratify=target)

# Train the SVM model on the training set
svm.fit(X_train, y_train)

# Predict and evaluate the model on the test set
y_pred_svm = svm.predict(X_test)

# Model evaluation metrics
accuracy_svm = accuracy_score(y_test, y_pred_svm)
conf_matrix_svm = confusion_matrix(y_test, y_pred_svm)
f1_svm = f1_score(y_test, y_pred_svm, average='macro')
recall_svm = recall_score(y_test, y_pred_svm, average='macro')
precision_svm = precision_score(y_test, y_pred_svm, average='macro')
```

```
print(f"Support Vector Machine Accuracy: {accuracy_svm * 100:.3f}%")

print("Confusion Matrix:\n", conf_matrix_svm)

print(f"F1 Score: {f1_svm:.3f}")

print(f"Recall Score: {recall_svm:.3f}")

print(f"Precision Score: {precision_svm:.3f}")


# Plotting the model evaluation metrics as a bar chart

metrics = {

   'accuracy_svm': accuracy_svm,

   'f1_svm': f1_svm,

   'recall_svm': recall_svm,

   'precision_svm': precision_svm

}


metrics_percent = {k: v * 100 for k, v in metrics.items()}


plt.figure(figsize=(10, 6))

plt.bar(metrics_percent.keys(), metrics_percent.values(), color=['skyblue', 'lightgreen',
'lightcoral', 'lightgoldenrodyellow'])

plt.title('Model Evaluation Metrics')

plt.ylabel('Percentage (%)')

plt.ylim(0, 100)

plt.show()


# Visualization of Confusion Matrix

plt.figure(figsize=(8, 6))

sns.heatmap(conf_matrix_svm, annot=True, fmt='d', cmap='Blues')

plt.title('Confusion Matrix')

plt.xlabel('Predicted Labels')

plt.ylabel('True Labels')
```

```
plt.show()


# Plotting True Positives and False Positives
# Get the list of unique classes that were present during training
present_classes = np.unique(y_train)

# Get the original labels for the classes that were seen during training
class_labels_present = label_encoder.inverse_transform(present_classes)

# Create a mapping from the class index to the label
class_labels = {i: label for i, label in zip(present_classes, class_labels_present)}

# Ensure the TP, FP, TN, and FN arrays match the number of present classes
TP = np.diag(conf_matrix_svm)  # True positives for present classes
FP = np.sum(conf_matrix_svm, axis=0) - TP  # False positives
FN = np.sum(conf_matrix_svm, axis=1) - TP  # False negatives
TN = np.sum(conf_matrix_svm) - (FP + FN + TP)  # True negatives

# Calculate each metric
Accuracy = 100.0 * (TP + TN) / (TP + FP + FN + TN)
TPR = 100.0 * (TP / (TP + FN))  # True Positive Rate (Sensitivity)
FPR = 100.0 * (FP / (FP + TN))  # False Positive Rate
TNR = 100.0 * (TN / (FP + TN))  # True Negative Rate (Specificity)

# Matthews Correlation Coefficient (MCC)
MCC = ((TP * TN) - (FP * FN)) / np.sqrt((TP + FP) * (TP + FN) * (TN + FP) * (TN + FN))

# Handle cases where the denominator in MCC is 0 to avoid division by zero
```

```
MCC = np.where(np.isnan(MCC), 0, MCC)  # Replace NaNs with 0


# Print the metrics


# Display the values
print(f"True Positives (TP): {TP}")
print(f"False Positives (FP): {FP}")
print(f"True Negatives (TN): {TN}")
print(f"False Negatives (FN): {FN}")
print(f"Accuracy: {Accuracy}")
print(f"True Positive Rate (TPR): {TPR}")
print(f"False Positive Rate (FPR): {FPR}")
print(f"True Negative Rate (TNR): {TNR}")
print(f"Matthews Correlation Coefficient (MCC): {MCC}")



# Plotting bar chart for True Positives and False Positives
labels = [f'Class {i} ({class_labels[i]})' for i in present_classes]  # Only label present classes
x = np.arange(len(present_classes))  # Adjust x locations for present classes
width = 0.35  # Width of the bars

fig, ax = plt.subplots(figsize=(10, 6))
rects1 = ax.bar(x - width/2, TP, width, label='True Positives')
rects2 = ax.bar(x + width/2, FP, width, label='False Positives')

ax.set_xlabel('Classes')
ax.set_ylabel('Count')
ax.set_title('True Positives and False Positives by Class')
ax.set_xticks(x)
```

```
ax.set_xticklabels(labels, rotation=45)
ax.legend()

fig.tight_layout()
plt.show()

# Plotting True Negatives and False Negatives
fig, ax = plt.subplots(figsize=(10, 6))
rects1 = ax.bar(x - width/2, TN, width, label='True Negatives')
rects2 = ax.bar(x + width/2, FN, width, label='False Negatives')

ax.set_xlabel('Classes')
ax.set_ylabel('Count')
ax.set_title('True Negatives and False Negatives by Class')
ax.set_xticks(x)
ax.set_xticklabels(labels, rotation=45)
ax.legend()

fig.tight_layout()
plt.show()



from sklearn.preprocessing import label_binarize
from sklearn.metrics import roc_curve, auc

# Binarize the labels for ROC curve computation
# Ensure the SVM was trained with probability=True
svm = SVC(kernel='linear', probability=True, random_state=42)
```

```
# Train the SVM model on the training set
svm.fit(X_train, y_train)

