UNIVERSITY OF KWAZULU-NATAL™

INYUVESI YAKWAZULU-NATALI

DEVELOPMENT OF A MATHEMATICAL MODEL TO ENABLE OPTIMAL EFFICIENCY OF THE INDABUKO LITHIUM-ION BATTERY

STUDENT: JOHANNES MPHAKA
STUDENT NUMBER: 217081527
SUPERVISOR: PROF N. CHETTY (UKZN)
CO-SUPERVISOR: PROF R.R. MAPHANGA (CSIR)
INSTITUTION: UNIVERSITY OF KWA-ZULU NATAL (PMB CAMPUS)

Abstract

Cathode materials are the foremost primary challenge for the vast scale application of lithium-ion batteries in electric vehicles and the stockpiles of power. Foreseeing the properties of cathode materials is one of the central issues in energy storage. In the recent past, density functional theory (DFT) calculations aimed at materials property predictions offered the best trade-off between computational cost and accuracy compared to experiments. However, these calculations are still excessive and costly, limiting the acceleration of new materials discovery. Now the results from different computational materials science codes are made available in databases, which permit quick inquiry and screening of various materials by their properties. Such gigantic materials databases allow a dominant data-driven methodology in materials discovery, which should quicken advancements in the field. This study was aimed at applying machine learning algorithms on existing computations to make precise predictions of physical properties. Thus, the dissertation primary goal was build best ML models that are capable of predicting DFT calculated properties such as, formation energy, energy band-gap and classify materials as stable or unstable based on their thermodynamic stability. It was established that the algorithms only require the chemical formula as input when predicting materials properties. The theoretical aspect of this work describes the current machine learning algorithms and presents "descriptors"-representations of materials in a dataset that plays a significant role in prediction accuracy. Also, the dissertation examined how various descriptors and algorithms influence learning model. The Catboost Regressor was found to be the best algorithm for determining all the properties that were selected in this study. Results indicated that with appropriate descriptors and ML algorithms it is feasible to foresee formation energy with coefficient of determination (R^2) of 0.95, mean absolute error (MAE) of 0.11 eV and classify materials into stable and unstable with 86% of accuracy and area under the ROC Curve (AUC) of 89%. Lastly, we build a web application that allow users to predict material properties easily.

Thesis Declaration

The work described in this dissertation was carried out in the School of Chemistry and Physics, University of KwaZulu-Natal, Pietermaritzburg campus, under the co-supervision of Prof Naven Chetty.

These studies represent original work by the author and have not otherwise been submitted by any candidate for any degree.

Signed [redacted]......... Johannes Mphaka (Candidate)

Signed.................................... Prof N Chetty (Supervisor)

Signed......[redacted].................. Prof RR Maphanga (Co-supervisor)

Acknowledgements

I want to convey my heartfelt thanks to the aforementioned individuals and organisations:

- Thanks to my supervisors, Prof. Regina Maphanga, Prof. Naven Chetty and Dr. Sphumelelo Ndlovu for always assisting me with this project.
- Thanks to Material Project for data used for this project.
- Thanks to indabuko for funding this project.
- And lastly, thanks to my family and friends for supporting me in this project.

Contents

1 Introduction **7**

1.1 Introduction 7

1.2 Problem Background of Lithium Batteries 8

1.2.1 Structural Stability 11

1.2.1.1 Cohesive Energy 11

1.2.1.2 Formation Energy 12

1.2.1.3 Gibbs Free Energy 13

1.2.1.4 Phonon Frequency 13

1.2.2 Band Structures 14

1.3 Literature Review 15

1.3.1 Machine Learning in Materials Science 15

1.3.2 Materials Property Prediction 16

1.3.3 Materials Design and Discovery 16

1.3.4 Crystal Structure Prediction 17

1.3.5 Materials Processing and Complex Materials Behaviour 18

1.4 Research Goal 18

1.5 Dissertation Structure 19

2 Theoretical Background **20**

2.1 Machine learning 20

2.2 Machine Learning Approaches 21

2.2.1 Supervised Learning 21

2.2.2 Unsupervised Learning 22

2.2.3 Semi-Supervised Learning 22

2.3 Machine Learning Algorithms 22

2.3.1 Input Features Development 22

2.3.1.1 Procedure for developing features using chemical formula 23

2.3.2 Decision Tress and Random Forest 23

2.3.3 Extremely Randomized Trees 25

2.3.4 Gradient Boosting 25

2.3.5 Light Gradient Boosting Machine 26
2.3.6 Feature Selection 26
2.3.7 Feature Construction and Transformation . . 28
2.3.8 Hyperparameter Tuning 28
2.4 Machine Learning Process 29

3 Model Development 31
3.1 Workflow 31
3.2 Dataset 32
3.2.1 Data collection 33
3.2.2 Dataset Split 33
3.2.3 Cross Validation 34
3.3 Feature Vector 35
3.4 Model Selection 37
3.5 Feature Importance/Feature Selection 38
3.6 Model Tuning 39
3.7 Results and Discussions 40
3.8 Web-Tool For Formation Energy Prediction 41

4 Thermodynamic Stability 43
4.1 Thermodynamic Stability Classification 43
4.2 Data Collection and Data Preparation 43
4.2.1 Dataset Split 44
4.3 Model Selection 45
4.4 Feature Importance 46
4.5 Model Tuning 46
4.6 Results and Discussions 47
4.7 Band Gap Prediction 50
4.7.1 Results and Discussions 50

5 Conclusion 52
5.1 Conclusion 52
5.2 Future Studies 52

A Developed Web Application 54

B Code Details 56

Bibliography 69

List of Figures

1.1 Cylindrical lithium-ion cell and pouch cell. 9
1.2 Lithium-ion battery materials 11
1.3 a) Temperature-dependent and b) pressure-dependent free energy curves (relative to C_2) of different Na_2FeSiO_4 structures. [1] 13
1.4 Phonon dispersion spectra of PmmaXO monolayers: a) PmmaCO, b) PmmaSiO, c) PmmaGeO, and d) PmmaSnO [2] 14

2.1 Primary approaches to machine learning. 21
2.2 Schematic representation of decision tree 24
2.3 Schematic representation of level-wise tree growth 26
2.4 Schematic representation of level-wise tree growth 27
2.5 Machine learning steps. 29

3.1 ML Workflow. 31
3.2 Dataset for selected lithium containing materials. 33
3.3 Dataset split. 34
3.4 Models comparison. 34
3.5 Composition class calculator. 36
3.6 Pre-proccessed features used in the ML model. 37
3.8 Important features selected by catboost model 38
3.9 Performance of catboost model in training set (left) and in testing set (right) 40

4.1 Part of materials we selected to train our algorithms. 44
4.2 Label distribution. Here we have 7324 unstable and 1616 stable compounds. 44
4.4 The best 18 features and their values from the xgboost model classifier. 46
4.5 Area under the curve (AUC) 49
4.6 Performance of Catboost model in training set (left) and in testing set (right) 51

A.1 Software tool view page 54
A.2 Software tool view page 55

B.1 Dataset collection 56

B.2 Feature vector calculations 57
B.3 Preprocesed data 57
B.4 Models performance 58
B.5 Model build 58
B.6 Feature importance 59
B.7 Performance of catboost model in training set (left) and in testing set (right) 59
B.8 Model performance on test data 60
B.9 Model predictions on the training data 60
B.10 Model performance 61
B.11 Model tuning 61
B.12 Model results dataframe 61
B.13 Model performance in terms of R^2 62
B.14 Model performance in terms of MSE 62

List of Tables

3.1 Tuned model parameters from training set. 39

3.2 Materials DFT and predicted formation energies using ML web-tool we developed. The material is more stable when, $E_f > 0$. 42

4.1 Tuned model parameters from training set. 47

4.2 TP, FP, FN, and TN mean true positive, false positive, false negative, true negative, respectively. 47

4.3 TP, FP, FN, and TN mean true positive, false positive, false negative, true negative, respectively. 48

Chapter 1

Introduction

1.1 Introduction

One of the most challenging global threats in the 21st century is to tackle global warming while expanding the availability of energy to all. Inefficient energy production and storage, through burning fossil fuels or using batteries containing heavy metals, is a significant cause of pollution, producing millions of tonnes of greenhouse gases and toxic waste while also depleting limited precious resources. In recent years, intense research aims to improve effective, reliable, and secure electrical energy storage technology since the energy demand is growing worldwide. Essentially, the exploration of innovative and high functional materials is key to all technical advances, including the implementation of improved conversion devices and energy storage [3].

The intensive search is underway to design and explore new materials with enhanced and optimized properties. However, this has proved challenging because of materials microstructure complexity and preparation, dependent on many materials. On the other hand, compared to the traditional methods of exploration that are complicated, costly, labor intensive, and time-consuming, the challenge offers the ability to create, manufacture, and deploy materials as quickly as possible [4]. Computational materials modelling are becoming the prime motivator for the production and discovery of new materials. Computational modeling practices in materials science (MS) have progressed from creating models to exploration and conception of new materials based on previous modeling findings, machine learning (ML), and data mining approaches. The design of materials based on simulations is anticipated to contribute to new materials discovery, accelerate the production of new content in products, and minimize the time and expense of material development [5, 6].

Experiments and computer simulations are traditional techniques commonly employed in the design and production of materials. Combined computational modeling and experiments have significantly decreased the design time and cost. It is complicated to use the two approaches to speed up material development and design because of their intrinsic constraints, such as differing theoretical basis and experimental conditions. Several efforts have been made, including recent advances in the Materials Genome Initiative promoting data-intensive and machine learning approaches in material science to develop ways to address such deficiencies. ML techniques can detect complicated correlations between the structures and the diverse chemical and physical properties of materials, which is difficult to detect using traditional mathematical models. Practical ML application in solving complex regression and classification problems for MS is not constrained to; physical properties prediction, crystal structure prediction [7], classification of silicate-based cathode [8], electronic properties [9] semiconductors [10], and alloys [11].

Novel and affordable battery materials with higher energy capacity and power density, improved stability, long life cycle, and environmentally safe are urgently needed. Generally, laboratory experiments for synthesizing and identifying functional materials require substantial time, while conventional computational modeling techniques such as density functional theory differ significantly. Lithium-ion batteries have been used in the previous decades as the primary energy storage system. However, the low availability of lithium, cobalt, and related expenses of these materials led to a concerted attempt to create better lithium-ion batteries with optimized properties. In light of issues related to cost and the potential supply of lithium, the emphasis on designing new materials for positive and negative electrodes has significantly grown in the area, which can improve rate capacity, energy density, and cycling stability. As an emerging field, machine learning offers immense potential for discovering alternative lithium-ion battery materials due to its versatility in handling difficult practical problems. It provides information for computers to learn and the requirements for analyzing algorithms. The forecasting is entirely based on mathematical statistics and physical or chemical features of the material [12, 13].

