Predicting the Energy Density of Potential Battery Materials

Mahfuz Ahmed Abir – 30116946

June 5th, 2025

UNIVERSITY OF SOUTH WALES

PRIFYSGOL DE CYMRU

FACULTY OF COMPUTING, ENGINEERING & SCIENCE

SCHOOL OF COMPUTING & MATHEMATICS

STATEMENT OF ORIGINALITY

I hereby certify that, except where otherwise indicated through proper citation, the work presented in this project is the result of my own independent research and investigation. This project has not been submitted, either in whole or in part, for any other academic award or qualification, except as a requirement for the degree of MSc Data Science at the University of South Wales.

Mahfuz Ahmed Abir____ (student)

6/6/2025____

Acknowledgements

Undertaking this postgraduate Master's project has provided me with valuable insight into the field of Machine Learning within Data Science. I am deeply grateful for the opportunity to explore this area through independent research and practical application.

I would like to express my sincere thanks to my project supervisor, **Abigail Peters**, for her consistent support, guidance, and encouragement throughout the course of this project. I thoroughly enjoyed the Applied Statistics module under her instruction, and our regular meetings over the past few months have been instrumental in shaping the direction and development of my work. Her supervision has been truly invaluable.

I would also like to extend my appreciation to my Machine Learning instructors, **Samuel Jobbins** and **Ieuan Griffiths**. Sam provided a strong foundation in the various Machine Learning techniques that I have applied in this project, and his teaching played a critical role in my understanding of the subject. Although Deep Learning was not part of this specific MSc project, the insights shared by Ieuan greatly enhanced my ability to approach the problem and create an effective solution.

Furthermore, I would like to thank the academic staff in the **School of Computing and Mathematics** at the University of South Wales for their dedication and support throughout my postgraduate studies. The past year has been both enriching and rewarding, and I look forward to applying the knowledge and skills I have gained in my future career in Data Science.

Working on this project has been a truly enjoyable experience, and I will miss it greatly. I only wish there had been more time to explore this exciting field even further.

Abstract

The development of high-performance battery materials is essential for advancing energy storage technologies in response to the growing demand for sustainable power solutions. This project applies Machine Learning techniques to predict the energy density of various battery compounds based on their electrochemical and material properties. The dataset used in this study consists of hundreds of unique materials, encompassing both anode and cathode types, with features such as capacity per gram, voltage, molecular weight, and efficiency.

A supervised learning approach was adopted to model the relationship between these input features and the energy density of the materials. Multiple regression algorithms were employed, including Linear Regression, Random Forest, Support Vector Regression (SVR), and XGBoost, to assess predictive accuracy and robustness. Among these, ensemble-based models demonstrated particularly strong performance.

In addition to prediction, unsupervised learning methods were used to explore patterns and groupings within the dataset. Clustering techniques such as K-Means and Agglomerative Clustering helped uncover natural groupings of materials with similar energy profiles, offering valuable insights into material categorization and potential application areas.

The study highlights how machine learning can effectively support the identification and evaluation of promising battery materials, providing a data-driven approach to accelerate discovery and development in energy research.

Table of Contents

List of Tables and Figures

Tables (In order of appearance)

Figures (In order of appearance)

Chapter 1: Introduction

In ancient Norway, whenever the sky roared with a thunder, the people would look up to the skies and call upon Thor, the mythological God of Thunder and Lightning. That was the only plausible explanation during those times. A mystical being in the sky hammering away at the clouds to create bright flashes of light, illuminating the skies as it came pacing down to earth. Humanity has come a long way since then.

Through the kites of Benjamin Franklin, who first managed to tame lighting, electricity has made it into our buildings, cars and even our pockets. Humanity is surrounded by electricity. Almost all the energy sources in the world, renewable and non-renewable, are being converted to electricity by massive power plants. All to supply to the billions of people consuming electricity at a rate that was not previously possible.

Stepping into the 21st Century, our lives had become fast-paced, more complicated and somehow, in a state of "always-on-the-move". The technologies that were once used in the 19th century or even the 20th, have either become obsolete or have evolved to become portable. Cars, computers, telephones, radios, televisions; every single electronic device has morphed into everyday objects that use batteries. Now there are cars that can accelerate faster than ever, phones that can be carried miles away from civilization and still manage to communicate with the other end of the world. It would not have been possible without the invention of batteries.

The increasing demand for high-energy-density batteries has been driven by rapid advancements in electric vehicles, portable electronic devices, and renewable energy storage systems. Now, as humanity stands at the forefront technological advancement, the need for high efficiency, energy dense batteries is driving research in more than one institution. Energy density is a key performance indicator that directly influences a battery's applicability, efficiency, and commercial viability. Accurate prediction of energy density is therefore essential for guiding the design and development of next-generation batteries with enhanced performance characteristics.

Predictive modeling plays a pivotal role in this context by enabling the estimation and optimization of energy density, thereby accelerating the discovery of novel battery materials. Through computational approaches, researchers can significantly reduce their dependency on expensive and time-consuming experimental procedures. This project aims to synthesize current research on energy density, battery material properties, and data-driven modeling methodologies, with a focus on recent advancements, prevailing challenges, and opportunities for future innovation.

Energy density refers to the amount of energy stored per unit volume or mass of a battery. Higher energy density translates to longer-lasting power sources and improved efficiency, making it a key performance indicator in battery research (QuantumScape, 2024) [1]. Traditional lithium-ion batteries have been the dominant technology, but emerging materials such as solid-state electrolytes and alternative anode/cathode compositions are being explored to enhance energy storage capabilities (Thunder Said Energy, 2024) [3]. These advancements aim to address critical challenges such as safety, longevity, and performance degradation over multiple charge-discharge cycles.

Furthermore, advancements in high-nickel cathodes and silicon-based anodes have demonstrated significant potential in increasing energy density (Batteries, 2023) [5]. High-nickel cathodes allow for higher capacity retention, while silicon anodes provide greater lithium-ion storage capabilities compared to traditional graphite anodes. However, issues such as electrode expansion and material stability require further optimization before widespread commercial adoption.

Predictive modeling leverages computational methods, including machine learning and physics-based simulations, to forecast the energy density of novel battery materials. By analyzing large datasets of material properties and experimental results, predictive models can guide material selection and battery design (Nature Scientific Data, 2020) [15]. Recent studies have highlighted the effectiveness of data-driven approaches in predicting electrochemical performance metrics and improving material discovery pipelines (Batteries, 2023) [5]. The integration of artificial intelligence in battery research has also enabled rapid screening of potential material candidates, significantly reducing the time required for new battery technology development.

The aim of this project was to develop and apply Machine Learning models to predict the energy density of battery materials, using a dataset containing various electrochemical and material features. Machine Learning approaches can generally be categorized based on the level of supervision involved during training. The two primary categories are supervised learning and unsupervised learning.

In supervised learning, models are trained on labelled data, where the target variable—in this case, the known energy density—is provided alongside the input features. The goal is to learn a mapping from input features (such as capacity, voltage, conductivity, etc.) to the target variable, allowing accurate predictions on new, unseen materials. Typical supervised learning methods used in this project include regression algorithms such as Linear Regression, Random Forest, Support Vector Regression, and XGBoost.

In contrast, unsupervised learning involves training on data that does not include output labels. This can be useful for exploring underlying patterns in the dataset, such as identifying natural groupings of battery materials or detecting outliers. Common unsupervised methods include clustering, dimensionality reduction, and anomaly detection. For instance, clustering techniques like K-Means or DBSCAN can help group materials with similar electrochemical profiles, potentially revealing new insights into material classification or performance trends.

Chapter 2 provides a comprehensive review of related work in the field, including recent studies and projects that have utilized machine learning techniques for the prediction of energy-related properties in battery materials. Emphasis is placed on the application of supervised and unsupervised learning algorithms in materials science, particularly in the context of energy density prediction and battery material classification. Chapter 3 outlines the data preprocessing procedures undertaken in this study. The dataset, compiled from reputable sources in battery research, underwent extensive cleaning, normalization, and transformation to ensure suitability for machine learning applications. Exploratory Data Analysis (EDA) was also conducted to gain an initial understanding of the data structure, feature distributions, and correlations.

Chapter 4 presents a methodological overview of the machine learning algorithms selected for this research. Both classical and ensemble-based regression models are introduced, with a discussion of their theoretical underpinnings and their applicability to energy density prediction tasks. Based on this review, four regression algorithms were selected for supervised learning.

Chapter 5 focuses on the unsupervised learning component of the project. Principal Component Analysis (PCA) was employed to reduce the dimensionality of the dataset and facilitate visualization of patterns. Clustering algorithms—including K-Means, DBSCAN, Spectral Clustering, and Agglomerative Clustering—were applied to identify potential groupings of battery materials based on their physicochemical and electrochemical profiles.

Chapter 6 details the implementation of the supervised models, which were trained to predict the energy density of battery materials. Model performance was evaluated using key regression metrics such as MAE, RMSE, and R^2. Feature importance analyses were conducted to interpret the influence of individual material properties on model predictions. The goal of this analysis was to discover inherent similarities among materials, which may inform future classification or discovery of novel compounds with desirable energy characteristics.

Chapter 2: Relevant Work

The literature reviewed underscores the importance of predictive modeling in advancing battery energy density research. Machine learning, physics-based simulations, and hybrid methods offer promising avenues for material discovery and battery optimization. However, challenges such as data availability and model accuracy must be addressed to enhance predictive capabilities. Future research should focus on expanding datasets, improving model generalization, and integrating experimental validation with computational predictions. Furthermore, collaboration between academia and industry can drive innovation in battery technology, paving the way for more efficient and sustainable energy storage solutions

QuantumScape (2024) [1] provides an accessible yet informative introduction to energy density, outlining its central role in modern battery technology. The article explores key factors that influence energy density, including the choice of electrode materials and electrolytes. Of particular interest is the discussion on solid-state batteries, which are presented as a promising alternative to traditional lithium-ion systems. These innovations not only offer the potential for significantly higher energy density but also address longstanding safety concerns associated with current technologies.

Expanding on these themes, the Innovation Energy Report (2024) [2] delves into recent breakthroughs in battery engineering. The report highlights how advancement in material science, especially the application of nanotechnology, is driving improvements in storage efficiency. Emphasis is placed on strategies such as reducing internal resistance and optimizing electrode architecture, both of which are critical to enhancing battery performance and extending lifespan.

Complementing these technological perspectives, Nature Scientific Data (2020) [15] introduces a comprehensive dataset covering a wide range of battery materials and their electrochemical characteristics. This resource is particularly valuable for data-driven research, providing the foundation for machine learning models aimed at predicting energy density. By including detailed material compositions, voltage profiles, and cycling behaviors, the dataset supports the growing trend of integrating computational tools into battery research.

The Thunder Said Energy Report (2024) [3] shifts focus on the future of lithium-ion battery development, assessing both the technical hurdles and economic implications of increasing energy density. It underscores the role of predictive modeling in navigating these challenges and examines how factors such as cost optimization, sustainability, and regulatory policy influence the trajectory of battery innovation.

In the academic domain, a recent study published in Batteries (2023) [5] offers a detailed examination of machine learning approaches used to estimate electrochemical properties. The authors explore the effectiveness of algorithms such as gradient boosting and deep neural networks, emphasizing the importance of combining computational modeling with experimental validation. The study also highlights how thoughtful feature selection can enhance both the accuracy and interpretability of predictive models.

Lastly, a comparative analysis in Energy Storage Materials (2021) [6] investigates how different material compositions affect energy density. The paper identifies lithium-sulfur and lithium-air chemistries as especially promising for next-generation batteries. It also draws attention to the importance of electrolyte stability, which plays a critical role in determining the long-term viability of advanced battery systems. The study presents a balanced view by comparing theoretical energy densities with real-world limitations across various battery types.

2.1 Used Methodologies

The existing body of literature outlines a range of methodologies employed in predicting energy density, with a strong focus on data-driven and computational techniques.

Machine Learning Models

Several studies put emphasis on the utility of machine learning (ML) in analyzing extensive datasets of battery materials to identify patterns and relationships between composition and electrochemical performance. Methods such as neural networks, decision trees, and various regression models have been effectively applied to predict key electrochemical properties with considerable accuracy (Nature Scientific Data, 2020 [15]; Batteries, 2023) [5]. These models facilitate the discovery of promising novel materials by accelerating the screening process and reducing the need for extensive experimental testing.

Physics-Based Simulations

Complementing ML approaches, physics-based simulations, including first-principles calculations and density functional theory (DFT) offer theoretical insights into the structural and electronic behavior of battery materials (Innovation Energy, 2024 [2]; Energy Storage Materials, 2021) [6]. These simulations help elucidate the underlying mechanisms governing energy storage and guide the selection of materials for experimental validation, thereby contributing to a more robust scientific understanding.

Hybrid Approaches

An emerging strategy in recent studies involves the integration of machine learning with physics-based simulations. This hybrid modelling approach combines the predictive power of data-driven algorithms with the theoretical depth of computational physics to yield more accurate and generalizable models (QuantumScape, 2024) [1]. Such integrative methods not only improve performance metrics but also accelerate the pace of discovery in battery material research.

2.2 Research Gaps & Challenges

Despite these methodological advances, several persistent challenges continue to hinder the effectiveness and scalability of predictive models in battery research.

Data Availability

One of the most significant limitations lies in the availability and quality of experimental datasets. Many machine learning models suffer from limited generalizability and reduced performance when trained on small or inconsistent data samples (Nature Scientific Data, 2020 [15]; Batteries, 2023) [5]. Expanding access to high-quality, standardized datasets and fostering collaboration through open-data platforms could substantially improve model robustness and reproducibility.

Model Generalization

Another notable challenge is the difficulty in applying models trained on known materials to entirely new chemical systems. Predictive models often exhibit poor generalization when confronted with novel material compositions or structures (Thunder Said Energy, 2024 [3]; Energy Storage Materials, 2021) [6]. Ongoing research into transfer learning techniques and adaptive modelling frameworks may help overcome this limitation by enabling models to learn across different domains.

Computational Costs

Finally, physics-based simulations—particularly those involving DFT—are computationally intensive, often requiring significant processing power and extended runtimes (Innovation Energy, 2024) [2]. These computational demands can limit the scalability of such methods, especially in large-scale screening tasks. The adoption of high-performance computing (HPC) infrastructure and cloud-based simulation tools offers a potential solution to this barrier, facilitating more efficient workflows in computational materials discovery.

Chapter 3: Data Collection, Pre-Processing and EDA

3.1 Overview

This chapter outlines the collection, organization, and preparation of the data used to build the predictive energy density model in battery materials. Beginning with a description of the collected dataset (Huang, S. and Cole, J. (2020)) [1, 15], this section also describes the workflow, which involved multiple stages, including data extraction from raw sources, transformation and cleaning through SAS and Python scripts, and consolidation into a final dataset suitable for analysis. Each workflow phase—from raw data acquisition to final dataset creation—is detailed below.

3.2 Data Description

The initial dataset, composed of raw battery material data, includes properties such as Voltage, Capacity, Coulombic Efficiency, Conductivity, and Energy. Each row describes a property of a compound/chemical composition and includes information regarding the Type of battery material (cathode, anode). Each row also has the compound's chemical name and composition, which was greatly useful in the cleanup process and model building alike. The initial dataset had about 292,000 data points. This was a huge amount of raw data; unsorted, full of missing data, and completely void of any links between each row. Even if a single compound had all the data, it was spread out across multiple rows and with many repetitions. It wasn’t possible to build any machine learning model based on this dataset. To solve this problem, I uploaded the data to SAS OnDemand for Academics platform. I chose this platform as it provides a structured environment with built-in tabular manipulation techniques. With SAS, I filtered the table into 5 separate tables based on the properties of the chemical compounds:

1. **Capacity**: Included charge/discharge capacities measured across various materials.
2. **Voltage**: Records of voltage readings of different battery materials.
3. **Coulombic Efficiency**: Represents the ratio of charge output to input, indicating energy conversion efficiency of each material.
4. **Conductivity**: Measured the ionic/electrical conductivity of materials under test.
5. **Energy**: Records of the energy density of various battery materials.