# Predict probabilities on the test set
y_score = svm.predict_proba(X_test)

# Binarize the labels for ROC curve computation
y_test_binarized = label_binarize(y_test, classes=present_classes)

# Compute ROC curve and ROC area for each class
fpr = dict()
tpr = dict()
roc_auc = dict()

for i, class_idx in enumerate(present_classes):
    fpr[class_idx], tpr[class_idx], _ = roc_curve(y_test_binarized[:, i], y_score[:, i])
    roc_auc[class_idx] = auc(fpr[class_idx], tpr[class_idx])

# Compute micro-average ROC curve and ROC area (if there are multiple classes)
if len(present_classes) > 1:
    fpr["micro"], tpr["micro"], _ = roc_curve(y_test_binarized.ravel(), y_score.ravel())
    roc_auc["micro"] = auc(fpr["micro"], tpr["micro"])

# Plot all ROC curves
plt.figure(figsize=(10, 8))
colors = plt.cm.get_cmap('tab10', len(present_classes))

for i, color in zip(present_classes, colors.colors):
```

```
    plt.plot(fpr[i], tpr[i], color=color, lw=2,
             label=f'ROC curve of class {class_labels[i]} (AUC = {roc_auc[i]:0.2f})')

plt.plot([0, 1], [0, 1], 'k--', lw=2)  # Diagonal line (random classifier)
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('SVM (Receiver Operating Characteristic (ROC) Curves - Multiclass)')
plt.legend(loc="lower right")
plt.show()



print('ANN')
# Encode categorical labels as integers
label_encoder = LabelEncoder()
target = label_encoder.fit_transform(Raw_DataFrame_filtered['Q20'])

# Standardize features
scaler = StandardScaler()
features_scaled = scaler.fit_transform(Raw_DataFrame_filtered.drop(columns=['Q20']))

# Feature selection using RFE with RandomForestClassifier
rf = RandomForestClassifier(random_state=42)
selector = RFE(rf, n_features_to_select=10, step=1)
features_selected = selector.fit_transform(features_scaled, target)



# Initialize the Artificial Neural Network (ANN) with reduced complexity
```

```
ann = MLPClassifier(hidden_layer_sizes=(10, 10), max_iter=1000, random_state=42)

# Function to perform cross-validation and display results
def evaluate_model_with_cv_ann(model, X, y, cv_folds=5):
    cv = StratifiedKFold(n_splits=cv_folds, shuffle=True, random_state=42)
    accuracy_scores, f1_scores, recall_scores, precision_scores = [], [], [], []
    for train_index, test_index in cv.split(X, y):
        X_train_fold, X_test_fold = X[train_index], X[test_index]
        y_train_fold, y_test_fold = y[train_index], y[test_index]
        model.fit(X_train_fold, y_train_fold)
        y_pred_fold = model.predict(X_test_fold)
        accuracy_scores.append(accuracy_score(y_test_fold, y_pred_fold))
        f1_scores.append(f1_score(y_test_fold, y_pred_fold, average='macro'))
        recall_scores.append(recall_score(y_test_fold, y_pred_fold, average='macro'))
        precision_scores.append(precision_score(y_test_fold, y_pred_fold, average='macro'))

    print("Overall Performance Across Folds with ANN:")
    print(f"Mean Accuracy: {np.mean(accuracy_scores) * 100:.2f}%")
    print(f"Mean F1 Score: {np.mean(f1_scores):.2f}")
    print(f"Mean Recall: {np.mean(recall_scores):.2f}")
    print(f"Mean Precision: {np.mean(precision_scores):.2f}")

# Evaluate the model using 5-fold cross-validation
evaluate_model_with_cv_ann(ann, features_selected, target, cv_folds=5)

# Train-test split (70-30)
X_train, X_test, y_train, y_test = train_test_split(features_selected, target, test_size=0.3,
random_state=42, stratify=target)

# Train the ANN model on the training set
```

```
ann.fit(X_train, y_train)

# Predict and evaluate the model on the test set
y_pred_ann = ann.predict(X_test)

# Get the list of classes in the original label encoding
class_labels = label_encoder.classes_

# Compute the confusion matrix
conf_matrix_ann = confusion_matrix(y_test, y_pred_ann, labels=range(5))

# Ensure it's a 5x5 confusion matrix
if conf_matrix_ann.shape != (5, 5):
    full_conf_matrix = np.zeros((5, 5), dtype=int)
    min_dim = min(conf_matrix_ann.shape[0], conf_matrix_ann.shape[1])
    full_conf_matrix[:min_dim, :min_dim] = conf_matrix_ann[:min_dim, :min_dim]
    conf_matrix_ann = full_conf_matrix

print("Confusion Matrix (5x5):\n", conf_matrix_ann)

# Model evaluation metrics
accuracy_ann = accuracy_score(y_test, y_pred_ann)
f1_ann = f1_score(y_test, y_pred_ann, average='macro')
recall_ann = recall_score(y_test, y_pred_ann, average='macro')
precision_ann = precision_score(y_test, y_pred_ann, average='macro')

print(f"Artificial Neural Network Accuracy: {accuracy_ann * 100:.3f}%")
print(f"F1 Score: {f1_ann:.3f}")
print(f"Recall Score: {recall_ann:.3f}")
```

```
print(f"Precision Score: {precision_ann:.3f}")