1.2 Problem Background of Lithium Batteries

Majority of current technological developments are increasingly reliant on the ability to locally store energy through the use of batteries; e.g. grid-scale implementation of renewable energy systems, worldwide proliferation of mobile devices and revolution in hybrid and fully electric vehicles. However, battery performance is the largest inhibitor in innovation, particularly lack of improvement in lithium-ion battery performance. The efficiency of LIBs has increased by only 27 percentage over the the past 20-30 years. LIBs efficiency can be improved by enhancing energy density, durability and power retention of cathode material properties. [14]. Lithium-ion batteries are multifunctonal devices with both scientific and technological difficulties for each segment as shown in Figure 1.1.

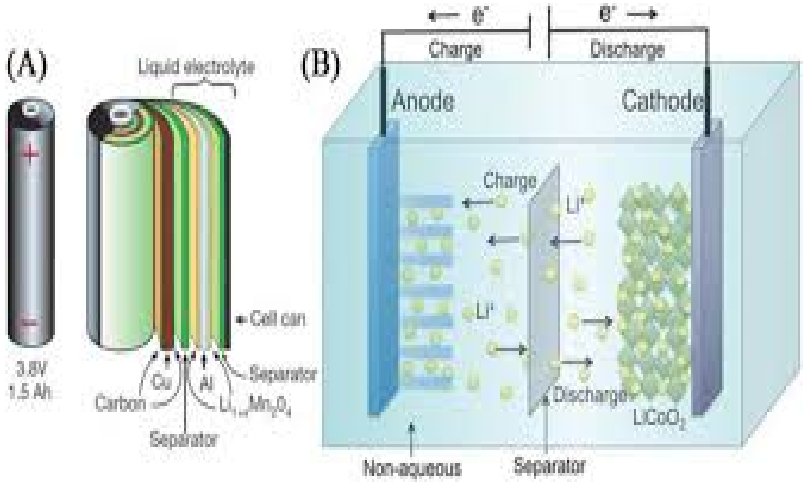


Figure 1.1: Cylindrical lithium-ion cell and pouch cell.

A standard LIB is made up of four main segments [15, 16]:

i. **Anode** - is a negative or reduction electrode which loses and oxidizes electrons to the external circuit during electrochemical reaction. Anode is painted on the Cu foil current collector.

ii. **Cathode** - is a positive electrode that absorbs electrons from an external circuit and is reduced during an electrochemical reaction. Cathode is painted on the Al foil current collector.

iii. **Separator** - is a thin sheet material that isolate cathode and anode electrodes which allows ions to be transported between cathode and anode.

iv. **Electrolyte** - electrolytes are organic solvents with dissolved Li salts. example of this solvents are:

   - Diethyl carbonate (DEC), ethylene carbonate (EC), and propylene carbonate (PC) [17].
   - Li salts: lithium hexafluorophosphate ($LiPF_6$) and lithium perchlorate ($LiClO_4$) [17].

The graphite anode LIB and the $LiCoO_2$ cathode demonstrate the fundamental principle. The intercalation/de-intercalation cycle during the charging/discharge process is shown in Figure 1.1. During charging in LIB, Li+ travels through the conductive electrolyte from the cathode ($LiCoO_2$) into the anode and bind it with porous graphite. One significant factors in the success of these market LIBs is that, electrode materials expect small volume changes during the charging/discharge process, compared with high Li content compounds. LIBs have been among the most successful energy storage technologies for a variety of purposes, spanning from mobile devices to transport [16]. However, LIBs remain trapped with safety issues: Boeing was forced to burn its whole fleet after an LIB pack found burning itself and millions of dollars and reports of a smoldering battery fire in Tesla model S, which destroyed Tesla Motor's stock by 6 percent [18]. LIBs have safety concerns, maily caused by the liquid electrolyte, a toxic and flammable chemical solvent in most marketed LIBs [19,20]. Recent studies showed that replacing the flammable liquid electrolyte in modern batteries with stable solid-state batteries, the safety lifetime and energy density of the batteries would improve significantly [21]. However, solid batteries are complicated systems and it has proved difficult to find new electrolytes that meet the many different requirements of battery life. Essentially, candidate material should have high lithium conductivity, flexible chemical and phase stability, a large window of electric stability, high electronic conductivity and be affordable [22,23].

A major component that restricts the efficiency of the batteries is the active element of the positive electrode (cathode) and is also the most costly component of LIB. Since 1980 to date, a consistent effort was made initially by Goodenough to suggest and research transition-metal (TM)-based oxide compounds with an emphasis on those compounds that crystallize into structures that support high mobility of Li+ ions, in order to acquire energy during redox reactions. The achievements for the $LiCoO_2$ system (layered) were developed in 1980, followed by $LiMn_2O_4$ system (spinel) in 1986 and $LiMPO_4$ family (olivine) in 1997. Consequently, the LIBs industries were rapidly expanded using these materials. Layered materials are utilized as cathodes for high-energy systems, whereas spinel oxides and olivines are preferred for high-power LIBs because of minimal cost and long-life specifications [24]. These lithium insertion materials must have special properties such as chemical, structural, good thermal stability, high specific capacity, rate capability, low electronic conductivity, toxicity, cost and be safety. LIBs are generally of high cost owing to their use of transition metals such as manganese, nickel and cobalt. Figure 1.2 shows various groups of Li-ion battery materials and their respective voltages.

Furthermore, dependence on natural electrolytes has caused combustibility and security concerns upon dendrite development. Another challenge is that there are no high capacity anodes. Hence, each segment of the battery would benefit from novel materials discovery and design. The next subsections discuss selected properties related to this study, due to their importance in battery performance [25].

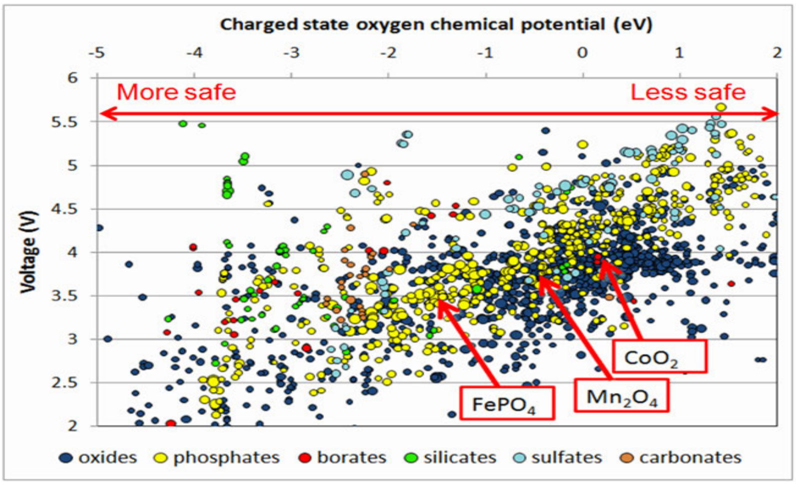


Figure 1.2: Lithium-ion battery materials

1.2.1 Structural Stability

The stability of a LIB material determines the life-time of the battery. The material stability can be determined by calculating any of these physical quantities and are explained briefly in the next subsections [26, 27]:

i. Cohesive energy

ii. Formation energy

iii. Gibbs free energy

iv. Phonon dispersion spectrum

1.2.1.1 Cohesive Energy

When compounds are created from free isolated atoms, they generate an energy called cohesive energy and is calculated analytical using the following expression: [27].

$$E_{\text{co}} = \frac{a \times E(X) + b \times E(Y) - E\left(X_a Y_b\right)}{a+b} \tag{1.1}$$

where E_{CO} represents cohesive energy, a is the amount of element X and b is the amount of element Y and X_aY_b is a chemical compound, $E(X)$ is the energy of X_a and $E(Y)$ is the energy of Y_b, $E(X_aY_b)$ is the energy of chemical compound X_aY_b. When cohesive energy is higher, the structure of of the material is more stable. The covalent organic frameworks (COFs) report has recently published calculated cohesive energy of the materials. COFs are LIBs anode materials with high porosity, covalent bonds and low density. The stability of the material structure is a critical factor in LIBs. In this context, [28] used equation (1.1) to calculate the cohesive energies of COF material, NUS2 ($[C_9H_6O_3N_3]$n, a copolymer of hydrazine hydrate and tri-formyl-phloroglucinol, and its complexes with some lithium atoms). The calculated cohesive energy of NUS2 were found to be (5.6 eV/atoms), which correlates to the good structural stability of NUS2. Furthermore, it was noted that the 14 lithium atoms with NUS2 complex already have a reasonable high cohesive energy (4.6 eV/atom), suggesting that NUS2 is a promising LIB material with high lithium storage capacity and thermodynamic stability.

1.2.1.2 Formation Energy

Materials that give both electrodes and electrolytes the necessary electrochemical stability serve as an interlayer that prevents unplanned reactions and improves battery cycling. The Li-ion conductivity of oxides electrolytes (10^{-3} S/cm at room temperature) could also be minimized by this chemical stability. A broad electrochemical stability window is a further prerequisite for stable electrolytes. Electrolytes with low oxidation stability might bind to the cathode and create extreme resistance interfacial layers that reduce ionic flow. Formation energy describes the energy transition when a compound is produced in its normal state from its constituent elements. The equation below is used to calculate formation energy of the material:

$$E_{\mathrm{f}} = \frac{a \times E(X) + b \times E(Y) - E\left(X_a Y_b\right)}{a+b} \tag{1.2}$$

where E_f represents the formation energy, a is the amount of element X and b is the amount of element Y and X_aY_b is a chemical compound, $E(X)$ and $E(Y)$ is the energy of X_a and Y_b in their normal states rather than isolated atoms, $E(X_aY_b)$ is the energy of chemical compound X_aY_b. The higher the formation energy, the more stable is the the structure of of the material. It is generally advisable to determine the structural stability from formation energy than cohesive energy since a compound is rarely produced from individual atoms. It is known that Si and Ge have a low stability and high specific capacity. Due to the demand of high specific capacity battery material, the two-dimensional (2D) materials are recently explored as materials for lithium-ion batteries. Hence, SiGe 2D alloy was recently proposed and using equation (1.2), formation energy of SiGe was recently calculated to be 1.51 eV, indicating a good thermodynamic stability [27].