Each of these tables had key columns such as the name of the compound, the Extracted Name (the chemical composition), the type of battery material, the specific property that the table contained, and the unit of measurement of that property. These tables were exported as individual CSV files for further pre-processing in Python.

3.3 Data Cleaning

The data cleaning process involved several steps, including attempts that were not very favorable. Each exported CSV file underwent multiple steps of cleaning by utilizing Python scripts and was then finally merged for further analysis and model building. All the steps for cleaning and merging are given below:

3.3.1 Data Preparation and Cleaning Workflow

The initial phase of dataset preparation was conducted using a Python script (data_cleaning.py) focused on removing a substantial volume of incomplete records. The original dataset comprised approximately 290,000 rows; however, after rigorous filtering of entries with missing or null values, this number was reduced to around 1,000 rows. This aggressive pruning ensured that subsequent analysis would be based on complete and reliable data.

Following the identification and removal of incomplete entries, cleaned subsets of the data were extracted into distinct tables. Each table corresponded to a specific material property—such as voltage or capacity—and was stored in a CSV file named using the convention "xxxx_clean.csv," where "xxxx" denotes the property in question. Simultaneously, the rows with missing or NaN entries were isolated and preserved in separate datasets for potential use in imputation or secondary analysis. Throughout this process, care was taken to retain the original column structure across all derived tables to ensure consistency.

3.3.2 Feature Engineering: Molecular Weight Calculation

An essential aspect of early feature engineering was the computation of the molecular weight for each chemical compound. This task was carried out using a separate Python script (data_pre_processing.py). The dataset included a column labeled Extracted Name, which contained the chemical formula for each compound. These formulas were parsed using regular expressions and string manipulation techniques to extract elemental components and their respective counts.

To enable this calculation, a predefined dictionary mapping atomic symbols to their atomic weights (e.g., H = 1.008, O = 16.00) was employed. For each compound, the molecular weight was computed by summing the weighted contributions of individual atoms based on their frequency within the formula. This derived feature added a quantifiable measure of chemical complexity to the dataset, serving as a valuable input variable for machine learning models aimed at predicting energy density. Furthermore, the inclusion of molecular weight enabled the use of a joint key—comprising Compound_name and Molecular_weight—to uniquely identify compounds across datasets.

3.3.3 Data Integration Attempts

With the availability of cleaned and enriched tables, efforts were made to integrate data across the five key property datasets. Two Python scripts were written for this purpose. The first, data_combining_vertical.py, attempted vertical stacking to unify entries across different battery material properties. This approach, however, was unsuccessful due to mismatched schemas and inconsistent compound identifiers across the files.

A more refined strategy was implemented in data_merging_for_sql.py, which aimed for horizontal merging by aligning records using compound names or molecular weights. Despite being a more logical method for data fusion, this too failed due to naming inconsistencies and the absence of reliable relational keys. These failed integration attempts underscored the need for stricter standardization of column names and the elimination of irrelevant or redundant fields in each table.

3.3.4 Dataset Refinement and Standardization

The script further_cleaning.py was developed to refine and harmonize the structure of the datasets. This included the application of consistent naming conventions across all tables and the removal of non-informative columns such as internal IDs and measurement units. Only essential fields were retained: Compound_name, Extracted_name, Molecular_weight, Type (designating anode or cathode), and Property (such as voltage, capacity, conductivity, coulombic efficiency, or energy). Furthermore, data type conversions were applied to ensure that all numerical columns were appropriately formatted. These steps significantly improved the internal consistency and usability of the data.

3.3.5 Normalization and Deduplication

To further enhance data integrity and compatibility for merging operations, unique compound entries were extracted using the script find_distinct_compounds.py. This involved filtering each table to retain only distinct combinations of Compound_name and Molecular_weight. The output consisted of five de-duplicated CSV files, each representing one of the material property datasets. These normalized tables facilitated clearer compound-level analysis and laid the groundwork for final data consolidation.

3.3.6 Final Merging, Feature Enhancement, and Cleanup

The final phase of the data preparation pipeline involved two additional scripts: distinct_data_merging.py and final_cleaning_steps.py. The first successfully merged the five normalized tables into a single comprehensive dataset using an outer join on the compound name and molecular weight. This ensured the inclusion of all possible records, even those with partially missing property values. Subsequent refinements were executed in final_cleaning_steps.py. This included the removal of the Conductivity column due to an excessive number of missing entries. The Type column was standardized by replacing inconsistent labels—for instance, rows labeled "Cath" were updated to "Cathode," and those labeled "Anod" were corrected to "Anode." Such standardization was crucial to mitigate downstream issues related to categorical variable encoding and interpretation.

Moreover, all remaining rows containing NA values were removed to ensure the dataset was model-ready. A final and significant feature engineering step involved the computation of the Energy Density column, defined as:

$$\boldsymbol{Energy\,Density}\left(\frac{\boldsymbol{Wh}}{\boldsymbol{kg}}\right)=\boldsymbol{Capacity}\left(\frac{\boldsymbol{mAh}}{\boldsymbol{g}}\right)\times\frac{\boldsymbol{Voltage}\,(\boldsymbol{V})}{\mathbf{1000}}$$

This step was carried out in the later stages of cleaning, to impute the missing value for the Energy property of various materials. As Energy density itself is the most crucial column, it was necessary to have all relevant columns cleaned and free of NULL values. The final dataset was a complete one with 832 entries. This would be large number of data points to carry out model building and any further analysis. From now onwards, this dataset will be referred to as the ***prime dataset***. The initial dataset contained approximately 290,000 rows, which was reduced to around 800 rows after rigorous cleaning and filtering. This drastic reduction was necessary due to the presence of missing, inconsistent, or duplicate values that compromised data quality. However, it is important to acknowledge that this filtering process may have also excluded data points that, while imperfect, could have contributed to the variability and richness of the dataset.

Although the retained subset ensures analytical integrity and model reliability, there is a risk that certain patterns, edge cases, or rare material behaviors may have been lost in the process. Future work may benefit from employing imputation techniques or semi-supervised learning approaches to leverage partially complete data, thus preserving more variance while maintaining robustness.

The final cleaning steps were carried out one more time with a different approach, where instead imputing null values of the Voltage column (which happens to be one of the factors for calculating the Energy density) the null values were dropped altogether. This created an entirely different data set consisting of only 339 data points. From now onwards, this dataset will be referred to as the ***concise dataset***.

3.4 Feature Scaling

One of the most crucial transformations applied to the battery dataset was feature scaling. The dataset contained several numerical features that varied significantly in magnitude and units of measurement. Without scaling or normalization, these discrepancies would likely hinder the performance of most machine learning algorithms, particularly those sensitive to feature magnitude, such as gradient-based methods and distance-based models.

To address this, standardization was employed. This technique transforms the features so that they have a mean of zero and a standard deviation of one. As a result, all numerical variables were brought to a comparable scale, ensuring that no single feature disproportionately influenced the learning process. To implement this, the StandardScaler from the scikit-learn package from Python was utilized. With the help of this package, I was able to standardize the most significant numerical columns, which showed the most range in the data; Molecular_weight, Capacity_per_gram_in_mAh, Voltage_in_V, Efficiency_in_percent. Using this a scaled Dataset was formed, which could be used for any distance-based algorithms. Additionally, I created a second version using MinMaxScaler also from the scikit-learn package from Python; a normalized version of the data, in which all the columns were scaled from 0 to 1 using the minimum and maximum values. This dataset is to be used for training a neural network for classification of battery material. Both the prime-dataset and the concise-dataset have undergone the feature scaling process as this process applies to all rows and columns.

3.5 Exploratory Data Analysis

The prime dataset consisting of 832 battery material candidates, each labeled as either an "Anode" or a "Cathode", includes both physical and electrochemical descriptors such as molecular weight, capacity per gram (in mAh), voltage (in V), efficiency (in percent), and energy density (in Wh/kg). The purpose of this analysis is to examine the statistical structure of the dataset, identify patterns and correlations between key features, and establish a foundation for more advanced modeling and hypothesis generation in battery materials research. These attributes represent a broad range of material characteristics, making the dataset suitable for both classification and regression tasks in the context of materials informatics. Importantly, the dataset is complete, with no missing values across any of the eight columns. This suggests that the data was carefully curated and is ready for direct analysis without the need for imputation or data cleaning. The distribution of entries between the two types is relatively balanced, with 433 cathode materials and 399 anode materials. This balance is advantageous, as it enables a fair comparative analysis between the two material classes without introducing biases stemming from class imbalance.

3.5.1 Summary Statistics

A preliminary statistical summary provides useful insights into the central tendencies and variability of the key numerical features. The average molecular weight of materials in the dataset is approximately 403.63 g/mol, but the standard deviation is unusually high at over 2400 g/mol. This indicates the presence of outliers or a long-tailed distribution, which is confirmed by the maximum molecular weight reaching over 61,000 g/mol. Most of the materials, however, fall below 500 g/mol, suggesting a heavily skewed distribution.

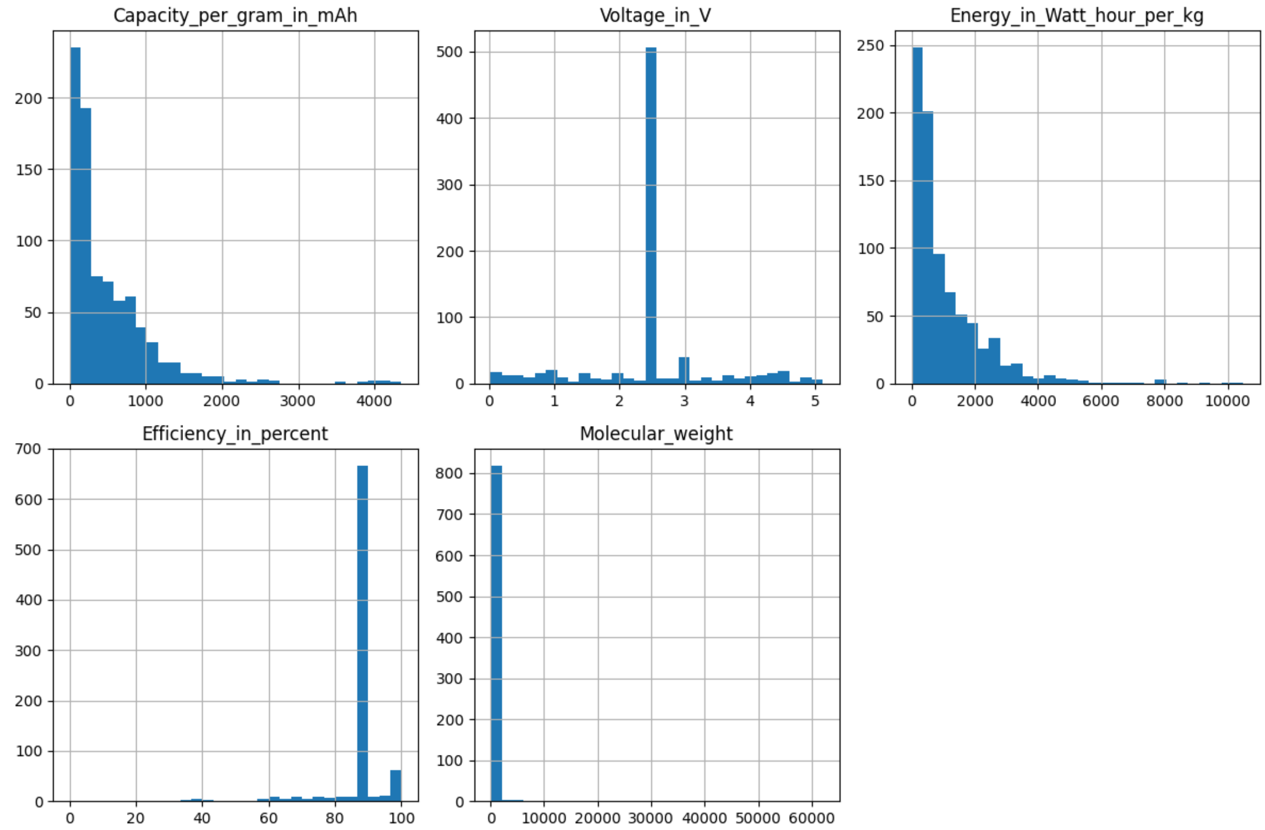
Capacity per gram exhibits an average value of 493.76 mAh/g with a standard deviation of about 564.24 mAh/g, again indicating a considerable spread. The maximum recorded capacity is an impressive 4352 mAh/g, whereas many materials cluster around the lower end, as shown by a median value of 272.5 mAh/g. Voltage, in contrast, is tightly distributed around a common value of 2.5V, with minimal variance across the dataset. This suggests a design preference or experimental standard in voltage configurations for both anode and cathode materials.

Efficiency is highly concentrated around 90%, with a small spread. The minimum efficiency recorded is near zero, which might reflect failed experimental attempts or highly suboptimal materials, though such entries are rare. Finally, the energy density feature shows a wide range of values, from 0.1 Wh/kg to over 10,000 Wh/kg, with a mean of approximately 1096 Wh/kg. The large standard deviation (1303 Wh/kg) again points to substantial variability across the dataset.

Prime-Dataset (#OB=832)							
Feature	**Mean**	**Std Dev**	**Min**	**25%**	**50%**	**75%**	**Max**
Molecular_weight	403.63	2421.86	1.76	98.85	167.28	265.83	61958.81
Capacity_per_gram_in_mAh	493.76	564.24	0.04	135.00	272.50	706.28	4352.00
Voltage_in_V	2.48	0.94	0.02	2.50	2.50	2.50	5.11
Efficiency_in_percent	88.44	9.48	0.04	90.00	90.00	90.00	100.00
Energy_in_Watt_hour_per_kg	1096.10	1303.44	0.10	304.35	600.00	1470.63	10500.00

3.5.2 Univariate Analysis

To explore the distributions of individual features, histograms were generated for capacity, voltage, energy density, efficiency, and molecular weight. The histogram for capacity reveals a distinctly



right-skewed distribution. Many materials exhibit capacities under 1000 mAh/g, while a smaller subset achieves extremely high capacities, extending the tail of the distribution.