# Plotting the model evaluation metrics as a bar chart
metrics = {
    'accuracy_ann': accuracy_ann,
    'f1_ann': f1_ann,
    'recall_ann': recall_ann,
    'precision_ann': precision_ann
}

metrics_percent = {k: v * 100 for k, v in metrics.items()}

plt.figure(figsize=(10, 6))
plt.bar(metrics_percent.keys(), metrics_percent.values(), color=['skyblue', 'lightgreen',
'lightcoral', 'lightgoldenrodyellow'])
plt.title('Model Evaluation Metrics')
plt.ylabel('Percentage (%)')
plt.ylim(0, 100)
plt.show()

# Visualization of Confusion Matrix
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix_ann, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix')
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.show()
# Plotting True Positives and False Positives
TP = np.diag(conf_matrix_ann)  # True positives are the diagonal values
```

```
FP = np.sum(conf_matrix_ann, axis=0) - TP  # False positives are the sum of columns minus TP

FN = np.sum(conf_matrix_ann, axis=1) - TP  # False negatives are the sum of rows minus TP

TN = np.sum(conf_matrix_ann) - (FP + FN + TP)  # True negatives are the remaining values

# Calculate each metric

Accuracy = 100.0 * (TP + TN) / (TP + FP + FN + TN)

TPR = 100.0 * (TP / (TP + FN))  # True Positive Rate (Sensitivity)

FPR = 100.0 * (FP / (FP + TN))  # False Positive Rate

TNR = 100.0 * (TN / (FP + TN))  # True Negative Rate (Specificity)


# Matthews Correlation Coefficient (MCC)

MCC = ((TP * TN) - (FP * FN)) / np.sqrt((TP + FP) * (TP + FN) * (TN + FP) * (TN + FN))


# Handle cases where the denominator in MCC is 0 to avoid division by zero

MCC = np.where(np.isnan(MCC), 0, MCC)  # Replace NaNs with 0


# Print the metrics


# Display the values

print(f"True Positives (TP): {TP}")

print(f"False Positives (FP): {FP}")

print(f"True Negatives (TN): {TN}")

print(f"False Negatives (FN): {FN}")

print(f"Accuracy: {Accuracy}")

print(f"True Positive Rate (TPR): {TPR}")

print(f"False Positive Rate (FPR): {FPR}")

print(f"True Negative Rate (TNR): {TNR}")

print(f"Matthews Correlation Coefficient (MCC): {MCC}")
```

```
# Plotting bar chart for True Positives and False Positives
labels = [f'Class {i}' for i in range(5)]  # Ensure labels cover all expected classes
x = np.arange(len(labels))  # the label locations
width = 0.35  # the width of the bars

fig, ax = plt.subplots(figsize=(10, 6))
rects1 = ax.bar(x - width/2, TP, width, label='True Positives')
rects2 = ax.bar(x + width/2, FP, width, label='False Positives')

ax.set_xlabel('Classes')
ax.set_ylabel('Count')
ax.set_title('True Positives and False Positives by Class')
ax.set_xticks(x)
ax.set_xticklabels(labels, rotation=45)
ax.legend()

fig.tight_layout()
plt.show()

# Plotting True Negatives and False Negatives
fig, ax = plt.subplots(figsize=(10, 6))
rects1 = ax.bar(x - width/2, TN, width, label='True Negatives')
rects2 = ax.bar(x + width/2, FN, width, label='False Negatives')

ax.set_xlabel('Classes')
ax.set_ylabel('Count')
ax.set_title('True Negatives and False Negatives by Class')
ax.set_xticks(x)
ax.set_xticklabels(labels, rotation=45)
```

```
ax.legend()

fig.tight_layout()
plt.show()

# ROC Curve for the ANN
from sklearn.preprocessing import label_binarize
from sklearn.metrics import roc_curve, auc
import matplotlib.pyplot as plt
import numpy as np

# Binarize the labels for ROC curve computation
# We binarize only for the classes that are actually present in the y_test
present_classes = np.unique(y_test)
y_test_binarized = label_binarize(y_test, classes=present_classes)
y_score = ann.predict_proba(X_test)

# Compute ROC curve and ROC area for each class
fpr = dict()
tpr = dict()
roc_auc = dict()

for i, class_idx in enumerate(present_classes):
    fpr[class_idx], tpr[class_idx], _ = roc_curve(y_test_binarized[:, i], y_score[:, i])
    roc_auc[class_idx] = auc(fpr[class_idx], tpr[class_idx])

# Compute micro-average ROC curve and ROC area (if there are multiple classes)
if len(present_classes) > 1:
    fpr["micro"], tpr["micro"], _ = roc_curve(y_test_binarized.ravel(), y_score.ravel())
```

```
    roc_auc["micro"] = auc(fpr["micro"], tpr["micro"])

# Plot all ROC curves
plt.figure(figsize=(10, 8))
colors = plt.cm.get_cmap('tab10', len(present_classes))

for i, color in zip(present_classes, colors.colors):
    plt.plot(fpr[i], tpr[i], color=color, lw=2,
             label=f'ROC curve of class {class_labels[i]} (AUC = {roc_auc[i]:0.2f})')

plt.plot([0, 1], [0, 1], 'k--', lw=2)  # Diagonal line (random classifier)
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ANN (Receiver Operating Characteristic (ROC) Curves - Multiclass)')
plt.legend(loc="lower right")
plt.show()