1.2.1.3 Gibbs Free Energy

The Gibbs free energy of a system is calculated using the following analytical equation:

$$\begin{gathered} G = H - TS \\ H = E + PV \end{gathered} \tag{1.3}$$

where G represents Gibbs free energy, H is the enthalpy, S is the entropy, T represents the temperature of the system, E is the internal energy, P is the pressure of the system and V is the volume of the system. Gibbs free energy is utilized to evaluate isomers or polymorphous stability at various temperatures or pressures. For example, Na_2FeSiO_4 cathode material has been explored at several different stages using computational modelling method. Due to orthosilicate polymorphism, it is difficult to deduce such information experimentally Na_2FeSiO_4. Fourteen structure models were configured and Gibbs free energies were calculated under various temperatures and pressures [1] and are shown in Figure 1.3. It was found that Na_2FeSiO_4 P_n phase has the lowest Gibbs free energy at low temperatures and pressures. The system converted to P_n at 700 ° C and to $Pca2_1$ at a pressure of 8 GPa [27].

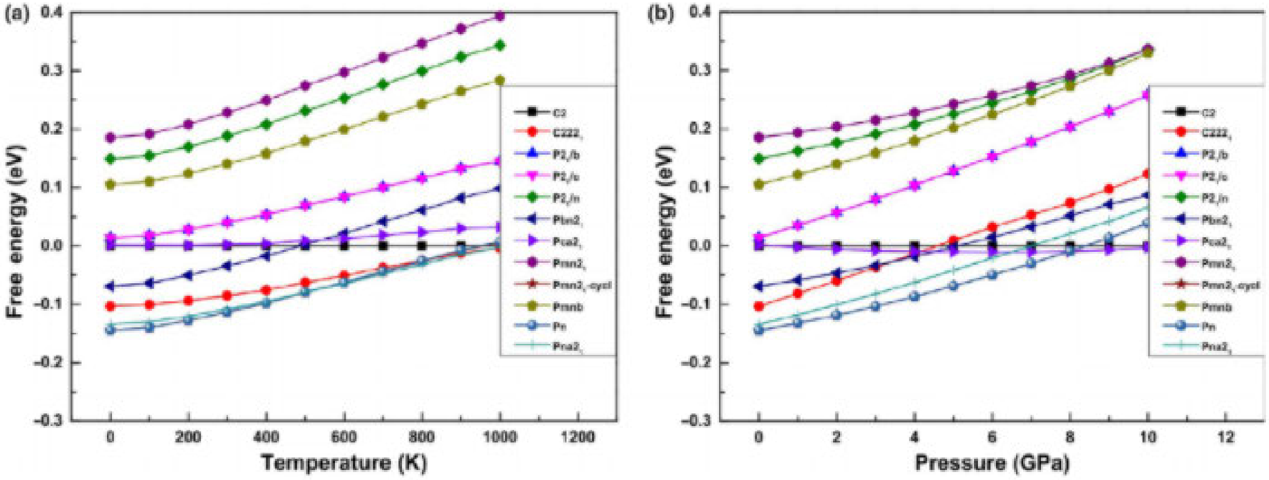


Figure 1.3: a) Temperature-dependent and b) pressure-dependent free energy curves (relative to C_2) of different Na_2FeSiO_4 structures. [1] .

1.2.1.4 Phonon Frequency

The spectrum of phonons shows the atoms vibrations mode of all atoms. When a computation cell contains m atoms, the counts of acoustic branches is 3 and optical branches is 3m^{-3}. The acoustic branch denotes the original cell's vibration, while similar vibration of the atoms within the cell defines the optical branch. An imaginary frequency from the measurement of the spectrum of the phonons indicates the instability of a structure of the material. The structural stability of monolayer structures utilizing phonon dispersion spectra of Pmma-XO,[X C, Si, Ge and Sn] is shown in Figure 1.4:

These materials are potential candidates for lithium−sulfur batteries (LSB) attaching polysulfide materials. As depicted in Figure 1.4, there are no imaginary frequencies in the phonon dispersion spectra. Slight negative frequencies appear over the high symmetry level G in the SnO spectrum. The graphs revealed that CO, SiO and GeO are dynamically stable while SnO is dynamically unstable [27].

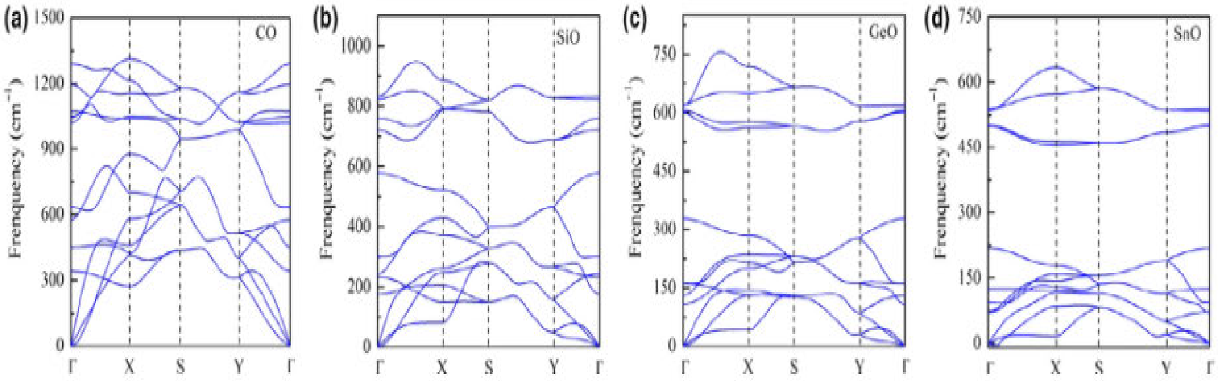


Figure 1.4: Phonon dispersion spectra of PmmaXO monolayers: a) PmmaCO, b) PmmaSiO, c) PmmaGeO, and d) PmmaSnO [2] .

While all of the four described physical quantities can estimate a material stability, they have distinct application areas. Overall, formation energy is a good predictor for stability of the materials as stated earlier. Both the cohesive energy and formation energy are used to determine the stability of materials at an absolute 0K temperature while Gibbs free energy is used to determine materials stability at various pressures and temperatures. The phonon frequency is computer-based and is mainly suitable for small systems calculations. In this study, machine learning algorithms are used to predict formation energy of lithium-ion battery materials, using dataset that was derived from density functional theory calculations.

1.2.2 Band Structures

According to solid state physics, the electronic structure gives data about the range of energy levels populated by electrons inside the material. It is well-established that the electronic conductivity of a material is correlated to the energy band-gap. The electronic structure of an electrode or electrolyte material plays an important role with regard to battery performance. Accordingly, the electronic structure of battery materials can be routinely obtained from density functional theory calculations. Properties such as band structure, density of states, charge distribution and molecular orbitals can be used to provide insights related to battery performance.

For example, recent study investigated Mg_3N_2 as a possible anode material for LIBs due to its (i) stable structure, (ii) lower ion transport barrier, (iii) intercalation potential and (iv) high theoretical capacity [27].

The findings showed a band-gap of 0.91 eV in the pristine Mg_3N_2, suggesting that it is a semiconductor. Upon intercalation, the intermediate $LiMg_3N_2$ and the final product $Li_7Mg_3N_2$ are metallic with no band-gap, implying that the lithiation process improves the electronic conductivity of Mg_3N_2 system. Electronic conductivity in a battery must be maximised across the cathode material. Hence, materials with broad band gap are desired to achieve high electronic conductivity. The correlation between electronic conductivity and energy band gap is defined and analytically expressed as [29]:

$$\sigma = \left(\mu_{\mathrm{e}} + \mu_{\mathrm{h}}\right) q \sqrt{N_{\mathrm{C}} N_{\mathrm{v}}} \mathrm{e}^{-E_{gap}/2kT} \tag{1.4}$$

where u_e is the electronic conductivity, u_h represents hole mobilities, N_C represents densities of states in the conduction, N_v is the valence band, q is the electrical charge of an electron, E_{gap} is the band gap, T is the absolute temperature, and k is the Boltzmann constant [23].

Parameters in equation (1.4) may vary from material to material, since the 1 eV band gap correlates to an electronic conductivity that is likely to be stable cathode. Therefore, in our prediction process, we find the maximal band gap and exclude all materials with smaller band gaps.

1.3 Literature Review

1.3.1 Machine Learning in Materials Science

ML has become increasingly relevant in today's society and particularly in the field of MS. This section aims to provide a brief review on application of ML and its growing position in a range of disciplines of material science, addressing in greater depth some of the problems and opportunities associated with using ML to forecast material properties or acceleration of novel materials design and discovery. ML is described as using computer systems that require no explicit programming in order to understand the work they execute. ML is separated into two main categories, supervised and unsupervised learning which will be discussed later in chapter 2. In this section several research that represent high impact opportunities in using machine learning in materials science are presented, highlighting some representative examples.

1.3.2 Materials Property Prediction

Materials property prediction is the most highly active area of research wherein machine learning is applied to unravel materials properties. Predicting new materials properties from existing databases is one of the most common and less complex area of research using ML. The properties are predicted via regression Y on X followed by the prediction of $Y*$ for new data. Currently, there is no unique approach for mapping feature vectors in X, and this has been a critical challenge. Despite this challenge, different ML algorithms is introduced to map attributes to material properties(target) and new material properties can then be predicted effectively using mapping function developed using training data.

Successful application of ML in materials property prediction includes but not limited to; prediction of bulk stability of perovskite oxides, garnet oxides, and elpasolites, superconducting critical temperatures of complex oxides, formability of novel ternary compounds, melting points of binary and unary solids, dielectric properties of perovskites and polymers, formability of novel half- and full-Heusler intermetallic compounds, casting size of metallic glass alloys, electronic band gap of different classes of inorganic materials, stability and band gap of halide perovskites, dilute metal element solute diffusion barriers in an array of metallic hosts, electromigration of impurity elements in metals, scintillator materials and piezoelectric materials with high electrostrains [30, 31].

In the case of LIBs, many considerations need to be addressed when exploring their property prediction. ML has been used to predict specific LIB properties, namely thermoelectric, superhard materials, thermochemical information, electronic properties, classification of silicate-based cathode, predicting the discharge and charge specific resistance, and structure classification [32]. The biggest challenge in applying machine learning for battery properties is finding a common and reliable definition of feature vectors that impact performance properties.

1.3.3 Materials Design and Discovery

Experiments and computational modeling are standard approaches commonly used in material exploration and innovation. Traditionally, novel materials are discovered through trial and error experimental methods, leading to the desired functional materials. However, the conventional practical methods are time-consuming and expensive. As a way of accelerating the discovery, computational materials modeling attracted researchers' attention in recent years due to their high performance, less costly, and relatively less time-consuming. Computational modeling techniques at various spatio-temporal scales ranging from quantum to continuum macroscopic approaches are well-established and are employed to design and discover new materials for multiple applications. These methods made it possible to calculate atomic structures and a wide range of materials properties across the time-length scale.