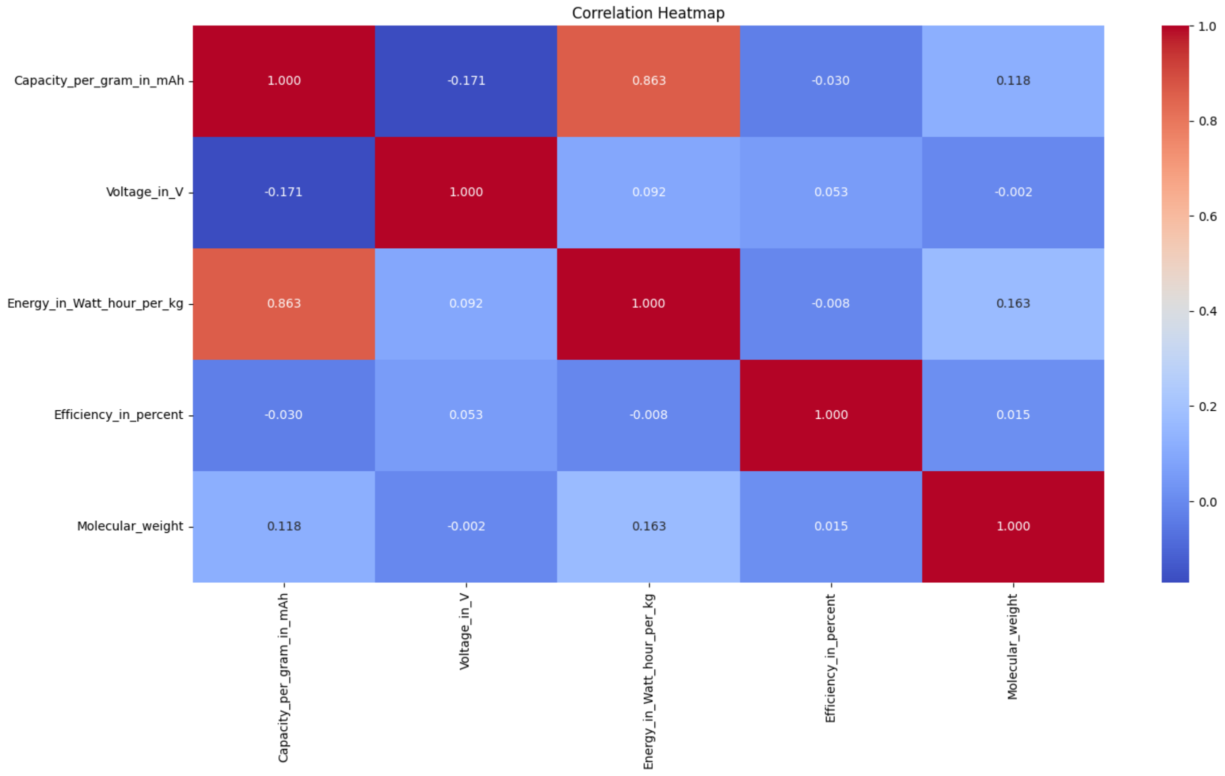
The voltage histogram is quite unique in that it demonstrates a sharp peak at exactly 2.5V. This dominant mode suggests that a large proportion of materials, regardless of type, have been standardized to this voltage—mainly due to the (mean) imputation during the dataset cleaning process. While there are a few entries with voltages exceeding 3.0V, these are relatively rare.

There are clusters at both low and moderate-to-high energy densities, which may indicate the presence of multiple material families or chemistries. The histogram for Energy density shows that materials that appear in the lower mode represent early-stage or low-performing compounds, while those in the upper mode are more promising candidates for high-performance applications.

Efficiency is tightly clustered, with most values hovering around 90%. This reflects either consistently high performance across materials or potential rounding in reported values. Either way, efficiency offers limited discrimination between compounds due to its narrow variance. Molecular weight, as mentioned earlier, displays a heavily skewed distribution, with most values falling below 500 g/mol and a few extending into the tens of thousands.

3.5.3 Bivariate Analysis

To understand how the different numerical features relate to one another, a correlation heatmap was created.



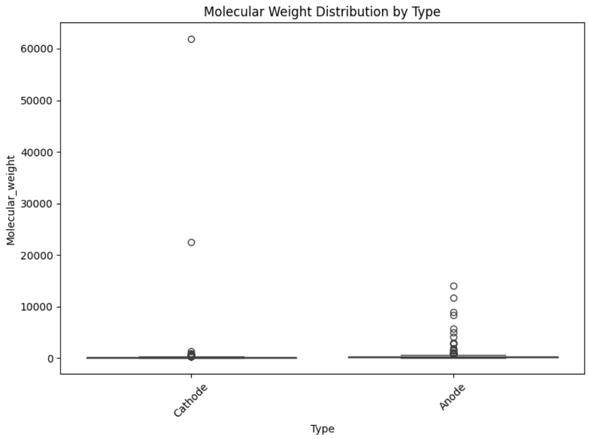
One of the most prominent relationships observed is the strong positive correlation between energy density and capacity, with a correlation coefficient near 0.863. This is expected, as energy is fundamentally a product of capacity and voltage, and in this dataset, capacity appears to be the more variable of the two inputs.

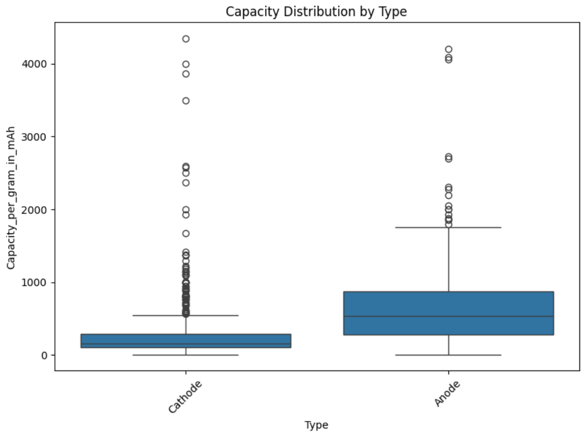
According to the **prime dataset**, there is little to no correlation between energy density and voltage (approximately 0.092), suggesting that voltage has very little contribution to the variance of energy density. The relationship between capacity and voltage is weaker still, indicating that high-capacity materials do not necessarily operate at high voltages, and vice versa.

Molecular weight, interestingly, exhibits weak or no correlation with any other feature. This suggests that molecular weight is not a good predictor of a material's electrochemical properties, at least not in a direct or linear manner. Its strongest correlation (0.163) is with Energy density. This means that even though the effect is low, it does, to some degree, contribute to the variance of the Energy density of a material.

3.5.4 Comparative Analysis by Material Type

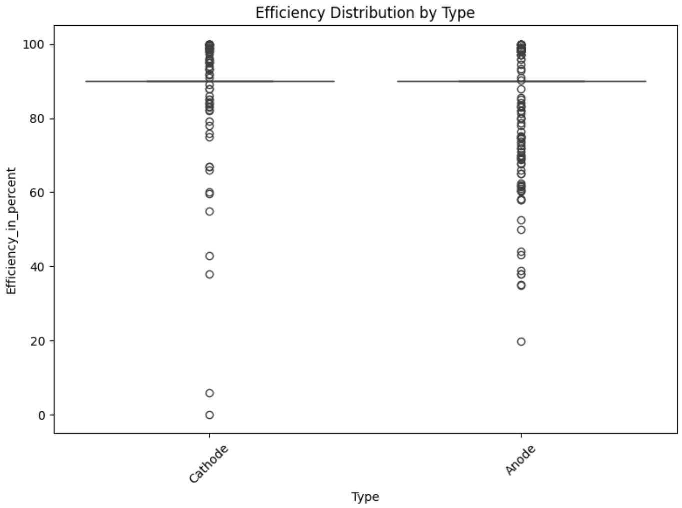
Boxplots were generated to compare the distributions of key features between anode and cathode materials. One of the clearest distinctions appears in molecular weight distributions. Cathode materials generally exhibit higher molecular weights than anodes, though this trend is somewhat confounded by a few outliers. This might reflect the use of heavier transition-metal compounds or polymeric frameworks in cathode designs.

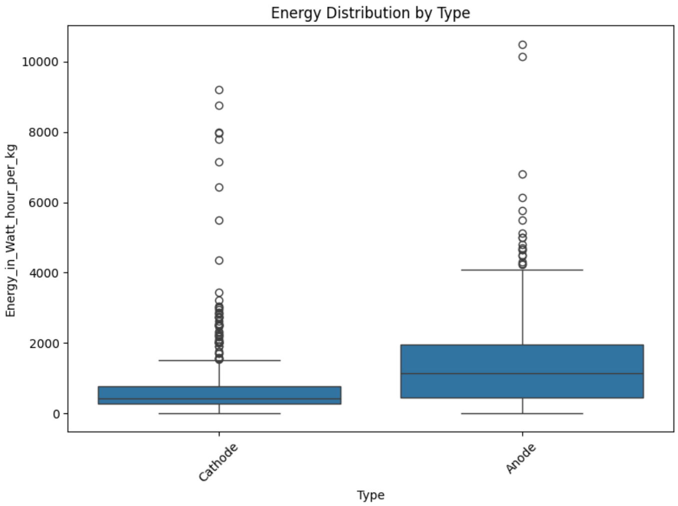




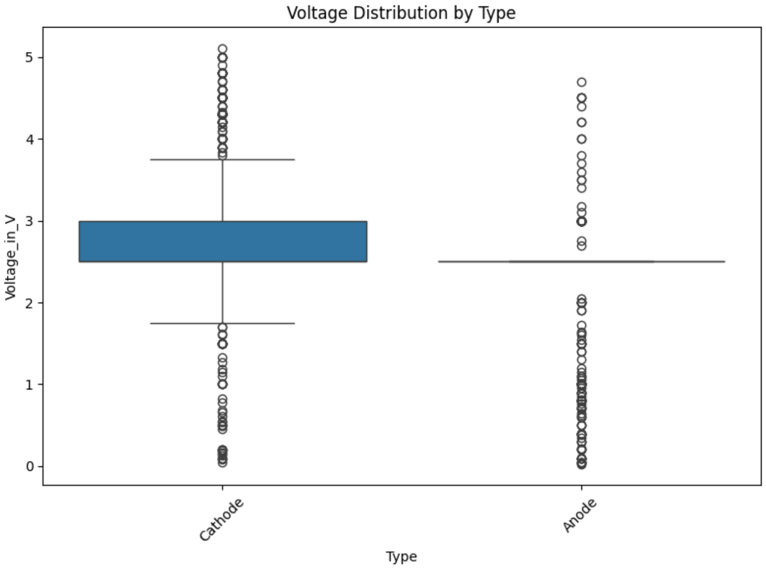
The difference in capacity is even more pronounced. Cathodes tend to have both higher median capacities and a wider spread, indicating a broader range of performance among cathode candidates. Anodes, in contrast, show a more constrained distribution with lower average capacities. This suggests that the cathode space may offer more room for innovation or variability, whereas anode capacities are more tightly optimized.

Efficiency values are similar across both types, with most materials clustering at or near 90%. There is no significant distinction in efficiency between anodes and cathodes, implying that this metric may not be particularly useful for classification or early-stage screening.





When examining energy density, the superiority of cathodes becomes evident. Not only do they achieve higher average energy densities, but they also include several extreme outliers that suggest potential for ultra-high performance. Anodes, by comparison, exhibit a tighter and lower distribution. The voltage distributions follow a similar pattern, with cathodes having a slightly wider and higher voltage range, though both material Types peak around 2.5V.

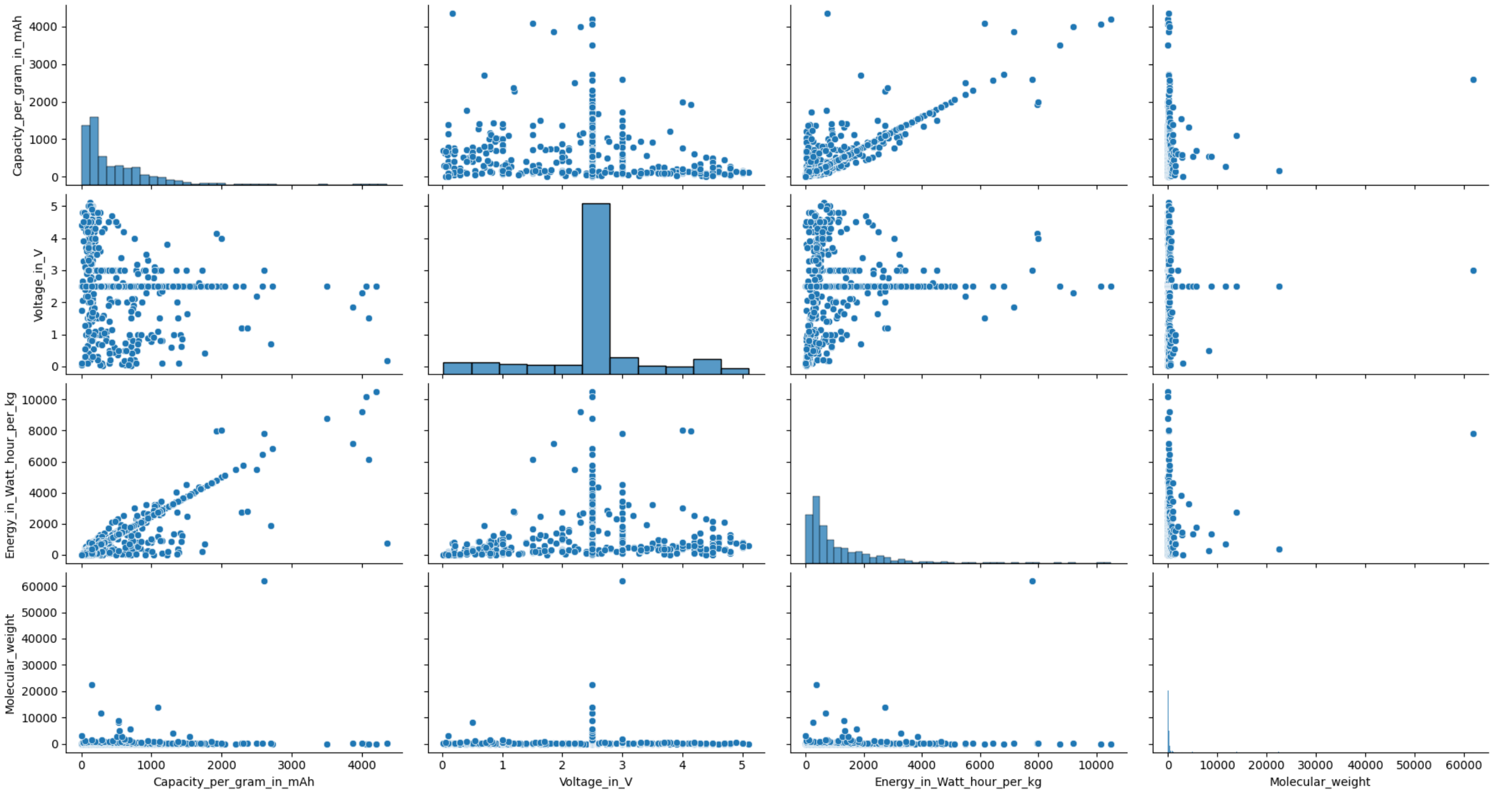


3.5.5 Multivariate Analysis

To explore multivariate relationships and potential patterns, a pair plot of the top numerical features was constructed. This visualizes scatter plots of all pairwise feature combinations, along with their respective distributions. The most striking observation is the linear relationship between capacity and energy density, which is consistent with earlier correlation findings.

Voltage introduces vertical spread in the energy vs. capacity plots, illustrating how voltage amplifies or attenuates the energy output of a material with a given capacity. However, voltage itself does not appear to provide strong discriminatory power when plotted against other features.

Molecular weight, once again, fails to exhibit strong linear associations with other variables, and its points appear scattered without forming clusters or gradients. No obvious material groupings or clusters are evident in the pair plot, indicating that the dataset may require more sophisticated dimensionality reduction techniques—such as Principal Component Analysis (PCA) or t-distributed Stochastic Neighbor Embedding (t-SNE)—to uncover latent structures.



This exploratory data analysis (EDA), encompassing 832 battery materials categorized as anodes and cathodes, provides significant insights into their electrochemical and structural properties. The dataset's robust curation and balanced distribution ensure the reliability of statistical comparisons. While most features, notably energy density and capacity, exhibit substantial variability, others, such as efficiency and voltage, demonstrate narrower distributions.

Cathode materials consistently surpass anode materials in both capacity and energy density, albeit often possessing higher molecular weights. The voltage distribution implies a degree of standardization across the dataset, potentially reflecting practical design constraints in battery development. A critical finding is the strong positive correlation between capacity and energy, indicating that capacity enhancement is a primary driver for improved energy performance.

Notably, molecular weight appears to have no direct influence on the analyzed performance metrics. This suggests that more intricate chemical features may be necessary for accurate modeling and prediction of battery performance.

This EDA establishes a robust foundation for future research, particularly in applying machine learning techniques to predict material performance or classify compounds. Subsequent work could involve unsupervised learning for cluster discovery or regression models to predict energy density from molecular descriptors. The analysis underscores the inherent complexity and rich structure within battery material datasets.