# Initialize models
dt = DecisionTreeClassifier(criterion='gini', max_depth=5, min_samples_leaf=5,
min_samples_split=10, random_state=42)
ann = MLPClassifier(hidden_layer_sizes=(10, 10), max_iter=1000, random_state=42)
svm = SVC(kernel='linear', probability=True, random_state=42)

import matplotlib.pyplot as plt
import pandas as pd
from sklearn.metrics import accuracy_score, f1_score, recall_score, precision_score
from sklearn.model_selection import StratifiedKFold
```

```
# Updated Cross-validation function with plotting
def evaluate_model_with_cv_table(model, X, y, model_name, cv_folds=5):
    cv = StratifiedKFold(n_splits=cv_folds, shuffle=True, random_state=42)
    results = {
        'Model': [],
        'Fold': [],
        'Accuracy': [],
        'F1 Score': [],
        'Recall': [],
        'Precision': []
    }

    fold_number = 1
    for train_index, test_index in cv.split(X, y):
        X_train_fold, X_test_fold = X[train_index], X[test_index]
        y_train_fold, y_test_fold = y[train_index], y[test_index]
        model.fit(X_train_fold, y_train_fold)
        y_pred_fold = model.predict(X_test_fold)

        # Append results for each fold
        results['Model'].append(model_name)
        results['Fold'].append(f"Fold {fold_number}")
        results['Accuracy'].append(accuracy_score(y_test_fold, y_pred_fold) * 100)
        results['F1 Score'].append(f1_score(y_test_fold, y_pred_fold, average='macro') * 100)
        results['Recall'].append(recall_score(y_test_fold, y_pred_fold, average='macro') *
100)
        results['Precision'].append(precision_score(y_test_fold, y_pred_fold,
average='macro') * 100)
        fold_number += 1
```

```
    # Convert results to DataFrame
    results_df = pd.DataFrame(results)
    print(f"Cross-Validation Results by Fold for {model_name}:")
    print(results_df.to_string(index=False))

    # Plotting cross-validation results
    plt.figure(figsize=(10, 6))
    plt.plot(results_df['Fold'], results_df['Accuracy'], marker='o', label='Accuracy',
color='skyblue')
    plt.plot(results_df['Fold'], results_df['F1 Score'], marker='s', label='F1 Score',
color='lightgreen')
    plt.plot(results_df['Fold'], results_df['Recall'], marker='^', label='Recall',
color='lightcoral')
    plt.plot(results_df['Fold'], results_df['Precision'], marker='d', label='Precision',
color='lightgoldenrodyellow')

    plt.title(f'Cross-Validation Results for {model_name}')
    plt.xlabel('Fold')
    plt.ylabel('Score (%)')
    plt.ylim(0, 100)
    plt.legend(loc='best')
    plt.grid(True)
    plt.show()

# Evaluate models
evaluate_model_with_cv_table(dt, features_selected, target, "Decision Tree")
evaluate_model_with_cv_table(ann, features_selected, target, "Artificial Neural
Network")
evaluate_model_with_cv_table(svm, features_selected, target, "Support Vector
Machine")
```

```
#Regression Models

print('LR')

target = Raw_DataFrame_imputed['Q20']

features = Raw_DataFrame_imputed.drop(columns=['Q20','Industry'])


# Standardize features

scaler = StandardScaler()

features_scaled = scaler.fit_transform(features)


# Split the data into training and testing sets

X_train, X_test, y_train, y_test = train_test_split(features_scaled, target, test_size=0.3,
random_state=42)


# Initialize the Linear Regression model

lr = LinearRegression()


# Fit the model on the training data

lr.fit(X_train, y_train)


# Predict on the training set

y_train_pred_lr = lr.predict(X_train)


# Predict on the testing set

y_test_pred_lr = lr.predict(X_test)


# Evaluate the model on the training set

mae_lr_train = mean_absolute_error(y_train, y_train_pred_lr)

mse_lr_train = mean_squared_error(y_train, y_train_pred_lr)
```

```
rmse_lr_train = np.sqrt(mse_lr_train)
r2_lr_train = r2_score(y_train, y_train_pred_lr)

# Evaluate the model on the testing set
mae_lr_test = mean_absolute_error(y_test, y_test_pred_lr)
mse_lr_test = mean_squared_error(y_test, y_test_pred_lr)
rmse_lr_test = np.sqrt(mse_lr_test)
r2_lr_test = r2_score(y_test, y_test_pred_lr)

# Print the evaluation metrics
print("\nLinear Regression Performance on Train Set:")
print(f"MAE: {mae_lr_train:.3f}")
print(f"MSE: {mse_lr_train:.3f}")
print(f"RMSE: {rmse_lr_train:.3f}")
print(f"R-squared: {r2_lr_train:.3f}")

print("\nLinear Regression Performance on Test Set:")
print(f"MAE: {mae_lr_test:.3f}")
print(f"MSE: {mse_lr_test:.3f}")
print(f"RMSE: {rmse_lr_test:.3f}")
print(f"R-squared: {r2_lr_test:.3f}")

# Plot Actual vs Predicted for the Test Set
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_test_pred_lr, alpha=0.7)
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red')  # Identity line
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual vs Predicted Values - Linear Regression')
```

```
plt.grid(True)
plt.show()

print('RFR')
target = Raw_DataFrame_imputed['Q20']
features = Raw_DataFrame_imputed.drop(columns=['Q20','Industry'])

# Standardize features
scaler = StandardScaler()
features_scaled = scaler.fit_transform(features)

# Feature selection using RFE with RandomForestRegressor
rf_regressor = RandomForestRegressor(random_state=42)
selector = RFE(rf_regressor, n_features_to_select=10, step=1)
features_selected = selector.fit_transform(features_scaled, target)