The application of machine learning to design and discover new materials is fairly a new research area that integrates autonomous high-throughput experimentation conducted via simulations with decision-making tools guided by ML model predictions. Examples include autonomous efficient experiment design for materials discovery with Bayesian model [33], accelerating the discovery of materials for clean energy [34], autonomy in materials research [35], self-driving laboratory for accelerated discovery of thin-film materials [36], discovery and crystallization of gigantic polyoxometalates [37] and autonomous scientific discovery [38]. This integration has proved to have the potential to perform guided exploration of large materials spaces with limited or no human intervention. The method accelerates materials discovery while reducing human biases in materials searches.

1.3.4 Crystal Structure Prediction

The crystal structure(CS) of a material previously not synthesized is one of the critical problems in computational material design [39,40]. A widely used CS prediction approach is to analyze different atomic configurations a try to find CS with low surface energy. This method can use ML algorithms to foresee the energy surface along with other molecular modeling approaches. However, ML techniques can also be used explicitly to model CS by the mining of a database of known CS to find the likelihood that a material of a given composition has a particular structure form, e.g., bc, hcp, fcc, etc [41].

First-principle CS prediction is challenging since the combinatorial space consists of all possible positioning of atoms in three-dimensional space and has an exceedingly complex energy surface [42]. The structural prediction tool called the "Data Mining Structure Predictor (DMSP)" was then built by [7]. Hautier enhanced the DMSP tool to ternary oxides and estimated new ternary oxide compounds [43]. Upon further filtering the candidate materials using the DFT method, 355 potential ternary oxides have been identified. The identified ternary oxides were not listed in the Inorganic Crystal Structure Database.

1.3.5 Materials Processing and Complex Materials Behaviour

ML techniques are effectively used in the field of process control [44], health sector [45], manufacturing [46] and in material science [47]. Neural networks are capable of mapping complex nonlinear input/output relationships under complex conditions. Surface roughness and material removal rate due to several machining factors provide amounts of interest expected in manufacturing. ML models significantly reduce the burden of repeated experiments and wasteful sections that are time-consuming. Moreover, ML algorithms are used to identify the behavior of the complicated material of alloys subject to high temperature and/or deformation procedures, as well as to model microstructures and phase diagrams that arise from heat treatment and/or deformation processes along with flow stress, stiffness, tensile strength, fracture resistance, and fatigue behavior.

1.4 Research Goal

This study aimed to build ML models to predict formation energy, band-gap, the thermodynamic stability of the lithium cathode materials, and build an accompanying software tool. The study investigates ML algorithms' efficiency based on element constituents of the material descriptors and identifies the best model that gives the maximum accuracy of prediction and classification.

The objectives of this research project were to:

i. acquire relevant data from the Materials Project Database

ii. perform pre-processing steps to generate a set of descriptive attributes as input features using well-known atomic properties to create a list of chemical and physical descriptors of data

iii. use properties (formation energy, band-gap, and thermodynamic stability) calculated from density functional theory to create correct labels of the models during training to predict formation energy, band-gap, and thermal stability of LIBs using ML algorithms

iv. build machine learning models that are capable of accurately predicting materials properties and validate the models.

v. develop a software tool to quickly predict the formation energy, band-gap, and thermodynamic stability of cathode material in LIBs.

1.5 Dissertation Structure

This dissertation is organized as follows:

Chapter 1 - this chapter introduces the study by briefly discussing LIB background, properties of interest to this study, broadly review application of machine learning in materials science and is concluded by listing objectives of the study.

Chapter 2 - gives theoretical background of machine learning algorithms and procedure that was followed to determine input features from constituent elements.

Chapter 3 - describes how the models were built and validated based on prediction of the formation energy and presents key findings of the study.

Chapter 4 - describes how the models were built and validated based on prediction of the thermal stability and energy band gap and presents key findings of the study.

Chapter 5 - presents concluding remarks and recommendations for future work.

Chapter 2

Theoretical Background

2.1 Machine learning

Machine learning (ML) "is a branch of artificial intelligence pertaining to the creation of models that can effectively learn from past data and situations to make decisions" [41]. Thus, ML is described as using computer systems that require no explicit programming to understand the work they execute. After learning from the prescribed data, the algorithms (also known as learning algorithms) construct a model used to make predictions or decisions [48]. Machine learning has dramatically influenced fields such as pattern identification, game theory, bioinformatics, and forecasting. They are continuing to make significant progress in MS research and have great potential for materials research. Recent examples of useful ML applications in material research include rapid and reliable forecasts of structure-property relationships (using past historical data), including crystal structure, material properties, and phase diagrams. [41, 48].

Data has been a hot topic in the 21st century. Data is being generated each day by 2.5 quintillions, and 90% of today's data in the world was developed in the past two years. Automated data processing tools would be desperately required to detect information inside the massive amounts of data [49]. As a matching approach, data mining was implemented. Data mining has attracted many individuals. Data scientists need lots of real data to strengthen their analysis methods. Companies need more detailed models, on the other hand, to make forecasts.

2.2 Machine Learning Approaches

Machine learning algorithms can be differentiated into supervised and unsupervised learning. Supervised learning is used when the training data has a target label, whereas unsupervised learning is used when the training data has no target label. A combination of the two approaches forms what is called semi-supervised learning.

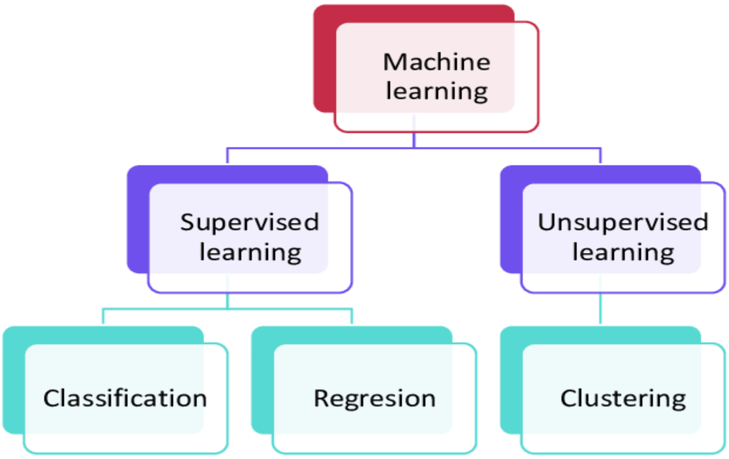


Figure 2.1: Primary approaches to machine learning.

2.2.1 Supervised Learning

The supervised learning algorithms are the most used algorithms to solve problems. Supervised algorithms require the input variable (A) and output variable (B), and a mapping function is used to learn an algorithm from A to B, as presented in equation 2.1.

$$B = f(A) \tag{2.1}$$

A and B referred to training data. To find a function that can correctly predict the B value for new input data, ML attempts to use training data and all other prior information. The method of understanding a function from a set of known values A and B is termed supervised learning. The idea is to bring the mapping function so near into line that you can estimate the B variable for that data from new data A. Also, this is called supervised learning since the method of algorithm learning from the training dataset may be considered an instructor in the learning cycle's supervision. The algorithm generates linear assumptions regarding the target variable and training dataset. Supervised learning algorithms can be divided into two main categories, namely, regression and classification.

i. **Classification** - when the target variable is a class like stable, unstable or True and False.

ii. **Regression** - when the target variable has a real value, like formation energy, or band gap. Popular algorithms used for both classifications and regression are: decision tree classifier, random forest classifier, ridge regression, and linear regression.

2.2.2 Unsupervised Learning

Although supervised learning aims to determine the function (f) that maps input data (A) to an acceptable output value (B), unsupervised learning is focused on the discovery of the correlation between input data (A) itself. Hence, supervised learning attempts by the conditional density of P(f|A, Y, g) to determine relationships of (A) and (B), unsupervised learning aims to determine the features of the joint, marginal density of P(A) [41].

Unsupervised learning means the training data has no association with the target variable. It is termed unsupervised learning as there are no right responses and no instructor as opposed to supervised training. Algorithms are left to explore and view the data's magnetic structure. Clustering and association problems can often include unsupervised learning issues.

i. **Clustering** - clustering issue is when you want to identify clusters with underlying data such as consumer sorting through purchase behaviour.

ii. **Association** - is when you try to find instruction that explain sections of your data, for example compounds with Li are always stable. K-means clustering and apriori association algorithm are common examples of unsupervised learning.

2.2.3 Semi-Supervised Learning

Problems with large volume of training data with no target variable and few training data with target variable are considered as semi-supervised Learning because these problems lie in both supervised and unsupervised learning. A good example are images (e.g. animals, objects, human, etc.) with labels and other images with no labels. This field includes several problems of machine learning in the real world. This is because the marking of data may necessitate expensive or time consuming access to domain experts. Unlabeled data can be obtained and processed cheaply and easily.

2.3 Machine Learning Algorithms

2.3.1 Input Features Development

ML algorithms create a relationship between the dependent and independent attributes and forecast outcomes for new input data based on that learning experience [50]. Descriptors or input variables were created from the composition of the cathode materials. Without the need for computer-demanding simulations, these descriptors should be readily accessible or conveniently measured. We make a mathematical description or descriptors of the composition using the atomic properties of the constituent elements. Sum, average, and variance atomic weight, miracle radius, electronegativity, etc., from the constituent elements' atomic properties, were calculated previously [51].

2.3.1.1 Procedure for developing features using chemical formula

For formula $X_xY_yZ_z$, where elements X, Y, Z share a property j. Formulas to calculate the average, sum, and variance features are shown below:

$$avg_j = \frac{x}{x+y+z}X_j + \frac{y}{x+y+z}Y_j + \frac{z}{x+y+z}C_j$$

stoichiometric sum:

$$sum_j = xX_j + yY_j + zZ_j$$

variance:

$$var_j = \frac{(X-\bar{U})^2 + (Y-\bar{U})^2 + (Z-\bar{U})^2}{N}$$

where:

$$\bar{U} = \frac{X_j + Y_j + Z_j}{N}$$

N is the number of elements

2.3.2 Decision Tress and Random Forest

Decision trees are the most popularly old ML algorithms built for classification problems, and detailed articles have been published. Decision tree algorithms are developed by dividing the data into sections repetitively and applying the simple model to predict output for every section. A decision tree splitting or branches can be depicted graphically. A decision tree has one root node, a collection of internal or divided nodes, and nodes for a leaf [52 56]. Within materials science research, decision trees have previously been used to forecast the tribological properties of several materials on the basis of readily available properties of the materials. Decision trees are analogous to a neural network since the function is displayed as a network of linked nodes, as shown in Figure 2.2. In a decision-tree, however, the network adopts a tree-like form, in which each node can have only one parent node, as illustrated in the figure.

An attribute is labeled on each internal node, the incoming edges of the internal node display values that satisfy attributes limitations. The target attribute that we want to predict is labeled on every leaf node. Figure 2.2 shows a decision tree model to classify three different classes with predictors: a, b, and c.