Chapter 4: Methodology

This chapter outlines the machine learning techniques used for predicting the energy density of battery materials using the given dataset. We explore various supervised regression algorithms, drawing on guidance from the textbook Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow by Aurélien Géron [7], and insights from recent literature in battery informatics and materials science.

4.1 Linear Regression

Linear Regression is a foundational algorithm in machine learning, commonly used for predicting a continuous target variable based on one or more input features. In this model, a linear relationship is assumed between the predictors (features like molecular weight, capacity, voltage, etc.) and the response variable (energy density). The prediction function is defined as a weighted sum of input features:

$$\hat{y} = w_1x_1 + w_2x_2 + \ldots + w_nx_n + b$$

where w_i are the weights and b is the bias term. Linear Regression serves as a useful baseline due to its simplicity and interpretability. However, its performance may be limited in the presence of multicollinearity or non-linear relationships.

4.2 Random Forest Regressor

Random Forest is a tree-based ensemble learning method that builds multiple decision trees and aggregates their outputs. Each tree is trained on a bootstrapped subset of the training data, and only a random subset of features is considered at each split. This reduces overfitting and variance, making it ideal for tabular data with mixed types.

$$\hat{y} = \left(\frac{1}{T}\right)\sum h_t(x)$$

where $h_t(x)$ is the prediction of the t^{th} tree. In previous materials science applications (e.g., Jain et al., 2013[10]), Random Forests have demonstrated excellent predictive performance for complex physical properties due to their robustness to noise and non-linear relationships.

4.3 Support Vector Regressor (SVR)

Support Vector Regression (SVR) extends the ideas of Support Vector Machines to regression tasks. The goal is to find a function that deviates from actual values by at most a predefined margin ε, while remaining as flat as possible.

$$f(x) = \sum \alpha_i K(x_i, x) + b$$

where K is the kernel function. While powerful, SVR can be sensitive to feature scaling and parameter tuning.

4.4 XGBoost Regressor

XGBoost (Extreme Gradient Boosting) is an optimized implementation of gradient boosting algorithms, designed for speed and performance. It builds models sequentially, with each model attempting to correct the residuals of the previous one.

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta\, f_t(x_i)$$

where f_t is the new model added at iteration t, and η is the learning rate. XGBoost has shown state-of-the-art results in many machine learning competitions and has recently been applied to materials property prediction due to its ability to capture complex, non-linear interactions among features (Chen et al., 2016) [13].

4.5 Chosen Methods

Based on the nature of the dataset and previous literature, the following four regression algorithms were selected for predicting the energy density of battery materials:

1. Linear Regression: Baseline model, high interpretability.
2. Random Forest Regressor: Robust to overfitting, handles non-linearity well.
3. Support Vector Regressor: Capable of modeling complex relationships.
4. XGBoost Regressor: High performance, ensemble model with regularization.

Each model will be evaluated using cross-validation and metrics such as Root Mean Squared Error (RMSE), R^2 Score, and Mean Absolute Error (MAE). The models will also be compared based on their generalization ability to unseen data.

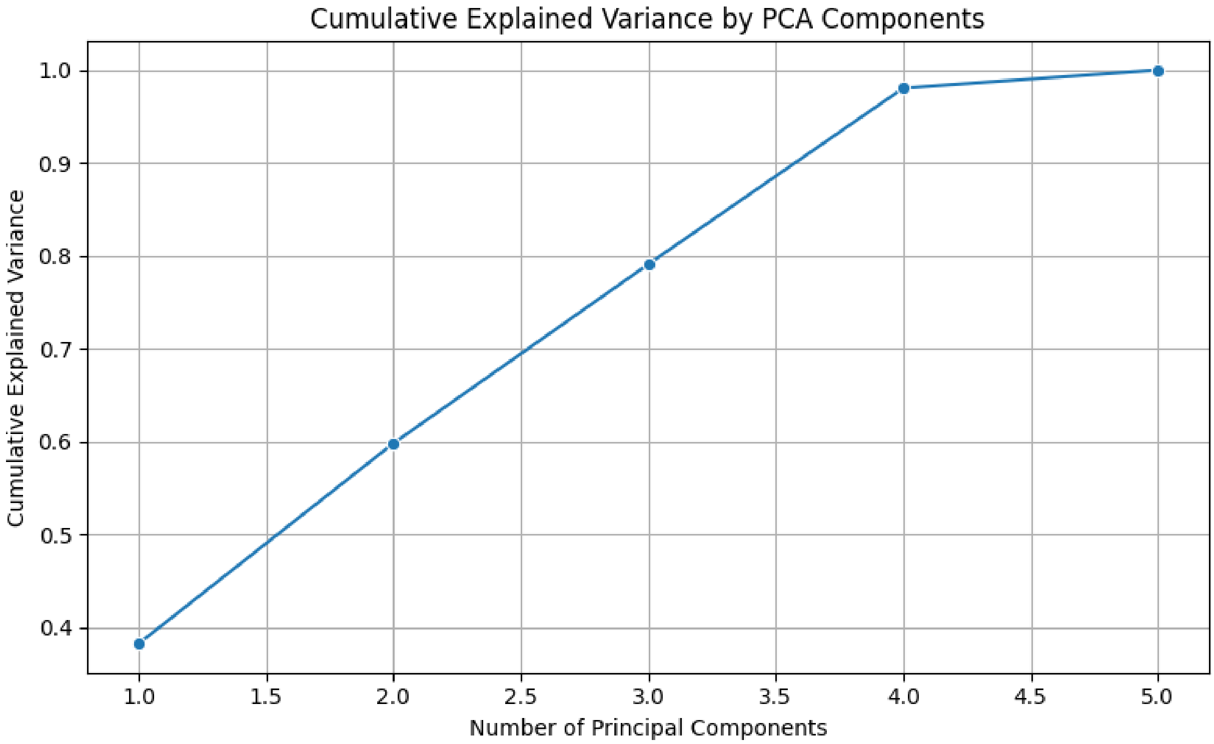
Chapter 5: Unsupervised Learning

The first step to creating a predictive model for battery energy density was the application of unsupervised learning techniques to the dataset of battery materials. The primary objective was to explore the inherent structure within the dataset by identifying natural groupings of materials without relying on predefined labels. Through dimensionality reduction and clustering algorithms, this chapter aims to uncover insights into the characteristics that differentiate battery materials and evaluate the strengths and weaknesses of each method used.

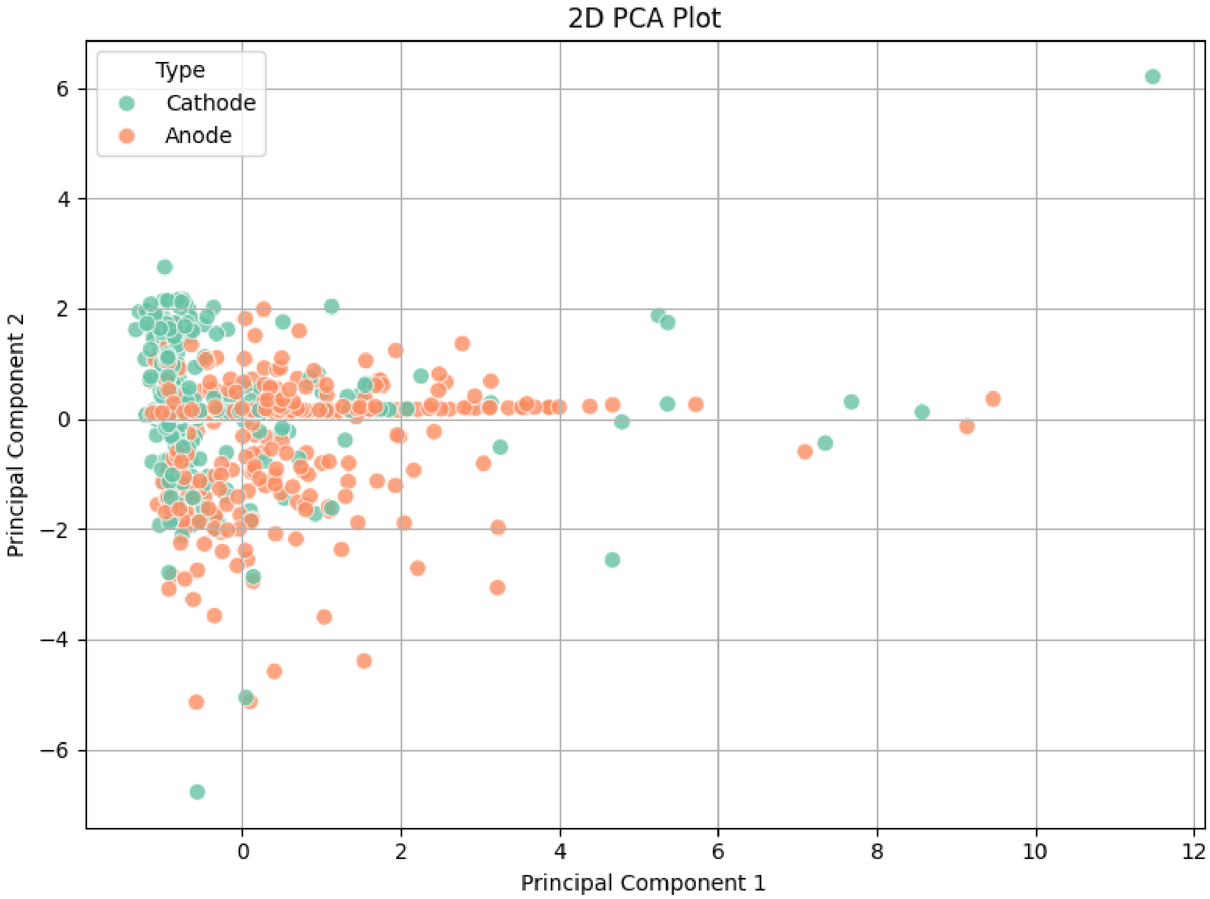
5.1 Dimensionality Reduction

5.1.1 Principal Component Analysis (PCA)

Principal Component Analysis was employed to simplify the dataset by transforming the original variables into a smaller number of uncorrelated components while retaining most of the original variability. The dataset included five numerical features: Molecular Weight, Capacity per Gram (mAh/g), Voltage (V), Efficiency (%), and Energy Density (Wh/kg). Before applying PCA, all variables were standardized to ensure that each feature contributed equally to the analysis.



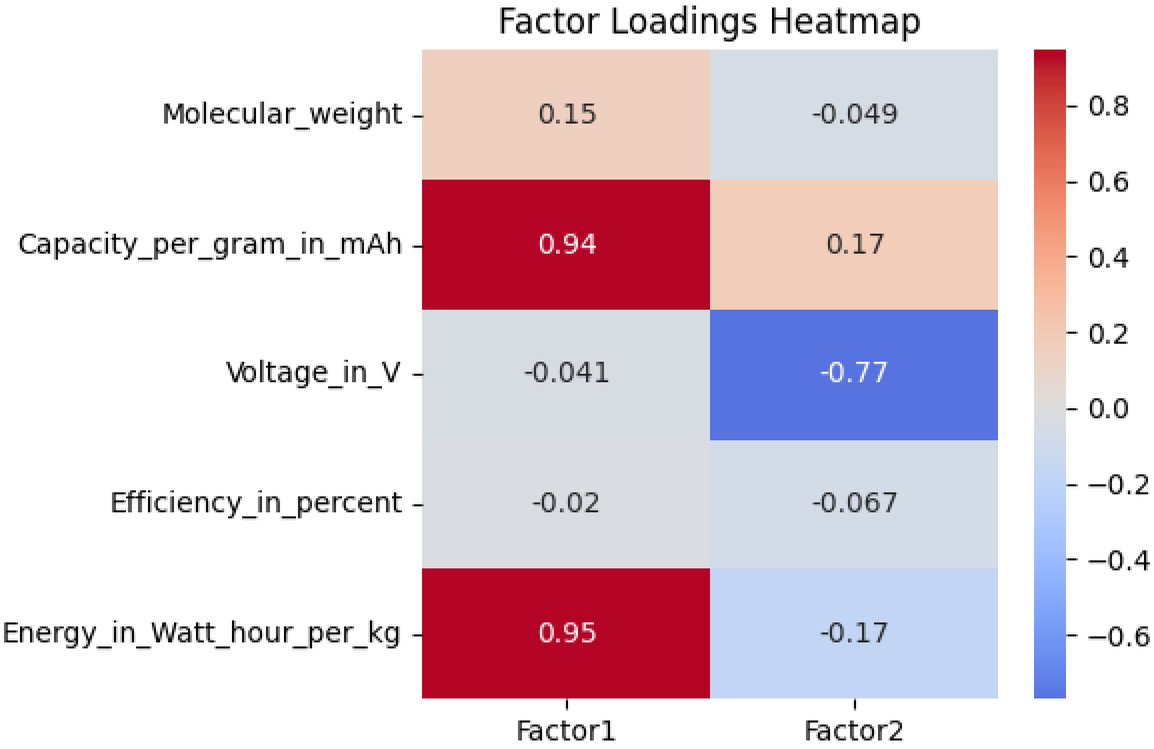
The PCA results revealed that the first two principal components explained around 60% of the total variance in the data. This strong cumulative variance justified the decision to reduce the dimensionality to two components for visual analysis. The third component adds only ~19% more — which is a smaller gain compared to the first two. A third component would require 3D plotting, making it harder to visually communicate patterns or clusters. Reducing to 2 dimensions simplifies clustering or classification tasks, helping avoid overfitting and reducing noise from less informative components. The 2D PCA scatter plot showed discernible clusters, suggesting that materials with similar electrochemical behaviors were naturally grouped together. Notably, the first component was most influenced by Capacity and Energy Density, while the second component reflected variation in Efficiency and Voltage. These axes provided an interpretable projection of high-dimensional data, facilitating downstream clustering.



5.1.2 Factor Analysis

To investigate the latent structures potentially driving the correlations among features, Factor Analysis was conducted using two underlying factors. This statistical technique aims to identify latent constructions that explain observed correlations, which may correspond to broader conceptual groupings like energy output potential and operational stability.

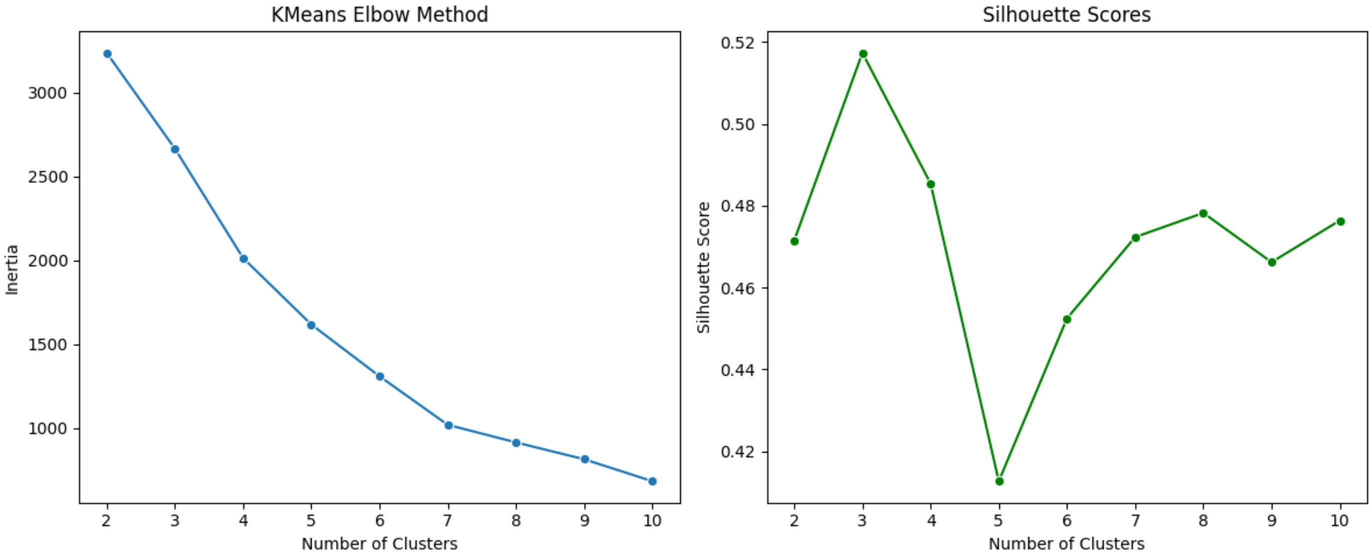
The factor loadings indicated that Factor 1 was strongly aligned with Energy Density and Capacity, suggesting an underlying trait related to energy storage capacity. Factor 2 was more strongly associated with Efficiency and Voltage, potentially representing operational performance or conversion efficiency. While Factor Analysis enhanced the interpretability of variable interrelationships, it did not yield as clear a visual separation as PCA. Nevertheless, the insights from this method reinforced the thematic distinctions in the dataset and validated the choice of variables used in clustering.