# Hyperparameter tuning with GridSearchCV
param_grid = {
   'n_estimators': [50, 100, 150],
   'max_depth': [None, 10, 20, 30],
   'min_samples_split': [2, 5, 10],
   'min_samples_leaf': [1, 2, 4],
   'max_features': ['auto', 'sqrt', 'log2']
}

grid_search_rf = GridSearchCV(estimator=RandomForestRegressor(random_state=42),
                              param_grid=param_grid,
                              cv=5,
                              scoring='neg_mean_absolute_error',
```

```
                    n_jobs=-1,
                    verbose=2)
grid_search_rf.fit(features_selected, target)

# Best parameters found by GridSearchCV
best_params_rf = grid_search_rf.best_params_
print(f"\nBest parameters found by GridSearchCV: {best_params_rf}")

# Refit the RandomForestRegressor with the best parameters
best_rf = grid_search_rf.best_estimator_

# Evaluate the best model on the training and test sets
X_train, X_test, y_train, y_test = train_test_split(features_selected, target, test_size=0.3,
random_state=42)

# Train the best model on the training set
best_rf.fit(X_train, y_train)

# Predict and evaluate the model on the train and test sets
y_train_pred_rf = best_rf.predict(X_train)
y_test_pred_rf = best_rf.predict(X_test)

# Model evaluation metrics for train set
mae_rf_train = mean_absolute_error(y_train, y_train_pred_rf)
mse_rf_train = mean_squared_error(y_train, y_train_pred_rf)
rmse_rf_train = np.sqrt(mse_rf_train)
r2_rf_train = r2_score(y_train, y_train_pred_rf)

# Model evaluation metrics for test set
mae_rf_test = mean_absolute_error(y_test, y_test_pred_rf)
```

```
mse_rf_test = mean_squared_error(y_test, y_test_pred_rf)
rmse_rf_test = np.sqrt(mse_rf_test)
r2_rf_test = r2_score(y_test, y_test_pred_rf)

print("\nRandom Forest Regressor Performance on Train Set (Best Model):")
print(f"MAE: {mae_rf_train:.3f}")
print(f"MSE: {mse_rf_train:.3f}")
print(f"RMSE: {rmse_rf_train:.3f}")
print(f"R-squared: {r2_rf_train:.3f}")

print("\nRandom Forest Regressor Performance on Test Set (Best Model):")
print(f"MAE: {mae_rf_test:.3f}")
print(f"MSE: {mse_rf_test:.3f}")
print(f"RMSE: {rmse_rf_test:.3f}")
print(f"R-squared: {r2_rf_test:.3f}")

# Plot Actual vs Predicted for the Test Set
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_test_pred_rf, alpha=0.7)
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red')  # Identity line
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual vs Predicted Values - Random Forest Regressor (Best Model)')
plt.grid(True)
plt.show()


print('GBR')
```

```
# Set Q20 as the target variable
target = Raw_DataFrame_imputed['Q20']
features = Raw_DataFrame_imputed.drop(columns=['Q20', 'Industry'])

# Standardize features
scaler = StandardScaler()
features_scaled = scaler.fit_transform(features)

# Hyperparameter tuning with GridSearchCV for Gradient Boosting Regressor
param_grid = {
   'n_estimators': [100, 200, 300],
   'learning_rate': [0.01, 0.05, 0.1],
   'max_depth': [3, 4, 5],
   'min_samples_split': [2, 5, 10],
   'min_samples_leaf': [1, 2, 4],
   'subsample': [0.8, 0.9, 1.0]
}

grid_search_gbr =
GridSearchCV(estimator=GradientBoostingRegressor(random_state=42),
                        param_grid=param_grid,
                        cv=5,
                        scoring='neg_mean_absolute_error',
                        n_jobs=-1,
                        verbose=2)
grid_search_gbr.fit(features_scaled, target)

# Best parameters found by GridSearchCV
best_params_gbr = grid_search_gbr.best_params_
print(f"\nBest parameters found by GridSearchCV: {best_params_gbr}")
```

```
# Refit the Gradient Boosting Regressor with the best parameters
best_gbr = grid_search_gbr.best_estimator_

# Evaluate the best model on the training and test sets
X_train, X_test, y_train, y_test = train_test_split(features_scaled, target, test_size=0.3, random_state=42)

# Train the best model on the training set
best_gbr.fit(X_train, y_train)

# Predict and evaluate the model on the train and test sets
y_train_pred_gbr = best_gbr.predict(X_train)
y_test_pred_gbr = best_gbr.predict(X_test)

# Model evaluation metrics for train set
mae_gbr_train = mean_absolute_error(y_train, y_train_pred_gbr)
mse_gbr_train = mean_squared_error(y_train, y_train_pred_gbr)
rmse_gbr_train = np.sqrt(mse_gbr_train)
r2_gbr_train = r2_score(y_train, y_train_pred_gbr)

# Model evaluation metrics for test set
mae_gbr_test = mean_absolute_error(y_test, y_test_pred_gbr)
mse_gbr_test = mean_squared_error(y_test, y_test_pred_gbr)
rmse_gbr_test = np.sqrt(mse_gbr_test)
r2_gbr_test = r2_score(y_test, y_test_pred_gbr)

print("\nGradient Boosting Regressor Performance on Train Set (Best Model):")
print(f"MAE: {mae_gbr_train:.3f}")
print(f"MSE: {mse_gbr_train:.3f}")
```