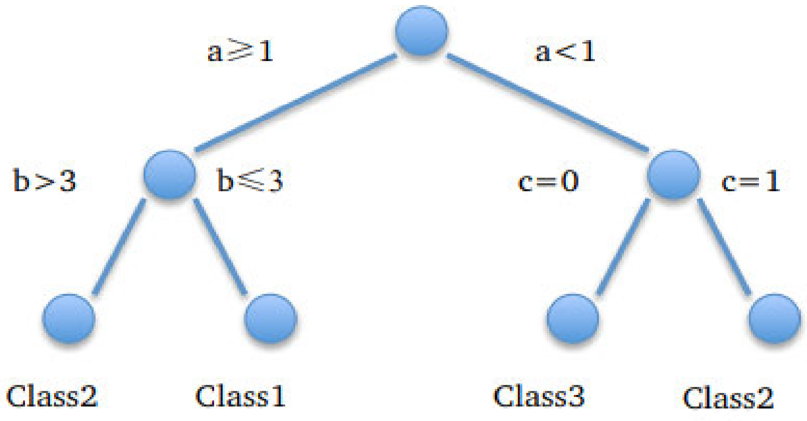


Figure 2.2: Schematic representation of decision tree

The Random Forest (RF) technique has been implemented using decision tree algorithms. RF integrates a variety of decision trees utilizing ensemble technique [57]. Building one model by integrating a group of base models is the highlight concept of the ensemble technique. It is evident that using ensemble techniques gives excellent performance than using a single model. For a classification problem, the data-set is classified based on the individual tree, and the trees vote for the class, and the most voted class is selected for the final outcome. With a regression problem, the outcome is estimated by combining the predictions. Every tree in the model is developed as follows:

i. In the training set, we set x as the number of examples, bootstrap sample of size x is chosen randomly with substitution from actual data. The training set for the growth of a tree will be a subset containing x examples. The remaining data set not used for modeling will then serve as test data to make an error-estimate, termed the error "out of the bag."

ii. Assume we have a total number of features N and some number of features n, where N>n, n features are randomly chosen at each node from the entire N features. In line with a specific objective function, the ideal split on selected n features is utilized to make a binary split on that node.

iii. When a tree grows, no pruning occurs and the trees grow to the maximum extent.

RF model accuracy relies upon the power of the individual trees and their relationships. A robust tree classification model has lower error rates. Strengthening each tree optimizes the forest model's performance while expanding the relationship reduces the random forest performance. Utilizing random feature selection when discovering an efficient tree split is a technique to minimize correlation. The explanation for random feature selection is if specific attributes have an excellent ability to forecast the target feature, many trees use them to make the trees correlate [58]. Minimizing the number of randomly chosen attributes n may mitigate the relationship and the power of the trees. The number of attributes n and the number of trees are some parameters that can be changed or adjusted for the RF.

2.3.3 Extremely Randomized Trees

RF and extremely randomized trees (ERT) are identical since they are based on a random subset of n features to determine the split at each node [59,60]. Significant differences in comparison to the RF model are that every tree is generated from the entire data set, and the discretization for every attribute is randomly chosen to decide a split, rather than to select the best discretization based on a specific objective function as the RF model. The number of random features chosen and the number of trees for the ERT model are also the two most critical customizable parameters.

2.3.4 Gradient Boosting

Boosting is another way to improve the accuracy of the algorithms. It integrates various low attributes that are then measured to some level in terms of their accuracy. The low measured attributes are extended to the strong overall attributes [61 63].

Gradient boosting trees employ boosting methods to combine the power of several distinct decision trees to create a better tree [64]. The algorithm is additively built. The parameters are strengthened by a shift in opposite directions of the gradient to minimize every new base predictor's loss function. The loss function is an arbitrary description of square loss, referred to as absolute loss.

Extreme gradient boosting (xgboost) is a high-speed, efficient ML algorithm that introduces "algorithms under a gradient-boosting framework, with a generalized linear model and gradient-boosted regression tree." Xgboost was implemented by Friedman in 2001 [65], is commonly used in competitions, and is used in several best solutions.

2.3.5 Light Gradient Boosting Machine

Light Gradient Boosting Machine (LGBM) is a new library for gradient boosting proposed in April 2017 by Microsoft [66]. The intention was to make the gradient boosting on decision trees faster. The idea is that when generating new leaves, instead of checking all the splits only some of the leaves are checked: firstly all attributes are sorted and then bucket the observation by developing discrete bins. Instead of iterating over all the leaves, we iterate over all the buckets when we want to break a leaf into a tree. This implementation, according to the developers, is called histogram implementation. The trees are grown in-depth (or leaf wise), keeping the preserved state instead of level wise as other gradient boosting methods. The algorithm chooses to raise the leaf with maximum delta loss and does not increase the level as a whole.

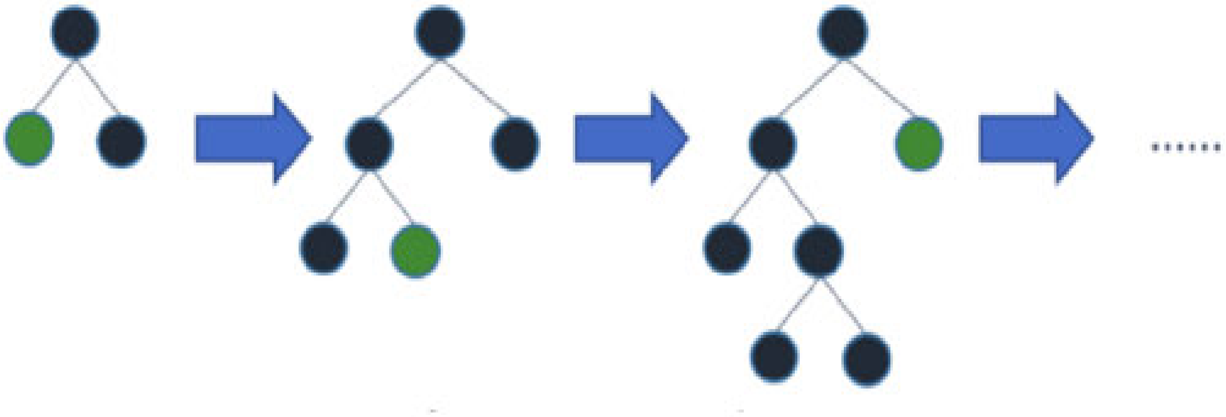


Figure 2.3: Schematic representation of level-wise tree growth

2.3.6 Feature Selection

Feature selection is a method in which some attributes are manually picked from the data correlated to the attribute of prediction or target variable. In practice, datasets have a vast number of features that will be accessible, but not all of them will be necessary for the problem. In many instances, the use of all attributes can degrade the efficiency of the models. This is a well-known challenge in machine learning, also described as the dimensionality curse. The high number of features leads to a very large room with several low densities or even empty vacuums. This

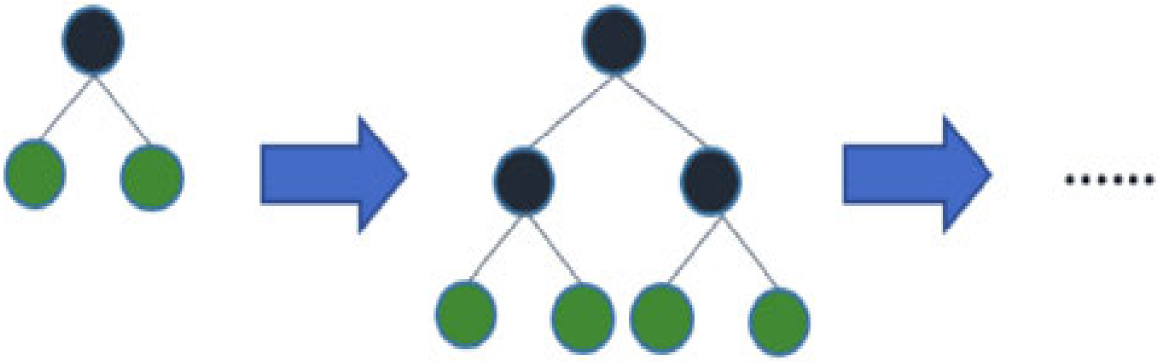


Figure 2.4: Schematic representation of level-wise tree growth

makes it hard to generate genuinely useful findings from the data. It then needs a degree of reduction in dimensions, wherein as much data as possible is preserved, but with fewer features. Before any data can be modeled, three advantages of practicing feature selection are to [67]:

i. **Minimize overtting -** less accurate data gives no ability to create noise-based decisions.

ii. **Increase performance -** Less inaccurate information means that the accuracy of models increases.

iii. **Minimize time to train the model -** Minimal information implies that the algorithms will learn faster.

There are three main feature selection techniques and are briefly summarized as follows:

i. **Univariate feature selection**

   Univariate feature selection reviews every attribute separately. The statistical test is then used to award a score for every attribute. All attributes are graded by ranking, which indicates the strength of the correlation between the attributes and the target attribute. Other standard methods are the information gain, chi-square test, and the correlation coefficients [67]. This approach is straightforward to use but is often inaccurate since only univariate dependencies are considered.

ii. **Recursive feature elimination**

   The recursive feature elimination principle is to continuously construct a model and delete poor attributes at each stage. For this approach, support vector machine (SVM) algorithms are commonly employed. SVM allocates weights to predictors utilized as a test to remove features. Initially, the SVM algorithm is trained using all the attributes and weights are allocated to every attribute. The attributes with the lowest absolute weights are thus omitted from the collection of attributes. The pattern will repeat until enough attributes are being removed. A ranking of features is generated according to the order in which they have been deleted [67].

iii. **Tree-based feature selection**

Other treed-based algorithms, such as categorical boosting (catboost) or xgboost, estimate the overall feature importance values [68]. The tree-based feature selection is simple to use; however, it is not apparent which features should be better combined, and this is a challenge with other approaches. One approach is to choose a learning algorithm and utilize it to create models with recursively discarded ranking-based attributes. Every model is evaluated, and the best set of features is the one with a high score or accuracy.

2.3.7 Feature Construction and Transformation

Feature construction is the process of building new features based on old features or given features. In some instances, the features offered are not enough to guarantee high predictive accuracy, and thus, it is critical to develop new features using the initial ones. This step is generally achieved manually based on intuition and creativity to understand the data. A critical technique is to integrate numerical features with numerous operators, and feature transformation [69]. For example, given features $y_1, y_2, y_3......y_n$, we can built new feature as: $y_1 + y_2$, y_1^2, y_1y_2, and $log(y_n)$.