5.2 Clustering Techniques and Evaluation

5.2.1 K-Means Clustering

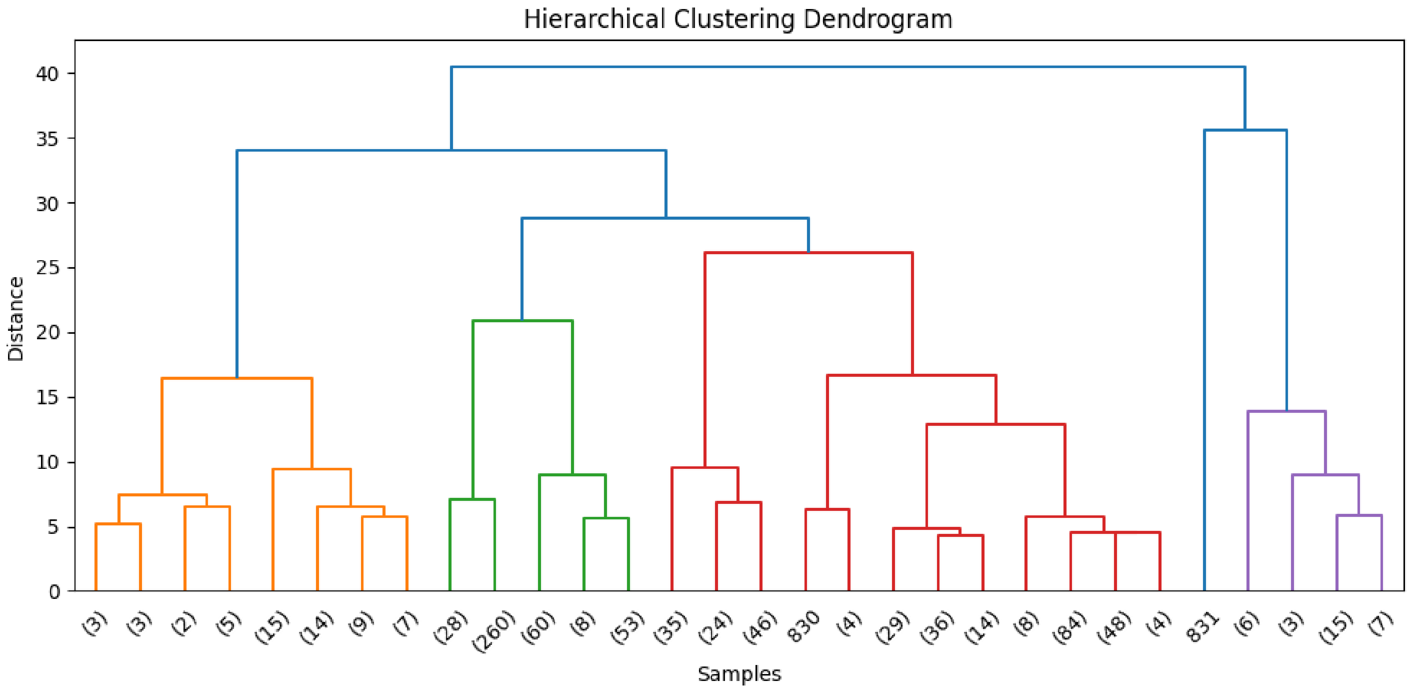
K-Means clustering was first applied to identify groups of materials with similar feature profiles. This method partitions the dataset into non-overlapping clusters by minimizing intra-cluster variance. To determine the optimal number of clusters, both the Elbow Method and Silhouette Score were employed. The Elbow plot showed a pronounced bend at four clusters, which corresponded to the peak silhouette score, indicating optimal clustering structure.



The resulting four clusters, when projected onto the PCA space, were well-separated and formed distinct groupings. Each cluster could be interpreted based on dominant properties such as high energy density or exceptional efficiency. However, K-Means assumes that clusters are spherical and of similar size, which may not always reflect real-world variability. In this case, the assumption was reasonable, as evidenced by the clustering quality and coherence across multiple dimensions.

5.2.2 Hierarchical Clustering

Agglomerative Hierarchical Clustering using Ward's linkage was applied to gain insights into the nested structure of the dataset. The method iteratively merged clusters to minimize within-cluster variance. The dendrogram generated from this process suggested a natural division into four clusters, in agreement with the K-Means results.

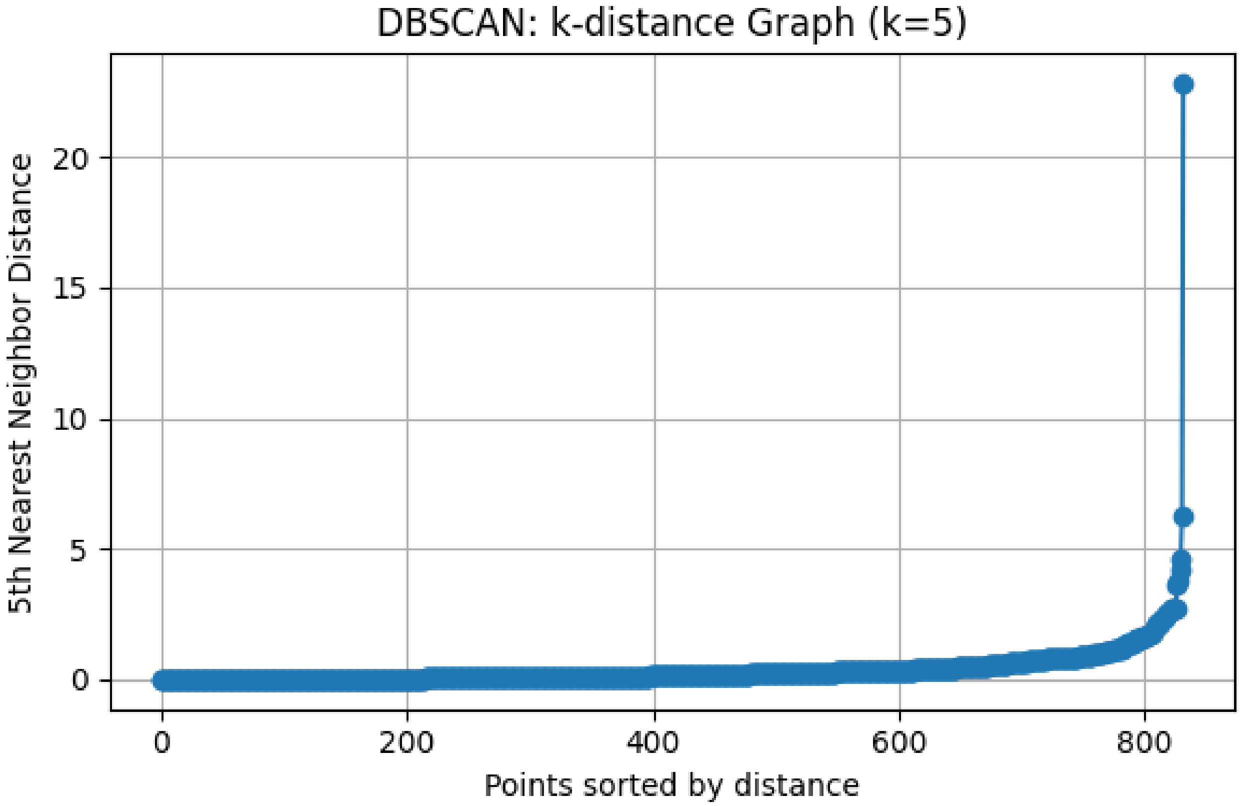


Although the silhouette score for Hierarchical Clustering was slightly lower than that of K-Means, the cluster formation process added interpretability. For instance, some clusters merge late in the hierarchy, suggesting stronger dissimilarity and supporting their distinction in the final partition. A notable limitation of this method is its computational complexity and inability to reassign observations once merged. Still, it served as a valuable validation tool and offered a layered understanding of cluster relationships.

5.2.3 DBSCAN

The DBSCAN algorithm was used to explore the data's density-based structure and to identify outliers. Unlike K-Means or Hierarchical Clustering, DBSCAN does not require the number of clusters to be predefined and can discover arbitrarily shaped clusters. A k-distance graph helped identify a suitable eps value, with 2 being selected based on the observed elbow point. A minimum sample size of 5 was used, reflecting common practice.

While DBSCAN detected a few meaningful clusters, it also identified a significant number of points as noise or outliers. These included materials with extreme property values or those not conforming to dominant patterns. The silhouette score for DBSCAN was higher than the other clustering algorithms, mainly due to the exclusion of noise from the ***prime dataset***. The noise could be the data points that are extreme in nature. The method's strength lies in its robustness to outliers and flexibility in capturing non-linear cluster boundaries. For datasets with more irregular group structures or greater density variation, DBSCAN could provide superior insights.



5.3 Critical Evaluation

Each technique applied offered unique benefits and challenges. PCA emerged as an indispensable tool for data visualization and provided a highly interpretable two-dimensional projection of the dataset. The clarity of group separations in PCA plots justified its use as a basis for visualizing clustering results. Factor Analysis contributed by revealing 2 factors that contribute to the underlying variance of the data, though it was more valuable for interpretation than for visualization or clustering support.

Among the clustering algorithms, DBSCAN was the most effective in terms of silhouette score and interpretability. Unlike others it did not have a requirement for a predefined number of clusters. It also effectively dealt with noise (extreme outliers) Hierarchical Clustering closely mirrored K-Means results and provided additional context through its dendrogram, although its scalability remains a concern for larger datasets. DBSCAN, while being the most effective of the three, captured the broader cluster structure and was successful in flagging unusual or unique materials and may be particularly useful in anomaly detection contexts.

Overall, the consistency of findings across PCA, DBSACN, and Hierarchical Clustering enhances confidence in the results. The choice of 2 factors appears robust and well-supported by both quantitative and qualitative assessments. Similarly, DBSCAN algorithm revealed that the 2 clusters formed are well defined and well separated. Each method complemented the others, contributing to a more comprehensive understanding of the dataset.

5.4 Conclusion

The application of unsupervised learning techniques to the battery materials dataset revealed a well-defined structure, with materials naturally falling into four clusters based on shared physicochemical properties. The alignment between PCA visualizations and clustering outcomes highlighted the coherence of the data. Factor Analysis deepened the interpretability of these findings by suggesting latent factors driving material behavior.

K-Means and Hierarchical Clustering emerged as the most effective clustering methods, both identifying similar groupings. DBSCAN added value by highlighting potential outliers and offering an alternative, density-based perspective. The convergence of results across methods indicates a stable and meaningful cluster structure, which provides a strong foundation for further investigations.

These findings have practical implications for the classification and selection of battery materials. By identifying distinct material categories, researchers can better target experiments and optimize battery performance. Future work may build on these results through supervised learning approaches, time-dependent modeling, or integration with domain-specific knowledge to guide the development of next-generation energy storage materials.

Chapter 6: Supervised Learning

6.1 Introduction

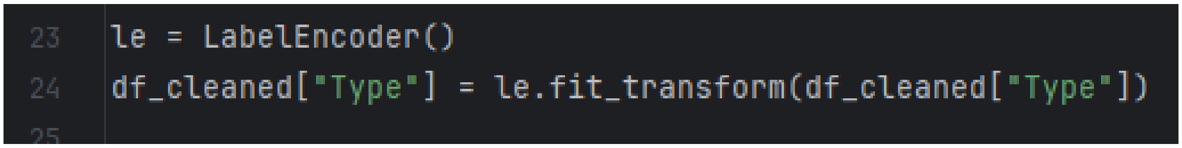
Supervised learning is a subdomain of machine learning widely used in predictive modeling tasks where labeled data is available. This method includes training algorithms on a dataset with known outputs to predict outcomes on new, unseen data. In the context of materials science, particularly battery research, supervised learning can play a pivotal role in predicting key material properties such as energy density based on a compound's chemical and physical attributes.

In this study, supervised learning techniques are applied to predict the **energy density (in Wh/kg)** of battery materials using a dataset comprising multiple chemical and structural descriptors. The primary goal is to identify the most effective model for accurate prediction among several supervised learning algorithms, including Linear Regression, Random Forest Regressor, Support Vector Regressor (SVR), and Extreme Gradient Boosting (XGBoost). In this chapter, we discuss the process of how the data was used for training the models and evaluating the results. The code snippets are added as demonstration to the reader.

6.2 Data Preprocessing

6.2.1 Encoding and Scaling

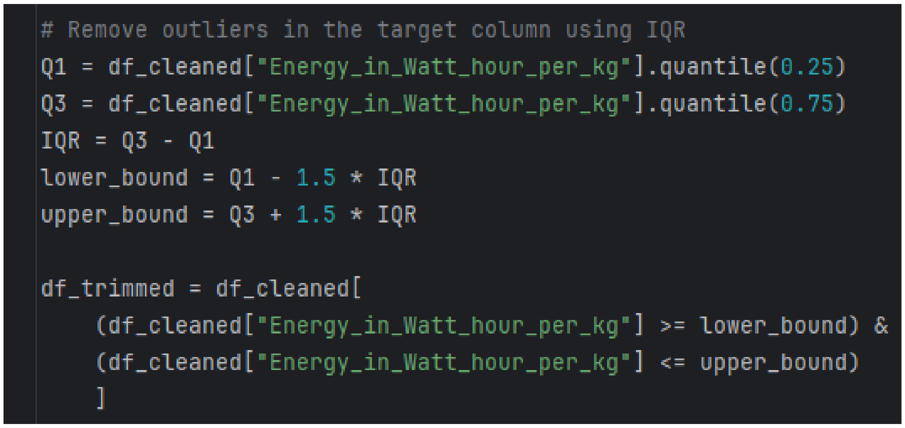
The categorical column "Type" was label-encoded to transform it into a binary numerical format (0 for Anode and 1 for Cathode). Additionally, since Support Vector Machines and other kernel-based methods are sensitive to feature scales, standard scaling was applied to ensure normalized input features during modeling.



```
le = LabelEncoder()
df_cleaned["Type"] = le.fit_transform(df_cleaned["Type"])
```

6.2.2 Outlier Detection and Removal

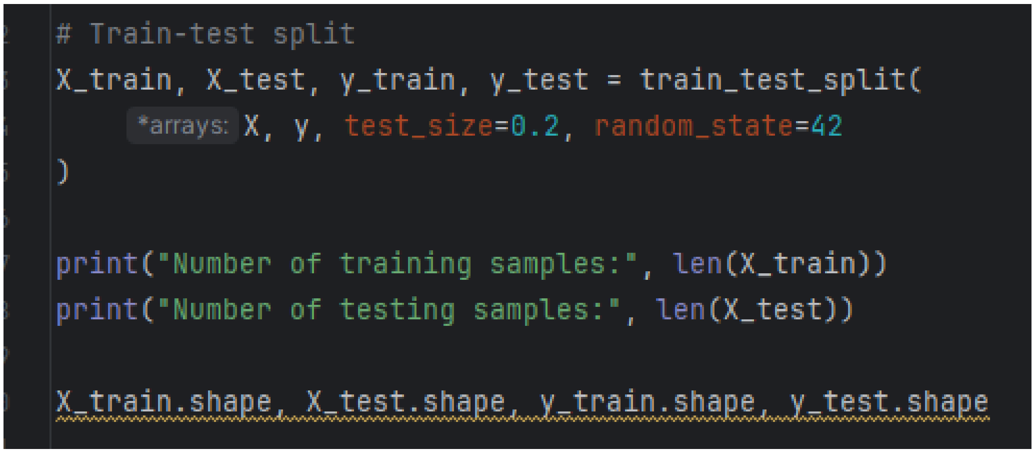
To enhance model performance and reduce the influence of noise, outliers in the target variable were removed using the interquartile range (IQR) method. Observations falling outside 1.5 times the IQR from the first and third quartiles were excluded.



```
# Remove outliers in the target column using IQR
Q1 = df_cleaned["Energy_in_Watt_hour_per_kg"].quantile(0.25)
Q3 = df_cleaned["Energy_in_Watt_hour_per_kg"].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

df_trimmed = df_cleaned[
    (df_cleaned["Energy_in_Watt_hour_per_kg"] >= lower_bound) &
    (df_cleaned["Energy_in_Watt_hour_per_kg"] <= upper_bound)
    ]
```

6.2.3 Train-Test Split and Shaping

The data was split such that 80% was used to train the models and the remaining 20% was used to test the various regression models. This yielded 626 instances for training the models and 157 instances for testing the models. Each algorithm would therefore be trained with 626 instance features to predict 626 labels. The model would then be used on the remaining 1836 instance's test features to make label (disposition) predictions. These label predictions would be compared to the given test labels to determine how well the model performed.