```
print(f"RMSE: {rmse_gbr_train:.3f}")
print(f"R-squared: {r2_gbr_train:.3f}")

print("\nGradient Boosting Regressor Performance on Test Set (Best Model):")
print(f"MAE: {mae_gbr_test:.3f}")
print(f"MSE: {mse_gbr_test:.3f}")
print(f"RMSE: {rmse_gbr_test:.3f}")
print(f"R-squared: {r2_gbr_test:.3f}")

# Plot Actual vs Predicted for the Test Set
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_test_pred_gbr, alpha=0.7)
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red')  # Identity line
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual vs Predicted Values - Gradient Boosting Regressor (Best Model)')
plt.grid(True)
plt.show()


# Assuming X_train, X_test, y_train, y_test have been defined earlier

# Hyperparameter tuning using GridSearchCV for LR, RFR, and GBR

# 1. Linear Regression (No hyperparameters to tune, but included for completeness)
lr = LinearRegression()
lr.fit(X_train, y_train)
y_train_pred_lr = lr.predict(X_train)
y_test_pred_lr = lr.predict(X_test)
```

```
best_score_lr = r2_score(y_train, y_train_pred_lr)
best_params_lr = 'N/A'

# 2. Random Forest Regressor
rf_params = {
    'n_estimators': [50, 100, 150],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'max_features': ['auto', 'sqrt', 'log2']
}
rf = GridSearchCV(RandomForestRegressor(random_state=42), rf_params, cv=5)
rf.fit(X_train, y_train)
best_params_rf = rf.best_params_
best_score_rf = rf.best_score_

# 3. Gradient Boosting Regressor
gbr_params = {
    'n_estimators': [100, 200, 300],
    'learning_rate': [0.01, 0.05, 0.1],
    'max_depth': [3, 4, 5],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'subsample': [0.8, 0.9, 1.0]
}
gbr = GridSearchCV(GradientBoostingRegressor(random_state=42), gbr_params, cv=5)
gbr.fit(X_train, y_train)
best_params_gbr = gbr.best_params_
best_score_gbr = gbr.best_score_
```