2.3.8 Hyperparameter Tuning

Grid search is the standard way of conducting hyper-parameter tuning, which is essentially an extensive search of individual defined model parameters [70]. For every hyperparameter value specified in the grid, the grid search will validate the model. In the beginning, the range of the parameters is defined. The idea is to identify the ranges in which the optimal parameter values are found. Depending on the range, the more detailed grid is extended. Suppose the catbooost hyper-parameters are to be changed. Firstly, two parameters must be adjusted: the number of trees (n_estimator) and attributes (max_features) chosen randomly for the optimal split. Then, the messy grid could at first be built as follows:

i. max_features: [1, 0.8, 0.4, 0.3, 0.2]

ii. n_estimators : [100, 150, 200, 250, 300, 1000]

Where max_feature grid values are the percentage of the total number of the features available in the training dataset.

Assuming that optimal parameters values are max_features 0.3 and n_estimators 200, then the precise grid will be generalized as follows:

i. max_features: [0.2, 0.3, 0.4]

ii. n_estimators: [100, 150, 200, 250, 300]

Grid search is straightforward to implement; however, it is continuous and costly for the vast space to search parameters. Another alternative approach is a random search that selects a certain number from an arbitrary array for the model parameter. For each one of the chosen parameter combinations, the model is evaluated. For instance, a search space may be used to modify an RF model hyper-parameter utilizing a random search technique and set to the following:

i. n_estimators: [100, 2000]

ii. max_features: [0.5, 0.9]

Lists of values are chosen randomly for 60 iterations from this field. Random searches are significantly cheaper than grid search, but the optimum search by grid search is also feasible. The two approaches are led by performance metrics usually determined by the cross-validation (CV) on the training set.

2.4 Machine Learning Process

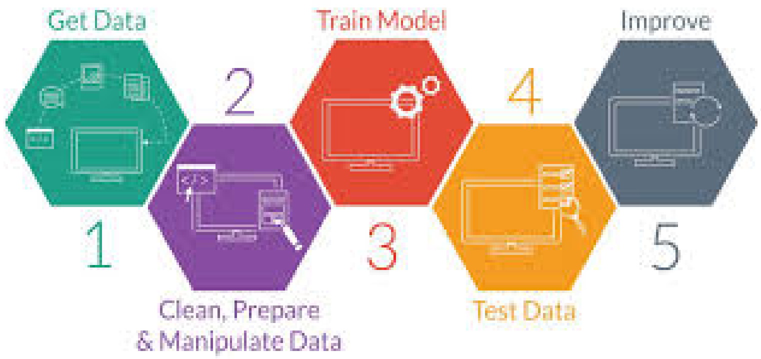


Figure 2.5: Machine learning steps.

ML normally involves the five steps as shown in Figure 2.5:

i. **Get data** - collect dataset from any source.

ii. **Clean, prepare and manipulate data** - data pre-processing aims to organize raw data for possible learning steps in the correct format. Raw data is potentially unstructured, messy, unreliable, and inconsistent. The pre-processing phase converts these data into a shape used by data cleaning retrieval, manipulation, and transformation to teach them.

iii. **Train model** - the training method uses pre-processed input data to create learning algorithms.

iv. **Test model** - the test step then decides the success of the models that have been trained. For e.g., classification performance evaluation includes data sets, output analysis, error estimation, and statistical analyses. The evaluation outcomes will trigger the selected learner algorithms to be modified and specific algorithms to be chosen.

v. **Improve the model** - this step is when the model selected is amended by tuning model parameters, performing feature selection and model assembly.

ML in MS is primarily based on the quantity and consistency of data available, which has been one of the most significant challenges in the field. In specific, for target properties that can only be experimentally tested [71].

Chapter 3

Model Development

3.1 Workflow

Several ML regression and classification models are tested to select the best performing model, i.e., catboost, xgboost, lightboost, extra random tree (ERT), etc. The ML module scikit-learn built on python was used to apply these models. The theoretical background of a few algorithms used was discussed in the previous chapter. The grid search technique was used to find optimum hyper-parameters for each model to boost its performance. The ML model is cross-validated by a randomly selected 70 % of the prediction model (train set), with the remaining 30 % (test set) used to verify the model built. The following are the main stages in the ML workflow:

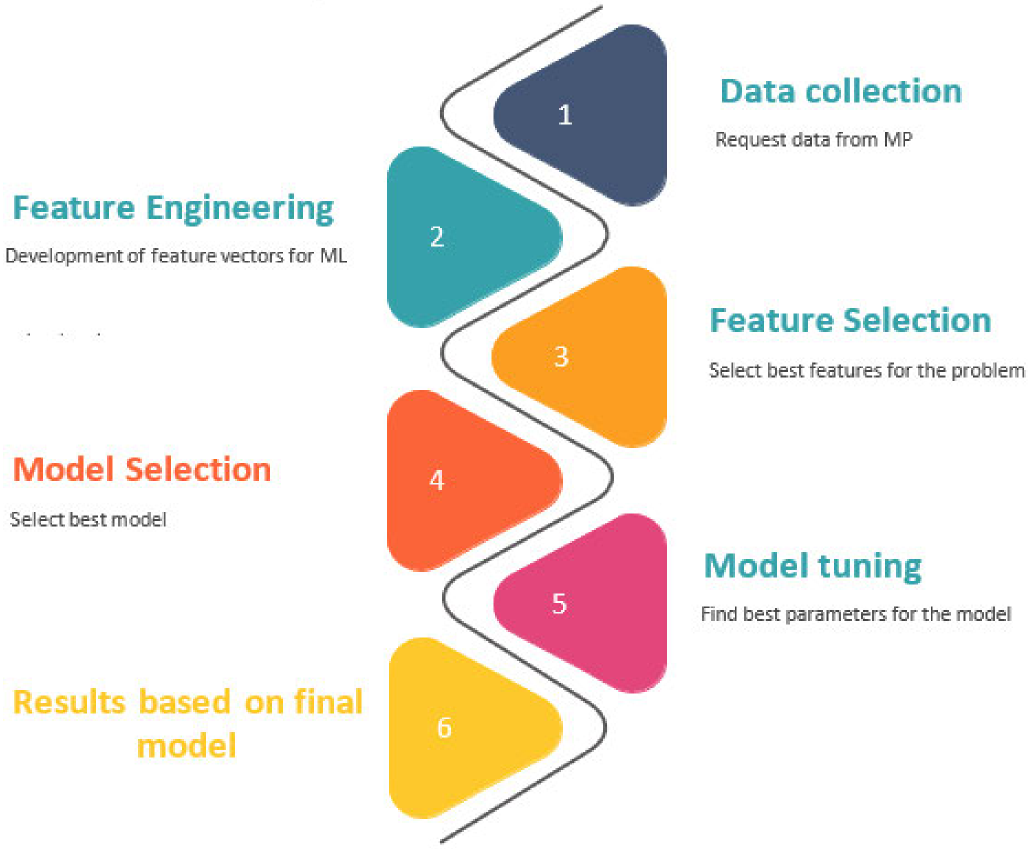

1
Data collection
Request data from MP
2
Feature Engineering
Development of feature vectors for ML
3
Feature Selection
Select best features for the problem
4
Model Selection
Select best model
5
Model tuning
Find best parameters for the model
6
Results based on final model

Figure 3.1: ML Workflow.

3.2 Dataset

We collected the dataset from the Materials Project Database (MPD), which contains the properties of over a million material compounds calculated utilizing DFT. The data hold materials with Li elements. The Vienna Ab initio Simulation Package (VASP) software was used to estimate and optimize DFT material properties stored in MPD. The dataset in MPD contains materials with several properties combined with the dataset from the inorganic crystal structural database (ICSD). The individual DFT computations can be focused on existing ICSD information, past calculations, and updated chemical structures.

The dataset we collected holds the chemical formula, formation energy per atom, energy above the hull, band gap, energy, density for every material. These properties are explained as stated by the MPD glossary.

i. Energy - calculated VASP energy for structure.

ii. Formation energy per atom - computed formation energy at 0 K, and 0 atm.

iii. Energy above hull - the energy of decomposition of this material into the set of most stable materials at this chemical composition, in eV/atom.

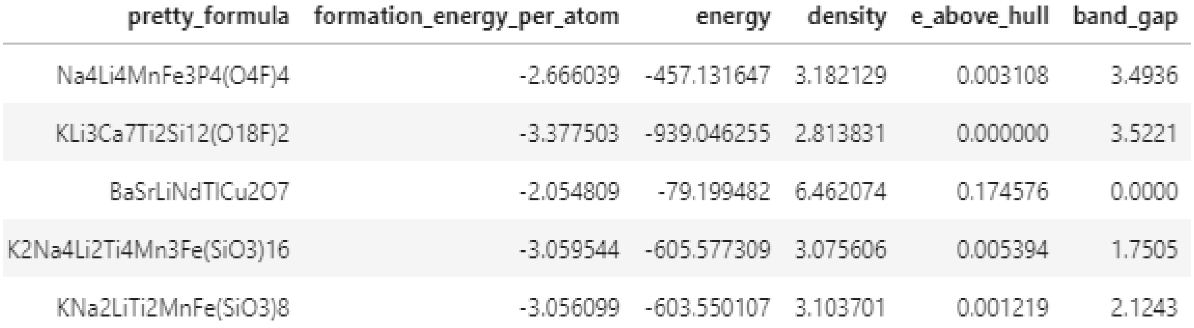
iv. Band gap - the band gap usually defined as the difference in energy (eV) between the top of the valence bands and the bottom of the conductive bands in insulator and semiconductor.

v. Chemical formula - chemical composition.

vi. Density - calculated bulk crystalline density, in grams/cc.

3.2.1 Data collection

We collected the data from the Materials Project Database (MPD), which contains the properties of over a million material compounds as calculated utilizing DFT [72]. We collected the data using python, and we chose a subset of 19479 lithium-containing compounds from MPD, as illustrated in Figure 3.2.



pretty_formula	formation_energy_per_atom	energy	density	e_above_hull	band_gap
Na4Li4MnFe3P4(O4F)4	-2.666039	-457.131647	3.182129	0.003108	3.4936
KLi3Ca7Ti2Si12(O18F)2	-3.377503	-939.046255	2.813831	0.000000	3.5221
BaSrLiNdTlCu2O7	-2.054809	-79.199482	6.462074	0.174576	0.0000
K2Na4Li2Ti4Mn3Fe(SiO3)16	-3.059544	-605.577309	3.075606	0.005394	1.7505
KNa2LiTi2MnFe(SiO3)8	-3.056099	-603.550107	3.103701	0.001219	2.1243

Figure 3.2: Dataset for selected lithium containing materials.

There is no viable replacement tool for looking at the raw data. Taking a look at the raw data, bits and pieces of knowledge and insights can be gained in some way. It can likewise plant seeds that may later develop into thoughts on the most proficient method to better be pre-processed and handle the information for machine learning tasks.