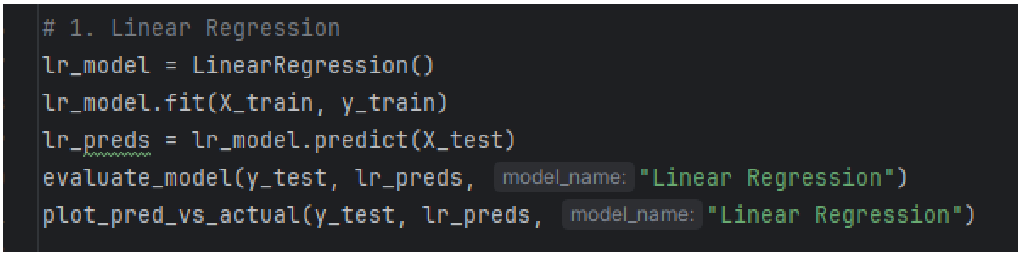


```
# Train-test split
X_train, X_test, y_train, y_test = train_test_split(
     *arrays: X, y, test_size=0.2, random_state=42
)

print("Number of training samples:", len(X_train))
print("Number of testing samples:", len(X_test))

X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

6.3. Modeling Approaches

6.3.1 Linear Regression

As an example of how the models were trained, here are some code snippets to understand the process better. It is interpretable and computationally efficient, making it an ideal baseline model.

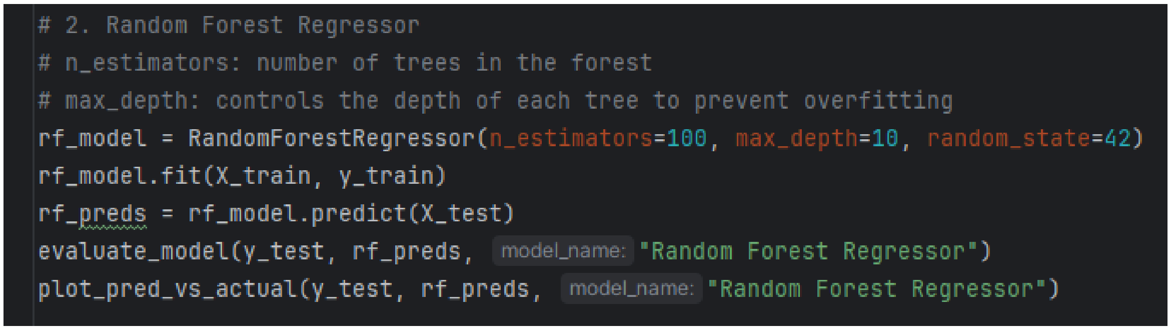


```
# 1. Linear Regression
lr_model = LinearRegression()
lr_model.fit(X_train, y_train)
lr_preds = lr_model.predict(X_test)
evaluate_model(y_test, lr_preds, model_name: "Linear Regression")
plot_pred_vs_actual(y_test, lr_preds, model_name: "Linear Regression")
```

The model was trained using the least squares approach. The evaluation metrics for all the models will be discussed in detail in the model evaluation section.

6.3.2 Random Forest Regressor

Random Forest is an ensemble learning method that constructs multiple decision trees and outputs the mean prediction. It is robust against overfitting and effective in capturing nonlinear relationships.

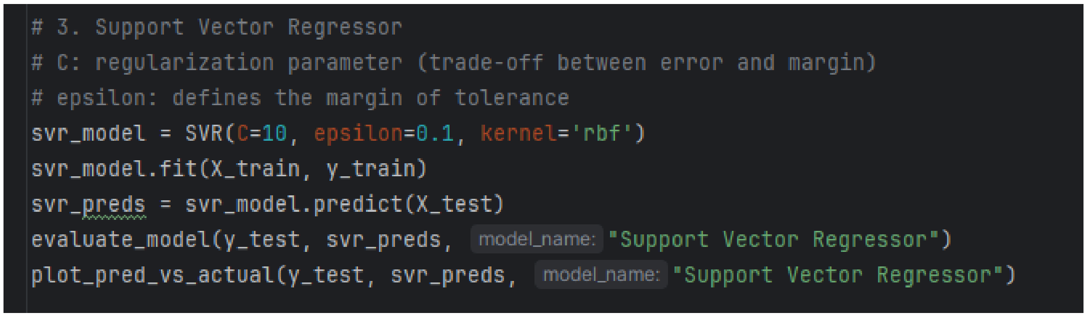


```
# 2. Random Forest Regressor
# n_estimators: number of trees in the forest
# max_depth: controls the depth of each tree to prevent overfitting
rf_model = RandomForestRegressor(n_estimators=100, max_depth=10, random_state=42)
rf_model.fit(X_train, y_train)
rf_preds = rf_model.predict(X_test)
evaluate_model(y_test, rf_preds, model_name: "Random Forest Regressor")
plot_pred_vs_actual(y_test, rf_preds, model_name: "Random Forest Regressor")
```

The Random Forest model significantly outperformed Linear Regression, indicating the presence of complex nonlinear dependencies among the features.

6.3.3 Support Vector Regressor (SVR)

Support Vector Machines aim to find a function that deviates from the target values by a value less than epsilon. With kernel tricks, SVR can handle high-dimensional, nonlinear problems.

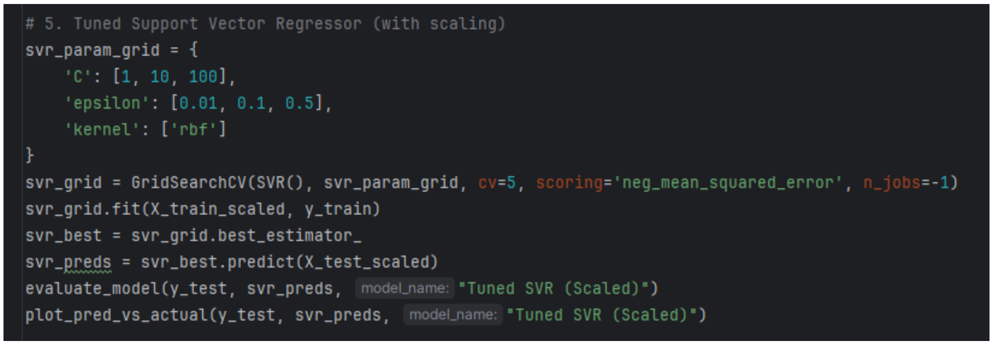


```
# 3. Support Vector Regressor
# C: regularization parameter (trade-off between error and margin)
# epsilon: defines the margin of tolerance
svr_model = SVR(C=10, epsilon=0.1, kernel='rbf')
svr_model.fit(X_train, y_train)
svr_preds = svr_model.predict(X_test)
evaluate_model(y_test, svr_preds, model_name: "Support Vector Regressor")
plot_pred_vs_actual(y_test, svr_preds, model_name: "Support Vector Regressor")
```

The SVR model performed poorly, it was less effective than Linear Regression as well as Random Forest, possibly due to its sensitivity to parameter settings and data scaling.

6.3.4 Tuned Support Vector Regressor

A GridSearchCV approach was adopted to optimize the SVR model across multiple hyperparameters (C, epsilon, and kernel). After tuning, performance improved slightly.

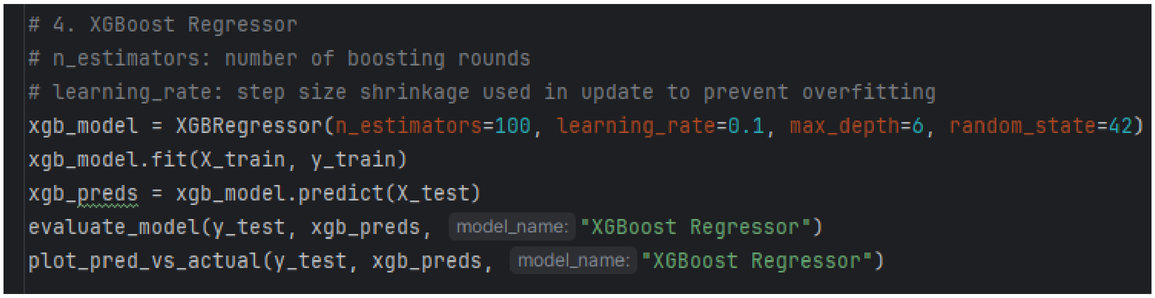


```
# 5. Tuned Support Vector Regressor (with scaling)
svr_param_grid = {
    'C': [1, 10, 100],
    'epsilon': [0.01, 0.1, 0.5],
    'kernel': ['rbf']
}
svr_grid = GridSearchCV(SVR(), svr_param_grid, cv=5, scoring='neg_mean_squared_error', n_jobs=-1)
svr_grid.fit(X_train_scaled, y_train)
svr_best = svr_grid.best_estimator_
svr_preds = svr_best.predict(X_test_scaled)
evaluate_model(y_test, svr_preds, model_name: "Tuned SVR (Scaled)")
plot_pred_vs_actual(y_test, svr_preds, model_name: "Tuned SVR (Scaled)")
```

Hyperparameter tuning yielded modest improvements, reinforcing the idea that SVR requires meticulous configuration. Using the scaled dataset produced more favorable results, outperforming the Linear Regression model.

6.3.5 XGBoost Regressor

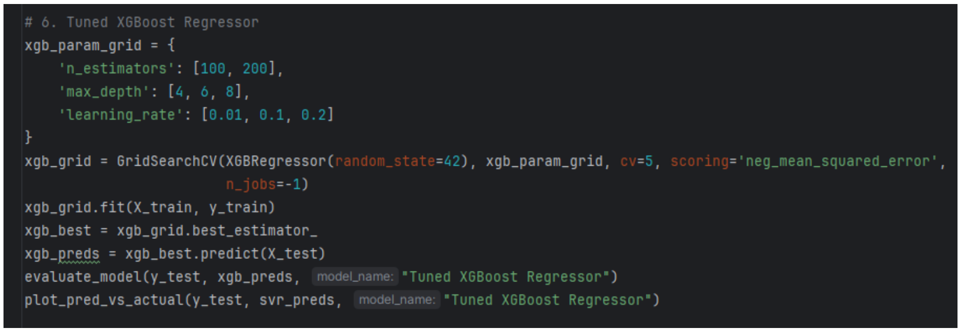
XGBoost outperformed both Linear Regression model and SVR model, demonstrating its capacity to capture complex interactions and minimize error. However, the XGBoost underperforms compared to the Random Forest model and the Tuned Support Vector Regressor.



```
# 4. XGBoost Regressor
# n_estimators: number of boosting rounds
# learning_rate: step size shrinkage used in update to prevent overfitting
xgb_model = XGBRegressor(n_estimators=100, learning_rate=0.1, max_depth=6, random_state=42)
xgb_model.fit(X_train, y_train)
xgb_preds = xgb_model.predict(X_test)
evaluate_model(y_test, xgb_preds, model_name: "XGBoost Regressor")
plot_pred_vs_actual(y_test, xgb_preds, model_name: "XGBoost Regressor")
```

6.3.6 Tuned XGBoost Regressor

Further tuning of XGBoost was performed using GridSearchCV over n_estimators, max_depth, and learning_rate. This yielded better results; the results were comparable to Random Forest and the Tuned SVR.