```
# Create Hyperparameter Table for Best Parameters (Table 3)
hyperparameters_data = {
    'Model': ['Linear Regression', 'RandomForestRegressor', 'GradientBoostingRegressor'],
    'Best Parameters': [best_params_lr, best_params_rf, best_params_gbr],
    'Best Score': [best_score_lr, best_score_rf, best_score_gbr]
}
hyperparameters_df = pd.DataFrame(hyperparameters_data)
print("TABLE 3: Hyperparameter Table for Best Parameters:")
print(hyperparameters_df)
```

## APPENDIX C – SMART METER DATASET QUESTIONS

Q1 - Question 1002: My organisation is interested in changing the way we use electricity if it reduces the electricity bill

Q2 - Question 1002: By changing the way the people I work with and I use electricity, my organisation can reduce its electricity bill

Q3 - Question 10012: My organisation has already done a lot to reduce the amount of electricity it uses

Q4 - Question 10012: My organisation has already made changes to the way we work in order to reduce the amount of electricity we use.

Q5 - Question 10012: My organisation would like to do more to reduce electricity usage

Q6 - Question 10012: My organisation knows what it needs to do in order to reduce electricity usage

Q7 - Question 10012: My organisation cannot control how much electricity it uses

Q8 - Question 170: Approximately what percentage of non wage costs does your organisation spend on electricity? ASSUME BUSINESS PROFILE, HOURS OF BUSINESS HAVE NOT CHANGED, EQUIPMENT, OPENING HOURS HAVE NOT CHANGED.

Q9 - Question 300: Has your organisation had an audit of its energy use in the past 12 months?

Q10 - Question 301: Is your organisation curently on a nightsaver tariff?

Q11 - Question 320: Does your organisation have regular monitoring and reporting of trends in its energy use?

Q12 - Question 340: Is there a person in your organisation who is responsible for energy usage checking, monitoring and feeding back?

Q13 - Question 350: Does your organisation engage in generating its own electricity from sources such as solar panels, wind turbines etc?

Q14 - Question 6012: My business has become more aware of the cost of the electricity it uses

Q15 - Question 6012: My business has invested in more energy efficient equipment to reduce its use of electricity of usage

Q16 - Question 6012: My business reviewed its electricity usage in order to identify ways of reducing it

Q17 - Question 6012: My business identified easy to implement changes which reduced the amount of electricity it used

Q18 - Question 6012: Managers were given targets to reduce their use of electricity

Q19 - Question 6012: Participation in the trial had no impact on the business

Q20 - Question 6020: By what amount do you think that your electricity bills changed as a result of the trial?

Q21 - Question 6030: Do you think that your overall electricity usage (units or kWh) changed during the trial?

Q22 - Question 6502: The tariff (the different prices charged for using electricity at different times of day and night) helped the business to reduce its overall usage of electricity

Q23 - Question 6502: The tariff made the business become more aware about how it used electricity

Q24 - Question 9030: When you first received this energy usage statement how effective did you find it in helping your business reduce its electricity usage, using a scale of 1 to 5 where 5 is very effective and 1 is not at all effective or 6 did not read it?

Q25 - Question 9040: And over the entire trial how effective did you find the energy usage statement in helping your business reduce its electricity usage using a scale of 1 to 5 where 5 is very effective and 1 is not at all effective or 6 did not read it?

Q26 - Question 20000: Have you installed timing switches on any of your electrical appliances/equipment?

Q27 - Question 200007: Is there a person in your organisation who is responsible for energy usage checking, monitoring and feeding back?

Q28 - Question 200008: Does your organisation engage in generating its own electricity from sources such as solar panels, wind turbines etc?

Q29 - Question 5432: My organisation is interested in changing the way we use electricity if it reduces the electricity bill

Q30 - Question 5432: By changing the way the people I work with and I use electricity, I can reduce my electricity bill

Q31 - Question 5442: My organisation has already done a lot to reduce the amount of electricity it uses

Q32 - Question 5442: My organisation has already made changes to the way we work in order to reduce the amount of electricity we use.

Q33 - Question 5442: My organisation would like to do more to reduce electricity usage

Q34 - Question 5442: My organisation knows what it needs to do in order to reduce electricity usage

Q35 - Question 5442: My organisation cannot control how much electricity it uses

Q36 - Question 5422: Approximately what % savings on the average bill did you achieve?

Q37 - Question 5512: The overall quality of the electricity supply

Q38 - Question 5512: The percentage of electricity being generated from renewable sources

Q39 - Question 5512: The number of estimated bills your organisation receives

Q40 - Question 5512: The level of wastage of electricity in the home or office

Q41 - Question 5512: The overall cost of electricity

Q42 - Question 5512: The environmental damage associated with the amount of electricity used

**APPENDIX D - ETHICAL APPROVAL REVIEW**

# SAGE-HDR (v3.8 24/04/23)

| Response ID | Completion date |
|---|---|
| 1046015-1045997-124621054 | 1 Jul 2024, 20:18 (BST) |

| | | |
|---|---|---|
| 1 | **Applicant Name** | Swetha Vijayanadhan |
| 1.a | **University of Surrey email address** | sv00633@surrey.ac.uk |
| 1.b | **Level of research** | Postgraduate Taught (Masters) |
| 1.b.i | **Please enter your University of Surrey supervisor's name. If you have more than one supervisor, enter the details of the individual who will check this submission.** | Ali Emrouznejad |
| 1.b.ii | **Please enter your supervisor's University of Surrey email address. If you have more than one supervisor, enter the details of the supervisor who will check this submission.** | a.emrouznejad@surrey.ac.uk |
| 1.c | **School or Department** | Surrey Business School |
| 1.d | **Faculty** | FASS - Faculty of Arts and Social Sciences |

| | | |
|---|---|---|
| 2 | **Project title** | Visualizing, Classifying and Predicting Smart Meter Electricity / Gas Data for SMEs and industries |

| | | |
|---|---|---|
| 3 | **Please enter a brief summary of your project and its methodology in 250 words. Please include information such as your research method/s, sample, where your research will be conducted and an overview of the aims and objectives of your research.** | Project Summary<br>The research project aims to develop advanced analytics tools for visualizing, classifying, and predicting smart meter electricity and gas data specifically for Small and Medium-sized Enterprises (SMEs) and industries. By leveraging methodologies used in household energy studies, the project seeks to optimize energy management, reduce costs, and promote sustainable practices within the industrial sector.<br>Methodology<br>The study employs a quantitative research method, focusing on large datasets of energy consumption. Data will be collected from smart meters installed in SMEs and industrial settings. Advanced data analytics techniques, including machine learning algorithms and statistical models, will be used to analyse this data. The research will involve the application of unsupervised clustering methods and supervised classification algorithms to identify consumption patterns and predict future energy use.<br>Sample and Research Location<br>The sample includes SMEs and industrial facilities equipped with smart meters. The research will be conducted in various industrial sectors to ensure a comprehensive understanding of energy consumption patterns across different types of businesses. |

| | | |
|---|---|---|