3.2.2 Dataset Split

Now that dataset is readable to ML algorithms, our data was partitioned into a train and test split. The training set will be used to train and optimize the model. The test set will be reserved till the end to ensure that the model is capable of accurate predictions beyond the data used for training. Our unique dataset was then split into two sections. Followed by training the algorithm on the rest part, make predictions on the subsequent part, and assess the forecasts against the normal outcomes. The size of the split can rely upon the size and points of interest of the dataset, despite the fact that it is entirely expected to utilize 70% of the dataset for training and the staying 30% for testing. This strategy is quick and perfect for enormous datasets (a huge number of records), where there is solid proof that the two parts of the dataset are illustrative of the hidden issue. When dataset is large, it is a good idea to split the dataset into 70/30 parts for the algorithm to run faster. A drawback of this strategy is that it can have a high variance. This implies contrasts in the train and test dataset, which bring about significant difference in the gauge of accuracy. In this study, material dataset was split into 70/30 parts for training and test and later assess the performance of a best regression model.

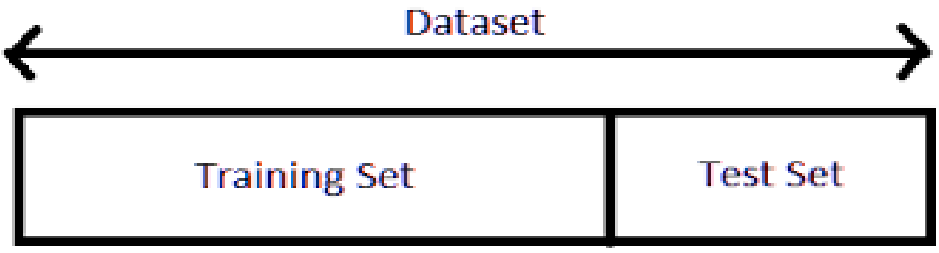


Figure 3.3: Dataset split.

Note that notwithstanding determining the size of the split, we additionally indicate the arbitrary seed in the code. Since the split of the data is irregular, we need to guarantee that the outcomes are reproducible. By indicating the irregular seed, it is guaranteed that similar arbitrary numbers are obtained each time the code is run and thus, a similar split of data. This is significant in the event that we need to contrast this outcome with the evaluated accuracy of another machine learning algorithm or a similar algorithm with an alternate configuration. To guarantee the correlation was apples-for-apples, it should be guaranteed that the models are prepared and tried on the very same dataset.

3.2.3 Cross Validation

Cross-validation (CV) is a technique in machine learning that is used for estimating the reliability of a model result. When computing a model based on all of the data available, it is easy to generate an overfitted model and generally (especially in many dimensions) hard to tell when it is overfitting. Cross-validation solves this problem by only training the model on the part of the data, then exposing it to "the rest of the data" and checking to see the results. Thus, start splitting the data into five-folds, each 30 % of the total dataset. In this study, the dataset was divided into five-folds, as shown in Figure 3.4.

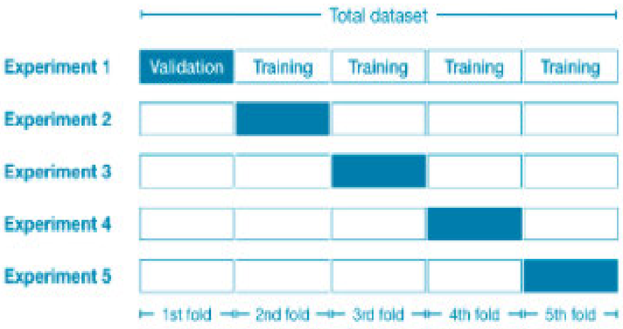


Figure 3.4: Models comparison.

Then for each fold one experiment was performed as explained below:

i. We have the first fold in experiment 1 as validation (or holdout) and everything as training data. These offer a model quality measurement based on a 30% holdout.

ii. We hold data from the second fold in experiment 2 (and use everything except for the second fold for training the model). The holdout set is then used to get the second gauge of model quality.

iii. We rehash this process, utilizing each fold once as the holdout set. Assembling this, 100% of the data is being used as holdout eventually. We end up with a proportion of model quality that depends on the entirety of the rows in the dataset (though we do not use each row simultaneously).

3.3 Feature Vector

This section discusses how the details of the chemical formula are converted to feature vectors. We make the chemical formula so that it can be readable to ML algorithms. We want to give the computer a vector that describes the formula in a meaningful way. The simplest version of this is a vector where each component represents a different formula, i.e. (Li, Mn, O, Zr). Each formula can now be easily encoded. For example $LiMn_2O_3$ can be encoded as (1, 2, 3). Each element in the compounds has atomic weight, so we make a vector called average atomic weight by taking the atomic weight average of all the compound elements. We follow the same procedure to calculate variance, geometric mean, etc. We can usually do better; however, instead of using just the elements we can make a feature from a combination of atomic and elemental properties to make a composition-based feature vector (CBFV).

We used open source xenonpy package to calculate features as shown in Figure 3.5. Xenonpy calculate 290 compositional features for a given chemical composition. This calculation uses the information of the 58 element-level property data recorded in periodic. For example, let us consider a binary compound, $A_{w_A}B_{w_B}$, whose element-level features are denoted by $f_{A,i}$ and $f_{B,i}(i = 1, \ldots, 58)$. Then, the 290 compositional descriptors are calculated: for i 1,. . . ,58.

i. Weighted average $f_{ave,i} = w_A^* f_{A,i} + w_B^* f_{B,i}$

ii. Weighted variance $f_{var,i} = w_A^* \left(f_{A,i} - f_{ave,i}\right)^2 + w_B^* \left(f_{B,i} - f_{ave,i}\right)^2$

iii. Geometric mean $f_{gmean,i} = \sqrt{[w_A + w_B]\, f_{A,i}^{w_A} * f_{V,i}^{w_B}}$

iv. Harmonic mean $f_{\text{hmean },i} = \frac{w_A+w_B}{\frac{1}{f_{A,i}} * w_A + \frac{1}{f_{B,i}} * w_B}$

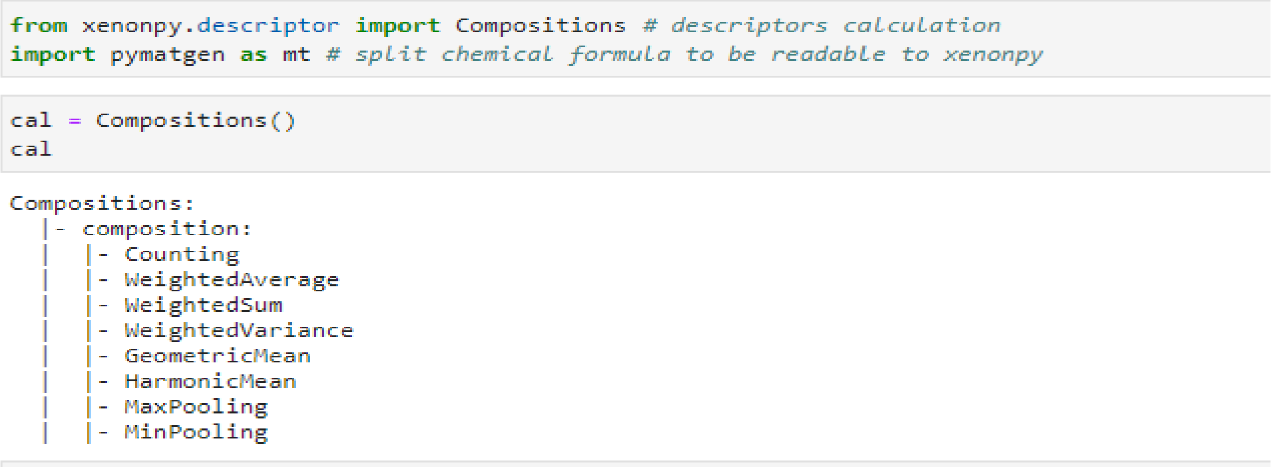
v. Max-pooling $f_{\max,i} = \max f_{A,i}, f_{B,i}$

vi. Min-pooling $f_{\min,i} = \min f_{A,i}, f_{B,i}$

vii. Weighted sum $f_{\text{sum},i} = w_A f_{A,i} + w_B f_{B,i}$

where w_A^* and w_B^* denote the normalized composition summing up to one.

Print cal object will show the structure information. This information tells us the cal has one featurizer group called composition with 5 featurizers in it as shown in Figure 3.5. To use this calculator, users only need to feed a iterable object containing composition information of compounds to the cal.transform or cal.fit_transform method of cal. The input type can be pymatgen. Structure, or dicts which have the structure such as {'H': 2, 'O': 1}. Using our sample data, users will obtain a pandas.DataFrame object containing the calculated descriptor [73, 74].



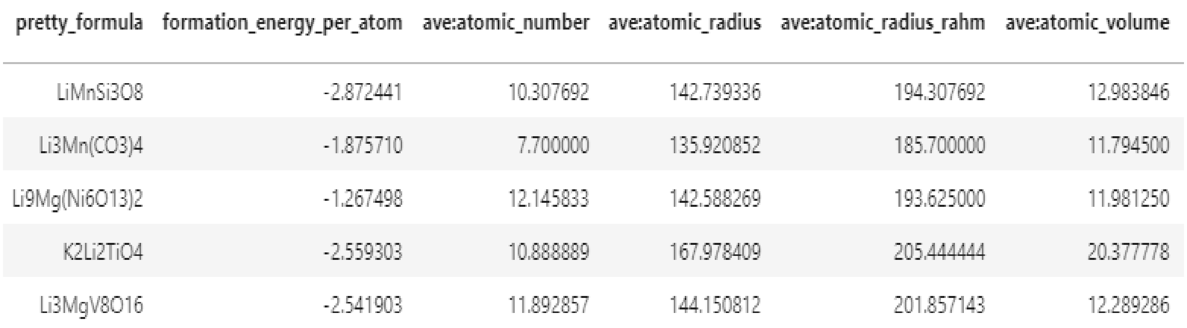
```
from xenonpy.descriptor import Compositions # descriptors calculation
import pymatgen as mt # split chemical formula to be readable to xenonpy
```

```
cal = Compositions()
cal
```

```
Compositions:
  |- composition:
  |   |- Counting
  |   |- WeightedAverage
  |   |- WeightedSum
  |   |- WeightedVariance
  |   |- GeometricMean
  |   |- HarmonicMean
  |   |- MaxPooling
  |   |- MinPooling
```

Figure 3.5: Composition class calculator.