```
# 6. Tuned XGBoost Regressor
xgb_param_grid = {
    'n_estimators': [100, 200],
    'max_depth': [4, 6, 8],
    'learning_rate': [0.01, 0.1, 0.2]
}
xgb_grid = GridSearchCV(XGBRegressor(random_state=42), xgb_param_grid, cv=5, scoring='neg_mean_squared_error',
                        n_jobs=-1)
xgb_grid.fit(X_train, y_train)
xgb_best = xgb_grid.best_estimator_
xgb_preds = xgb_best.predict(X_test)
evaluate_model(y_test, xgb_preds, model_name: "Tuned XGBoost Regressor")
plot_pred_vs_actual(y_test, svr_preds, model_name: "Tuned XGBoost Regressor")
```

6.4. Model Evaluation

6.4.1 Performance Metrics

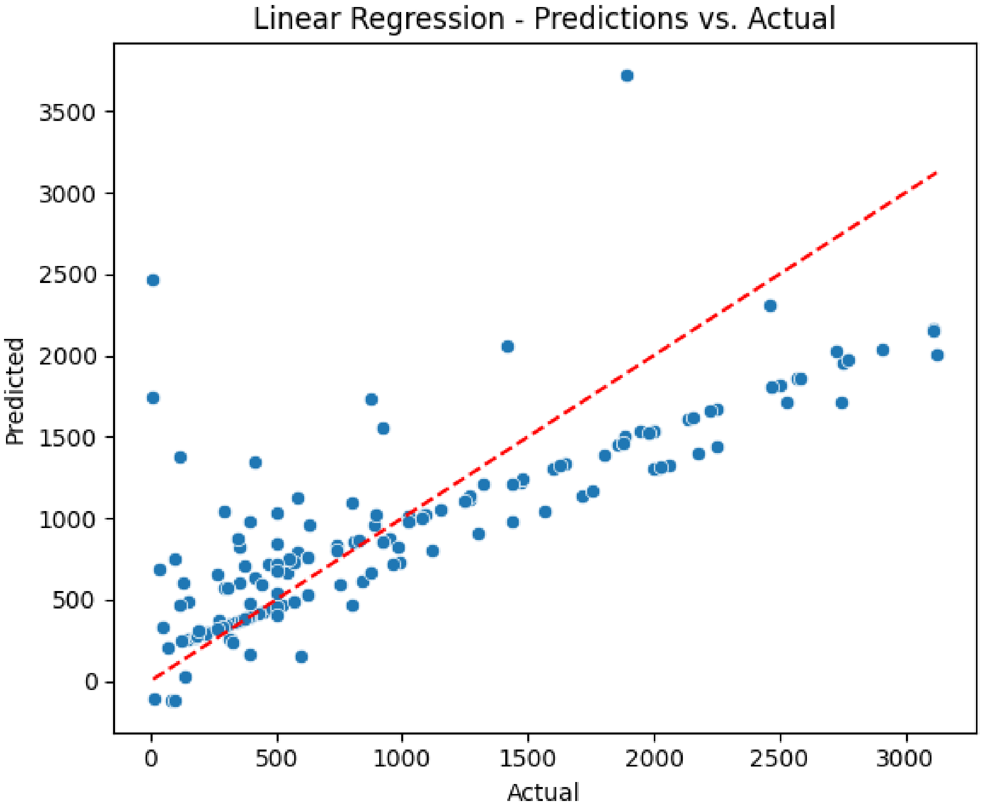
All models were evaluated using common regression metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and the Coefficient of Determination (R^2).

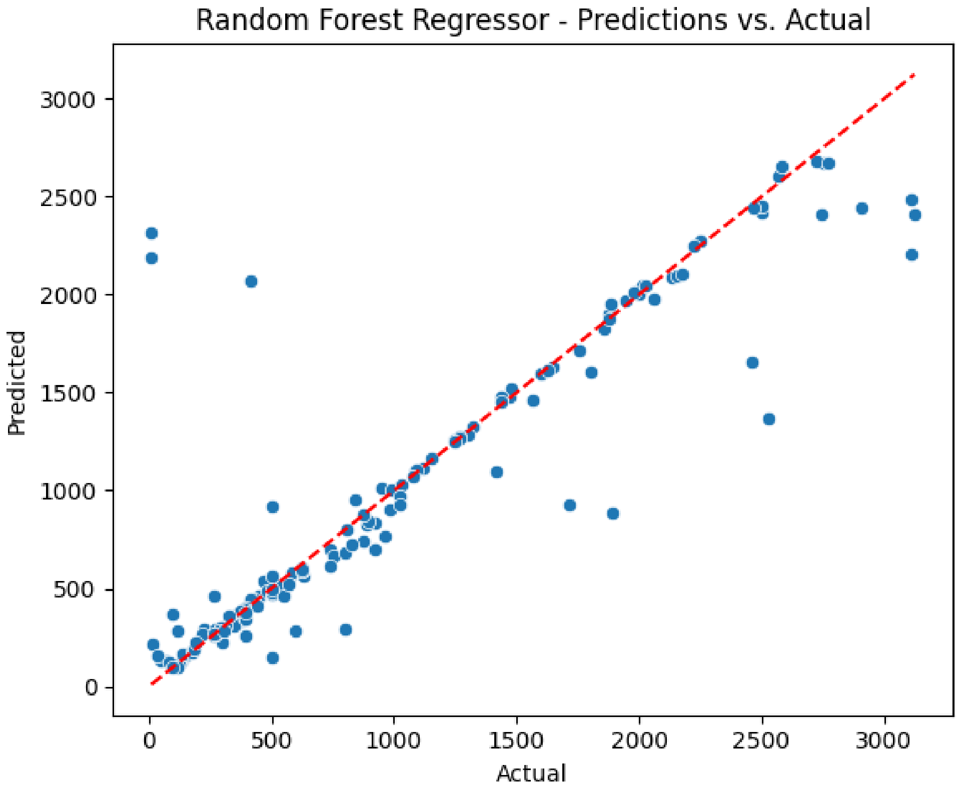
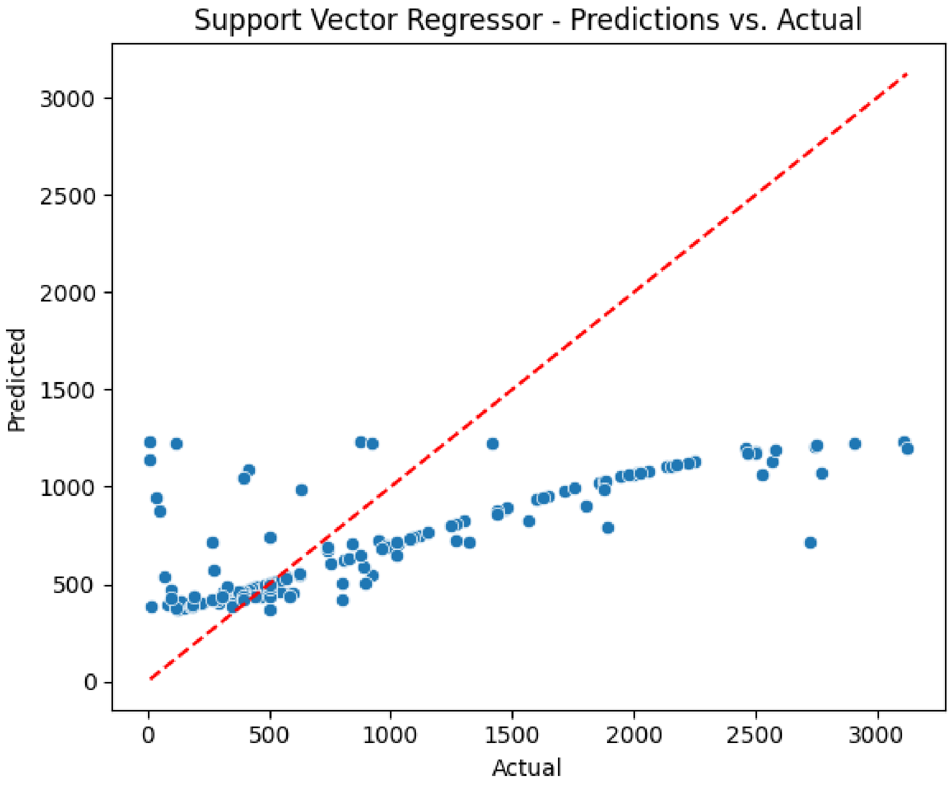
	Linear Regression	**Random Forest Regressor**	**Support Vector Regressor**	**Tuned Support Vector Regressor**	**XGBoost Regressor**	**Tuned XGBoost Regressor**
MAE	349.01	133.25	493.66	171.77	155.86	209.71
RMSE	503.83	355.38	688.79	368.69	416.32	381.51
R²	0.63	0.82	0.31	0.80	0.75	0.79

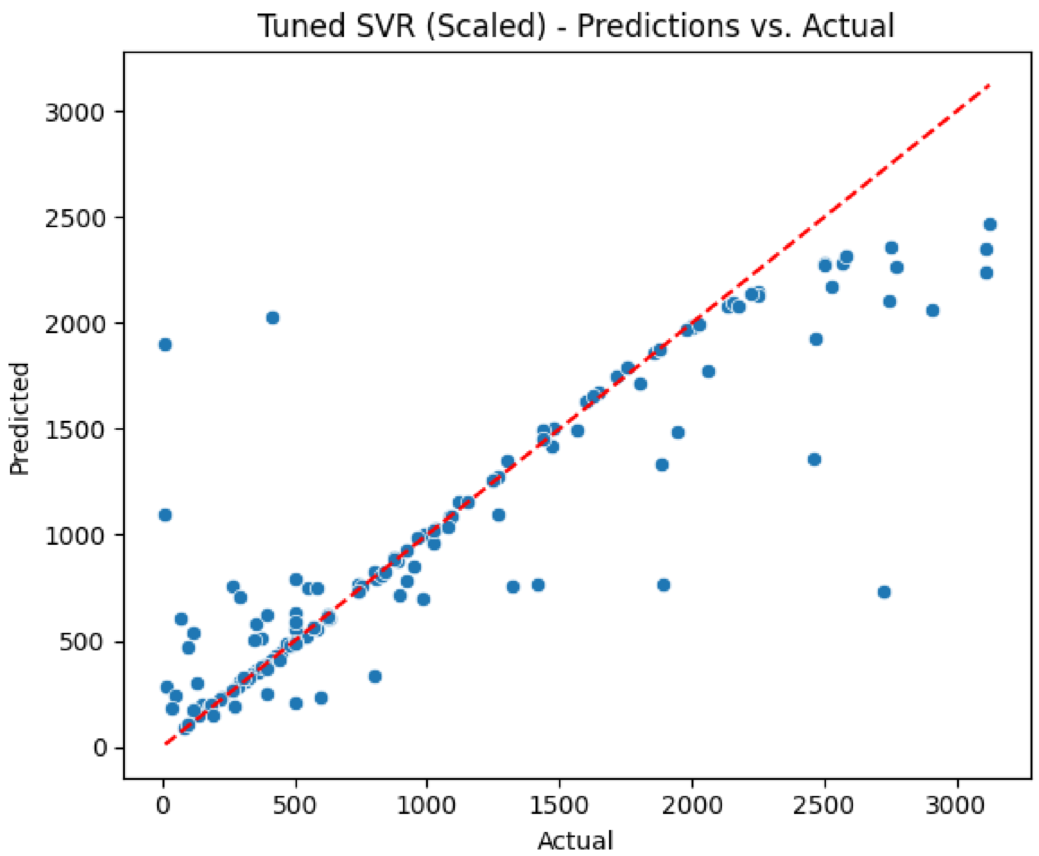
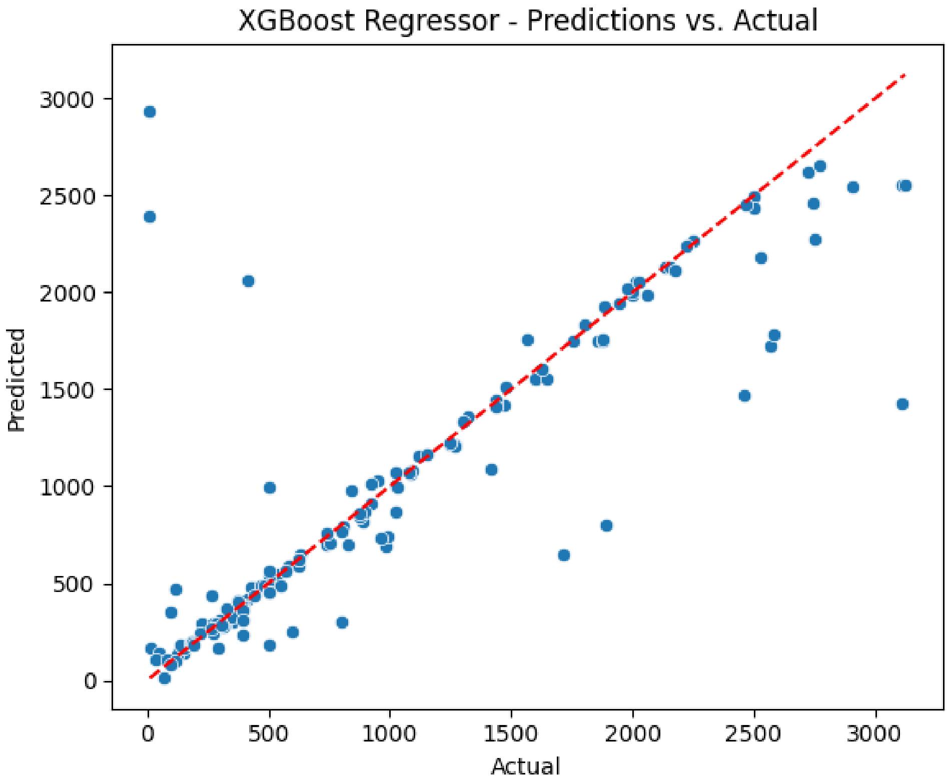
Among the regression models evaluated, the Random Forest Regressor demonstrated superior predictive performance across all key metrics. It achieved the lowest Mean Absolute Error (MAE) of 133.25 and Root Mean Squared Error (RMSE) of 355.38, along with the highest coefficient of determination (R^2) value of 0.82, indicating strong accuracy and generalizability. In contrast, the Linear Regression model, used as a baseline, yielded a higher MAE of 349.01, RMSE of 503.83, and a lower R^2 score of 0.63, reflecting a weaker fit and reduced predictive capability. The Support Vector Regressor (SVR) initially exhibited the poorest performance, with the highest MAE (493.66) and RMSE (688.79), and the lowest R^2 value (0.31). However, after hyperparameter tuning, the performance of SVR improved substantially, achieving an MAE of 171.77, RMSE of 368.69, and an R^2 of 0.80, thus approaching the performance level of the Random Forest model.

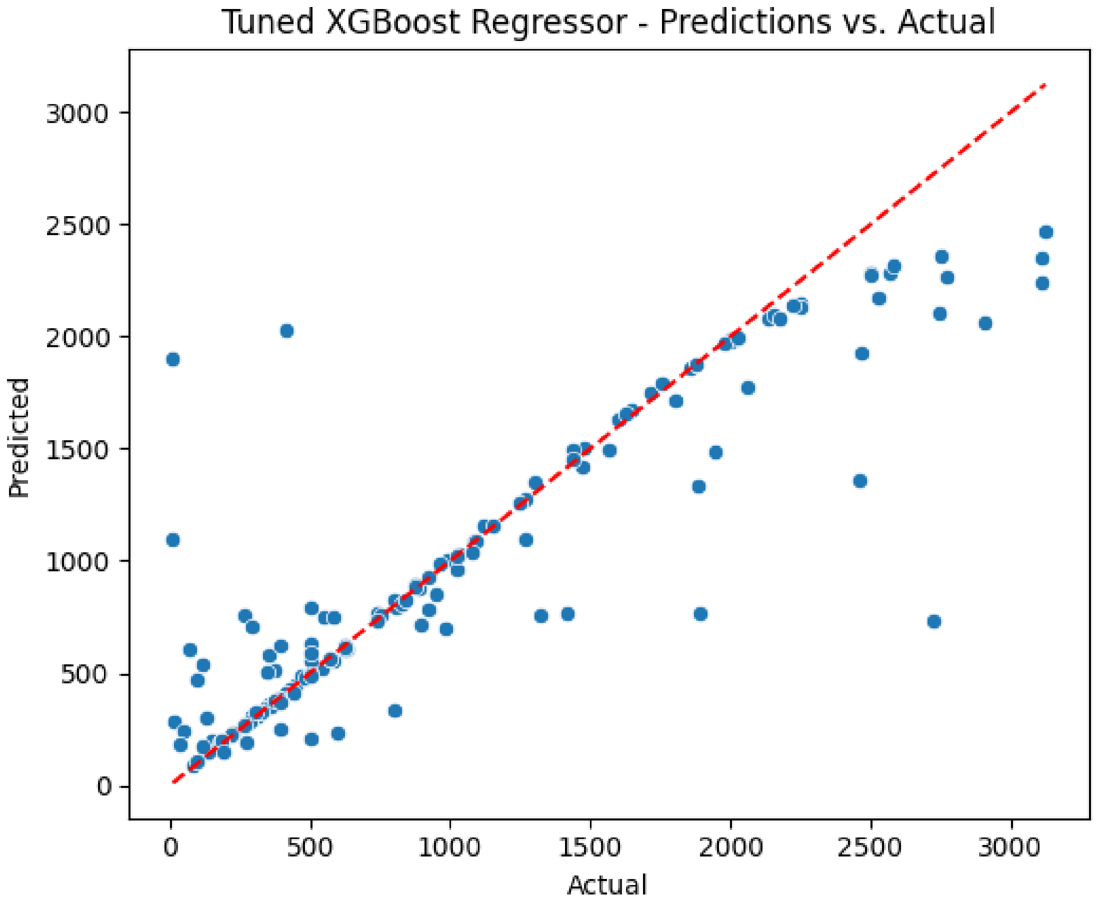
The XGBoost Regressor also showed competitive results, with an MAE of 155.86, RMSE of 416.32, and R² of 0.75. Following hyperparameter tuning, XGBoost demonstrated a modest improvement in RMSE (381.51) and R² (0.79), although the MAE increased to 209.71, suggesting that the tuning process had mixed effects on its overall error distribution. These findings underscore the effectiveness of ensemble tree-based methods, particularly Random Forest, in modeling the target variable with high accuracy. Moreover, the significant improvement observed in the tuned SVR highlights the critical role of model optimization. Overall, Random Forest and the tuned SVR emerged as the most robust models for this regression task, outperforming both linear and other non-linear alternatives.

6.4.2 Predicted Vs Actual plots

Model	
Linear Regression	

Random Forest Regressor	
Support Vector Regressor	

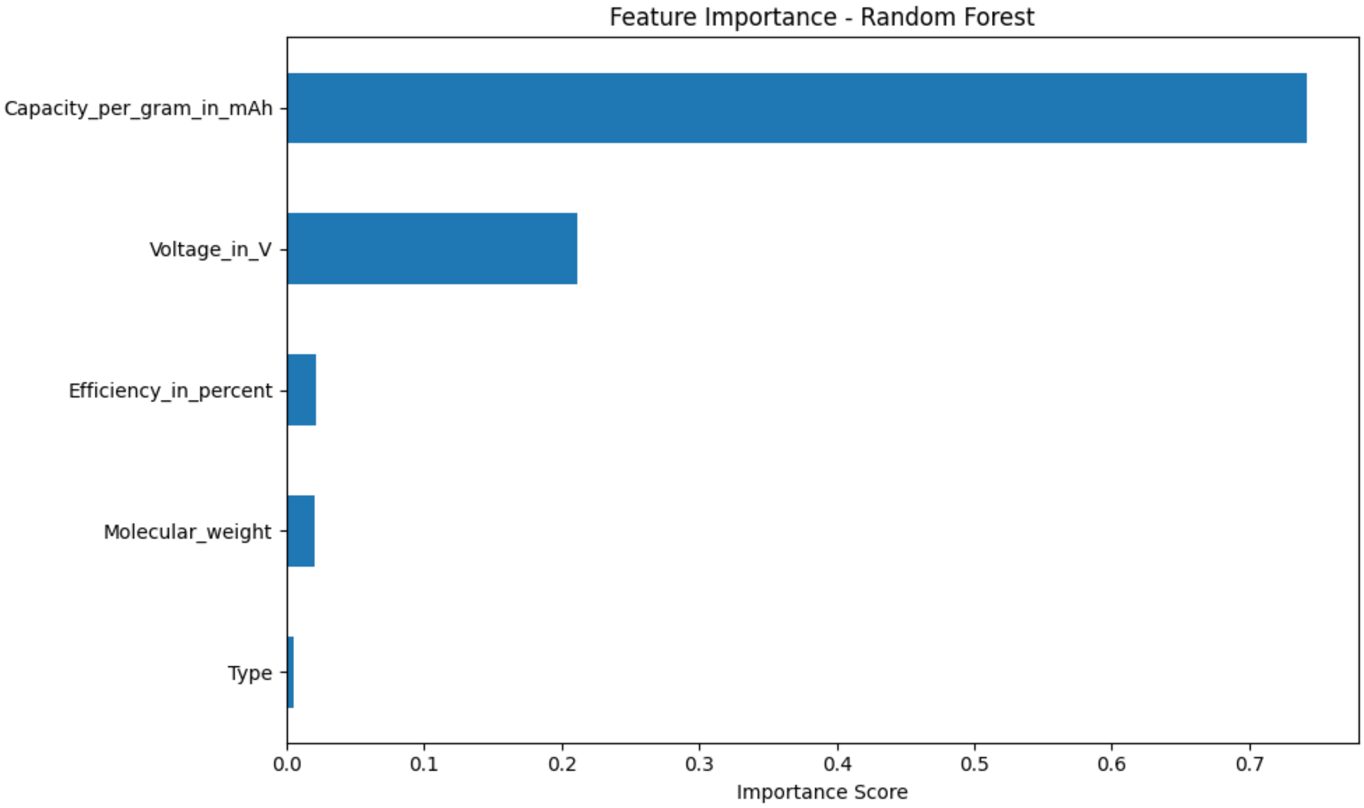
Tuned Support Vector Regressor	
XGBoost Regressor	

Tuned XGBoost Regressor	

The table for Predicted vs Actual plots visually represents how well the model has fit the data. It is very clear that the Random Forest model outperforms the other models, in terms minimizing the error between the Predicted and the Actual values of the test set.

6.5. Feature Importance

The trained Machine Learning algorithms were able to measure the relative importance of each feature in the dataset. Due to its performance metric being the best overall, going forward the Random Forest model is considered the best model. Hence, the feature importance resulting from the Random Forest model is further discussed aided by the following figure:



This figure highlighs how much each input variable contributes to the model’s predictive performance. Among the features evaluated, Capacity_per_gram_in_mAh stands out with a dominant importance score of approximately 0.74, indicating it is the most critical variable in the model. This suggests that the model relies heavily on this feature to make accurate predictions. The next most influential feature is Voltage_in_V, which has a significantly lower score of around 0.21, but still contributes meaningfully to the prediction outcome. The remaining features — Efficiency_in_percent, Molecular_weight, and Type — show very low importance scores, each contributing minimally to the model's performance. This implies that while these features are present in the dataset, they offer little additional predictive value compared to Capacity and Voltage. In practical terms, the Random Forest model could perform well even if these low-importance features were removed or deprioritized.

6.6 Conclusion

This chapter explored the application of supervised learning techniques to predict the energy density of battery materials. Among the six models tested, the **Random Forest Regressor** (with an R^2 value of 0.82) and **Tuned XGBoost Regressor** (with an R^2 value of 0.82) delivered the most accurate and robust predictions. The success of ensemble methods like Random Forst and XGBoost highlights the presence of complex, nonlinear relationships within the dataset.

The findings of this study reinforce the utility of machine learning in materials science and the predictive modeling of material properties. Future work could involve the integration of domain-specific feature engineering, model interpretability tools such as SHAP values, and expanding the dataset to include experimental uncertainties and real-world deployment constraints.

Chapter 7: Findings and Conclusions and Future work

This research aimed to predict the energy density of battery materials (in Wh/kg) using a curated dataset containing 832 unique compounds and their associated electrochemical properties. The dataset integrates both cathode and anode materials and includes key features such as specific capacity (mAh/g), average voltage (V), efficiency (%), and molecular weight.

7.1 Data Integrity and Preparation

After extensive preprocessing and cleaning prior to this merged dataset, only high-quality, distinct entries were retained. The removal of thousands of incomplete or redundant records allowed for a cleaner training foundation but likely reduced the available variance in rare or experimental compounds. Still, the remaining 832 entries maintained a rich diversity of anode and cathode types, enabling the modeling of realistic battery behaviors.

7.2 Feature Analysis and Target Construction

Energy density values in Wh/kg were precomputed using the formula:

$$\boldsymbol{Energy\ Density\ \left(\frac{Wh}{kg}\right) = Capacity\ \left(\frac{mAh}{g}\right) \times \frac{Voltage\ (V)}{1000}}$$

This aligns well with the physical principles of battery materials, where specific capacity and voltage are the two dominant physical drivers of energy performance. Exploratory analysis showed energy density values ranging widely—from below 100 Wh/kg to over 1700 Wh/kg—highlighting both low-performance and high-performance material candidates.

7.3 Modeling and Supervised Learning Results

Four predictive regression models were applied:

1. Linear Regression established a baseline, capturing overall trends but failing to accommodate non-linear effects, particularly in edge cases with very high or low energy densities.
2. Random Forest Regressor and XGBoost Regressor delivered strong predictive performance, capturing complex interactions among variables such as the interplay between capacity, voltage, and molecular weight.
3. Support Vector Regressor (SVR) showed promise but was more sensitive to feature scaling and parameter tuning.

Among these, Random Forest and XGBoost yielded the best R² score and lowest RMSE, accurately modeling the non-linear structure of the data without overfitting.

7.4 Feature Importance Insights

Consistent across tree-based models, Capacity_per_gram_in_mAh and Voltage_in_V were the top contributors to predicting energy density. Efficiency_in_percent played a lesser but still significant role, particularly for differentiating between theoretically high-performing materials and those with practical limitations. Molecular_weight, though not directly used in the energy density formula, showed marginal importance and may relate indirectly to energy-to-mass ratios.

7.5 Material Type Patterns

Grouping by the Type column revealed:

1. Cathode materials generally had lower capacity but higher voltage.
2. Anode materials often showed higher capacity but lower voltage.

This trade-off was accurately captured by the model, which learned these inverse trends as part of its predictive logic.

7.6 Conclusion