| | | Aims and Objectives<br>1. Visualize Energy Data: Develop interactive visualization tools to help SMEs and industries understand their energy consumption patterns.<br>2. Classify Consumption Behaviours: Use advanced data analytics to classify different types of energy consumption behaviours in industrial settings.<br>3. Predict Energy Use: Create predictive models to forecast future energy consumption based on historical data.<br>4. Optimize Energy Management: Provide actionable insights to SMEs and industries to optimize their energy use, reduce costs, and implement sustainable practices.<br>This research is significant as it extends the application of smart meter data analytics from residential to industrial settings, providing SMEs and industries with the tools needed to enhance their energy management strategies and contribute to overall sustainability goals. |

| | | |
|---|---|---|
| 4 | **Are you planning to join on to an existing Standard Study Protocol (SSP)? SSPs are overarching pre-approved protocols that can be used by multiple researchers investigating a similar topic area using identical methodologies. Please note, SSPs are only being used by 3 schools currently and cannot be used by other schools. Using an SSP requires permission and sign-off from the SSP owner** | NO |

| | | |
|---|---|---|
| 5 | **Are you making an amendment to a project with a current University of Surrey favourable ethical opinion or approval in place?** | NO |

| | | |
|---|---|---|
| 6 | **Does your research involve any animals, animal data or animal derived tissue, including cell lines?** | NO |

| | | |
|---|---|---|
| 8 | **Does your project involve human participants (including human data and/or any human tissue*)?** | NO |

| | | |
|---|---|---|
| 9 | **Will you be accessing any organisations, facilities or areas that may require prior permission? This includes organisations such as schools (Headteacher authorisation), care homes (manager permission), military facilities, closed online forums, private social media pages etc. This also includes using University mailing lists (admin permission). If you are unsure, please contact ethics@surrey.ac.uk.** | NO |

| | | |
|---|---|---|
| 10 | **Does your project involve any type of human tissue research? This includes Human Tissue Authority (HTA) relevant, or non-relevant tissue (e.g. non-cellular such as plasma or serum), any genetic material, samples that have been previously collected, samples being collected directly from the donor or obtained from another researcher, organisation or commercial source.** | NO |

| | | |
|---|---|---|
| 11 | **Does your research involve exposure of participants to any hazardous materials e.g. chemicals, pathogens, biological agents or does it involve any activities or locations that may pose a risk of harm to the researcher or participant?** | NO |

| | | |
|---|---|---|
| 12 | **Will you be importing or exporting any samples (including human, animal, plant or microbial/pathogen samples) to or from the UK?** | NO |

| | | |
|---|---|---|
| 13 | **Will any participant visits be taking place in the Clinical Research Building (CRB)? (involving clinical procedures; if only visiting the CRB to collect/drop-off equipment or to meet with the research team (i.e. for informed consent/discussion) select 'NO').** | NO |

| | | |
|---|---|---|
| 14 | **Will you be working with any collaborators or third parties to deliver any aspect of the research project?** | NO |

| | | |
|---|---|---|
| 15 | **Are you conducting a service evaluation or an audit? Or using data from a service evaluation or audit?** | NO |

| | | |
|---|---|---|
| 16 | **Does your funder, collaborator or other stakeholder require a mandatory ethics review to take place at the University of Surrey?** | NO |
| 17 | **Does your research involve accessing students' results or performance data? For example, accessing SITS data.** | NO |
| 18 | **Will ANY research activity take place outside of the UK?** | NO |
| 19 | **Are you undertaking security-sensitive research, as defined in the text below?** | NO |
| 20 | **Does your project require the processing of special category1 data?** | NO |
| 21 | **Have you selected YES to one or more of the above governance risk questions on this page (Q10-Q20)?** | NO |

| | | |
|---|---|---|
| 22 | **Does your project process personal data2? Processing covers any activity performed with personal data, whether digitally or using other formats, and includes contacting, collecting, recording, organising, viewing, structuring, storing, adapting, transferring, altering, retrieving, consulting, marketing, using, disclosing, transmitting, communicating, disseminating, making available, aligning, analysing, combining, restricting, erasing, archiving, destroying.** | NO |

| | | |
|---|---|---|
| 23 | **Are you using a platform, system or server external to the University approved platforms (Outside of Microsoft Office programs, Sharepoint, OneDrive Qualtrics, REDCap, JISC online surveys (BOS) and Gorilla)** | NO |

| | | |
|---|---|---|
| 24 | **Does your research involve any of the above statements? If yes, your study may require external ethical review or regulatory approval** | NO |

| | | |
|---|---|---|
| 25 | **Does your research involve any of the above? If yes, your study may require external ethical review or regulatory approval** | NO |

| | | |
|---|---|---|
| 26 | **Does your project require ethics review from another institution? (For example: collaborative research with the NHS REC, the Ministry of Defence, the Ministry of Justice and/or other universities in the UK or abroad)** | NO |

| | | |
|---|---|---|
| 27 | **Does your research involve any of the following individuals or higher-risk methodologies? Select all that apply or select 'not applicable' if no options apply to your research. Please note: the UEC reviewers may deem the nature of the research of certain high risk projects unsuitable to be undertaken by undergraduate students** | NOT APPLICABLE - none of the above high-risk options apply to my research. |

| | | |
|---|---|---|
| 28 | **Does your research involve any of the following individuals or medium-risk methodologies? Select all that apply or select 'not applicable' if no options apply to your research.** | NOT APPLICABLE - none of the above medium-risk options apply to my research. |

| | | |
|---|---|---|
| 29 | **Does your research involve any of the following individuals or lower-risk methodologies? Select all that apply or select 'not applicable' if no options apply to your research.** | NOT APPLICABLE - none of the above lower-risk options apply to my research. |

| | | |
|---|---|---|
| **30** | **Declarations** | • I confirm that I have read the University's Code on Good Research Practice and ethics policy and all relevant professional and regulatory guidelines applicable to my research and that I will conduct my research in accordance with these.<br>• I confirm that I have provided accurate and complete information regarding my research project<br>• I understand that a false declaration or providing misleading information will be considered potential research misconduct resulting in a formal investigation and subsequent disciplinary proceedings liable for reporting to external bodies<br>• I understand that if my answers to this form have indicated that I must submit an ethics and governance application, that I will NOT commence my research until a Favourable Ethical Opinion is issued and governance checks are cleared. If I do so, this will be considered research misconduct and result in a formal investigation and subsequent disciplinary proceedings liable for reporting to external bodies.<br>• I understand that if I have selected 'YES' on any governance risk questions and/or have selected any options on the higher, medium or lower risk criteria then I MUST submit an ethics and governance application (EGA) for review before conducting any research. If I have NOT selected any governance risks or selected any of the higher, medium or lower ethical risk criteria, I understand I can proceed with my research without review and |

| | | |
|---|---|---|
| | | acknowledge that my SAGE answers and research project will be subject to audit and inspection by the RIGO team at a later date to check compliance. |

| | | |
|---|---|---|
| **31** | **If I am conducting research as a student:** | • I confirm that I have discussed my responses to the questions on this form with my supervisor to ensure they are correct.<br>• I confirm that if I am handling any information that can identify people, such as names, email addresses or audio/video recordings and images, I will adhere to the security requirements set out in the relevant Data Protection Policy |