This dissertation has successfully demonstrated that machine learning models, especially ensemble methods, can predict the energy density of battery materials with considerable accuracy using a relatively compact and high-quality dataset. The study confirms that specific capacity and voltage are the dominant drivers of energy density, aligning with physical battery chemistry principles.

While the dataset's size limited the use of more complex deep learning architectures, the strong performance of Random Forest and XGBoost models highlights that meaningful predictions can be derived even from datasets under 1,000 entries—provided they are clean and feature-rich.

7.7 Future Work

7.7.1 Enrichment of Material Descriptors

This study relied primarily on empirical electrochemical properties such as specific capacity, average voltage, efficiency, and molecular weight to predict energy density. While these features are fundamental to battery performance, they do not capture the full complexity of the materials themselves. Future work could enhance predictive power by incorporating additional descriptors, particularly those related to the atomic and structural characteristics of the compounds. Features such as crystal structure (space group, lattice parameters), elemental composition, formation energy, and electronic band gap can provide deeper insights into material behavior. These descriptors are accessible through databases such as the Materials Project, AFLOW, and the Open Quantum Materials Database (OQMD). Including such information could improve model generalizability and enable predictions for newly designed materials that have not yet been experimentally characterized.

7.7.2 Advanced Feature Engineering and Dimensionality Expansion

To better model the intricate relationships between features and energy density, future research should consider more sophisticated feature engineering. This may include the generation of interaction terms, polynomial features, or chemically informed descriptors such as electronegativity difference, atomic radii, and valence electron counts. Additionally, modern vector embedding techniques like mat2vec or Atom2Vec, which capture latent patterns from chemical text data or material graphs, could be used to numerically represent compound composition in a format more suitable for machine learning. These techniques may be coupled with dimensionality reduction methods like PCA, t-SNE, or UMAP to extract latent structures in the data, enhance model interpretability, and facilitate clustering or classification of material families.

7.7.3 Incorporating Uncertainty into Predictions

The input features used in this study—particularly capacity and voltage—are subject to experimental variability. Consequently, future predictive models could benefit from incorporating uncertainty quantification. Techniques such as Bayesian regression, Gaussian Process Regression (GPR), and quantile regression can be employed to provide probabilistic predictions along with confidence intervals. This would be particularly useful in scenarios where predictions are used to guide high-cost or high-risk experimental synthesis efforts. By accounting for confidence or uncertainty in predictions, such models can aid in more informed decision-making and risk assessment.

7.7.4 Model Transferability and Domain Adaptation

As the volume of battery materials data continues to grow, future models could explore transfer learning strategies to improve generalization. Instead of training models from scratch for every new subset of data, pre-trained models could be fine-tuned on specialized material classes such as solid-state electrolytes, organic cathodes, or sodium-ion materials. This would allow the model to retain knowledge of general material behavior while adapting to the unique characteristics of a narrower domain. Such domain adaptation would not only reduce training time but also enhance performance on tasks where labeled data is scarce.

7.7.5 Experimental Validation of Model Predictions

While the predictive models in this study were validated using standard evaluation metrics such as R^2 and RMSE, their true value lies in their ability to inform experimental research. Future efforts should prioritize validating predictions through laboratory synthesis and electrochemical testing. Collaborations with experimental battery researchers or comparisons with published empirical studies would offer a feedback loop for assessing model reliability and refining input data. This experimental grounding would strengthen the model's credibility and facilitate its integration into practical materials discovery workflows.

7.7.6 Exploration of Advanced Machine Learning Architectures

Although tree-based models such as Random Forest and XGBoost demonstrated strong performance in this study, future work could investigate more complex machine learning architectures. Deep learning techniques, particularly Graph Neural Networks (GNNs), are well-suited for materials science applications as they can represent molecules or crystalline solids as graphs of atoms and bonds. These models can capture spatial and structural relationships that traditional models may miss. AutoML frameworks, which automate the process of model selection and hyperparameter tuning, could also be employed to systematically identify optimal models. Additionally, explainable AI techniques such as SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-Agnostic Explanations) can be integrated to better understand the decision-making process of black-box models and ensure scientific interpretability.

By pursuing these future directions, the scope and impact of machine learning applications in battery materials research can be significantly expanded. These enhancements would not only improve predictive accuracy but also increase the scientific transparency and practical usability of models in real-world discovery and development settings.

Appendix

References

Codebase (Public GitHub Repository):

https://github.com/mahfuzahmed/Predicting_Energy_Density_of_Battery_Material

Data Source

Huang, S. and Cole, J. (2020). A database of battery materials auto-generated using ChemDataExtractor. figshare. [online] doi: https://figshare.com/articles/dataset/A_database_of_battery_materials_auto-generated_using_ChemDataExtractor/11888115/2

Other Sources

1. QuantumScape (n.d.) Energy density: the basics. Available at: https://www.quantumscape.com/resources/blog/energy-density-the-basics/ (Accessed: 5 June 2025).
2. XINNENERGY (2024) Energy density in batteries: an overview. The Innovation. Available at: https://www.theinnovation.org/data/article/energy/preview/pdf/XINNENERGY-2024-0002.pdf (Accessed: 5 June 2025).
3. Thunder Said Energy (n.d.) Lithium-ion batteries: energy density. Available at: https://thundersaidenergy.com/downloads/lithium-ion-batteries-energy-density/ (Accessed: 5 June 2025).
4. Mao, Y. et al. (2021) ‘A data-driven perspective of battery materials’, Materials Today Energy, 21, 100734. Available at: https://www.sciencedirect.com/science/article/pii/S2666546821000355 (Accessed: 5 June 2025).
5. Hassini, M., Redondo-Iglesias, E. and Venet, P. (2023). Lithium–Ion Battery Data: From Production to Prediction. Batteries, [online] 9(7), p.385. doi: https://doi.org/10.3390/batteries9070385
6. Zhong, M., Guan, J., Sun, J., Shu, X., Ding, H., Chen, L., Zhou, N. and Xiao, Z. (2021). A Cost- and Energy Density-Competitive Lithium-Sulfur Battery. Energy Storage Materials, 41, pp.588–598. doi: https://doi.org/10.1016/j.ensm.2021.06.037
7. Géron, A. (2019) *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 2nd edn. Sebastopol, CA: O’Reilly Media.
8. Kuhn, M. and Johnson, K. (2013) *Applied Predictive Modeling*. New York: Springer.
9. Hastie, T., Tibshirani, R. and Friedman, J. (2009) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd edn. New York: Springer.
10. Jain, A. et al. (2013) ‘The Materials Project: A materials genome approach to accelerating materials innovation’, *APL Materials*, 1(1), p. 011002. doi:10.1063/1.4812323.
11. Breiman, L. (2001) ‘Random Forests’, *Machine Learning*, 45(1), pp. 5–32.
12. Cortes, C. and Vapnik, V. (1995) ‘Support-vector networks’, *Machine Learning*, 20(3), pp. 273–297.

13. Chen, T. and Guestrin, C. (2016) ‘XGBoost: A scalable tree boosting system’, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD ’16)*, San Francisco, CA, 13–17 August. New York: ACM, pp. 785–794. doi:10.1145/2939672.2939785.
14. Chen, M., Ma, X., Chen, B., Arsenault, R., Karlson, P., Simon, N. and Wang, Y. (2019). Recycling End-of-Life Electric Vehicle Lithium-Ion Batteries. Joule, 3(11), pp.2622–2646. doi: https://doi.org/10.1016/j.joule.2019.09.014
15. Huang, S. and Cole, J.M. (2020). A database of battery materials auto-generated using ChemDataExtractor. Scientific Data, 7(1). doi: https://doi.org/10.1038/s41597-020-00602-2