Using our sample data, users will obtain a pandas.DataFrame object containing the calculated descriptors shown in Figure 3.6. If the input is a pandas.DataFrame object, the calculator will first try to read the data columns that have the same name as the featurizer groups. For example, the name of the featurizer group in the example above is composition. Therefore, the whole object entry can be fed into the calculator methods without explicitly extracting the composition column in the samples.



pretty_formula	formation_energy_per_atom	ave:atomic_number	ave:atomic_radius	ave:atomic_radius_rahm	ave:atomic_volume
LiMnSi3O8	-2.872441	10.307692	142.739336	194.307692	12.983846
Li3Mn(CO3)4	-1.875710	7.700000	135.920852	185.700000	11.794500
Li9Mg(Ni6O13)2	-1.267498	12.145833	142.588269	193.625000	11.981250
K2Li2TiO4	-2.559303	10.888889	167.978409	205.444444	20.377778
Li3MgV8O16	-2.541903	11.892857	144.150812	201.857143	12.289286

Figure 3.6: Pre-proccessed features used in the ML model.

3.4 Model Selection

Catboost regressor seems to perform better than all ML models tested as observed in Figure 3.7. Hence, we selected catboost to model the data and for further investigations.

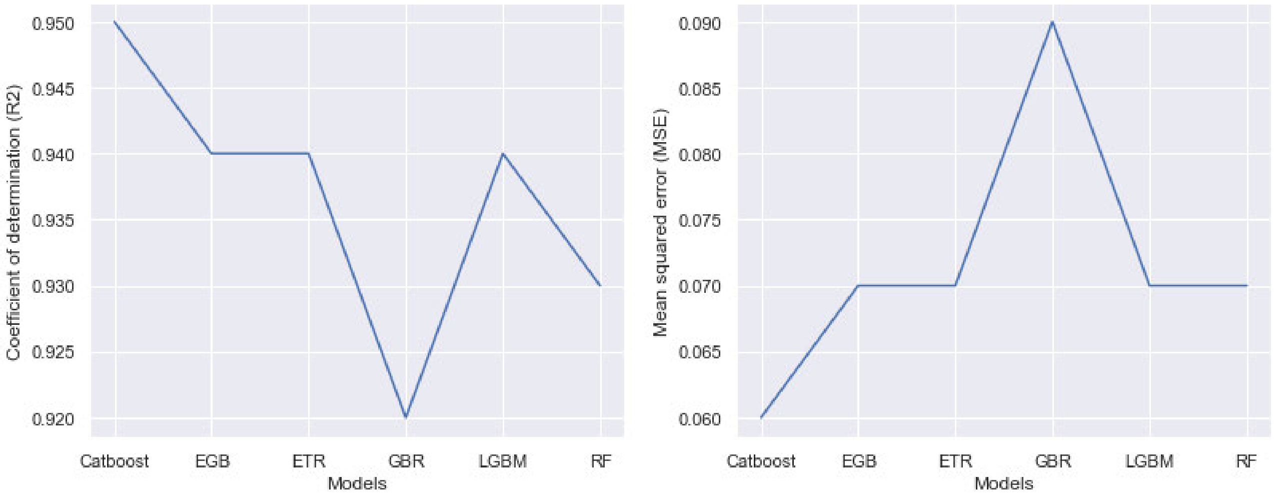


Figure 3.7: Comparison of various ML algorithms to predict properties of materials from the MPD. Categorical boosting (catboost), extreme gradient Boosting (EGB), extra trees regressor (ETR), light gradient boosting machine (LGBM), and random forest regressor (RF).

3.5 Feature Importance/Feature Selection

In order to discover out which descriptors are most essential for effective predictions, descriptor imports can be obtained from model important. The top 18 descriptors are shown in Figure 3.8. Maximum electron negativity, electron negativity Pauling, average d valence are among the top descriptors. A good learning model will enable the researchers to define the properties of a battery system easily and reliably, without the need for extensive testing or simulation.

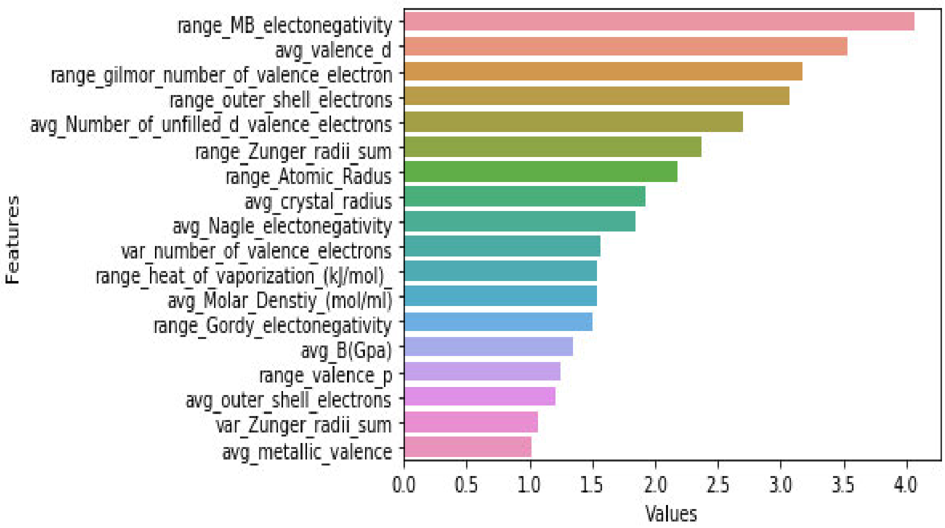


Figure 3.8: Important features selected by catboost model

3.6 Model Tuning

We used catboost to model the dataset. The parameters used in the modeling to get better or worse models were changed. These parameters generally dictate the amount of ’regularization’ applied to the model. Regularization is the metaphorical dial for adjusting model complexity. Using a small amount of regularization may lead to a model that is too complex, grossly overfiting on the training data. Using too much regularization makes the model simple incapable of learning anything useful. The balance is reached by searching over a large range of possible parameter values. We used a grid search technique to tune catboost model. Grid search is done using a cross validation scheme. We defined the parameter space we want to search over. Since catboost have lot of parameters, this study focused only on important parameters and below are parameters that were tuned:

i. Number of trees - "the maximum number of trees" [75].

ii. Max depth - "the maximum depth of the tree" [75].

iii. L2 regularization - "coefficient at the L2 regularization term of the cost function. Any positive value is allowed" [75].

iv. Bagging temperature - "defines the settings of the Bayesian bootstrap. It is used by default in classification and regression modes." Bagging temperature ranges $[0, \infty]$ [75].

Initially we had number of trees 1000, max depth 6, L2 regularization 3, and bagging temperature 1 as default parameter values as presented in Table 3.1. After tuning with 5 fold CV, we found number of trees 350, max depth 10, L2 regularization 5, and bagging temperature 20 to be generally effective values for the models. R^2 score optimized from 0.91 to 0.95.

Table 3.1: Tuned model parameters from training set.

Catboost	Number of trees (default=1000)	Max depth (default=6)	L2 regularization (default=3)	Bagging temperature (default=1)	R^2
Without CV	1000	6	3	1	0.91
After 5 fold CV	350	10	5	20	0.95

3.7 Results and Discussions

Figure 3.9 shows a graphical representation of model performance, containing data points that reflect the predicted formation energy vs. DFT calculated formation energy from the MPD. Catboost regression with elemental descriptors predict formation energy achieving MSE 0.08 eV and coefficient of determination (R^2) 0.92. The performance of the algorithm optimized to mean squared error (MSE) 0.06 eV and R^2 0.95 through model tuning. Performance of catboost, is displayed below:

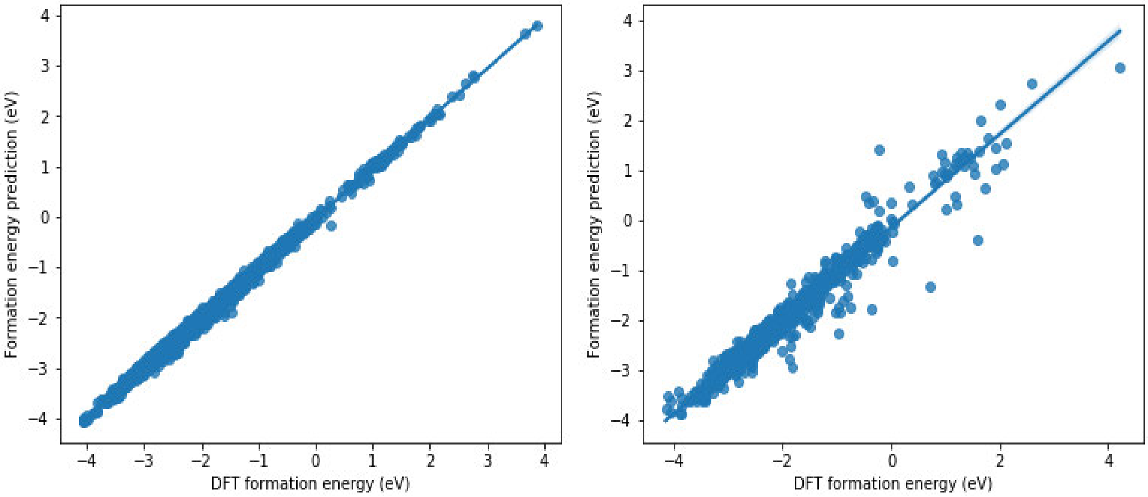


Figure 3.9: Performance of catboost model in training set (left) and in testing set (right)

R^2 is calculated using the following expressions:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} \left(\hat{Y}_i - Y_i\right)^2}{\sum_{i=1}^{n} \left(Y_i - \bar{Y}_i\right)^2} \tag{3.1}$$

where, $\hat{Y}_i$, Y_i, $\bar{Y}$, n are ML predicted values, actual y test value, mean of y test, and sample size of testing set, respectively. R^2 1 indicate that the model predicted the actual values 100% correct, hence the model should achieve R^2 close to 1.

MSE is calculated using the following expressions:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} \left(Y_i - \hat{Y}_i\right)^2 \tag{3.2}$$

where, MSE, n, Y_i, $\hat{Y}_i$ are mean squared error, number of data points, actual values, and ML predicted values, respectively. MSE 0 indicates that the model predicted the actual values 100% correct, hence the model should achieve MSE close to 0.

When data points are focused through the blue line, it indicates that the algorithm performs well and the predicted values are proximate to DFT calculated formation energy. It is necessary to analyze the training dataset's accuracy and visualize a forecast on a test set. This makes it easy to monitor underfitting and overfitting. Suppose data points fit imperfectly with the blue line in the training. In that case, there is an excess chance that the algorithm may not accurately predict formation energies for new materials in the test set. In comparison, if training data points struggle to match, it is a clear indication of poor performance in fresh instances. Catboost, lightgbm, and xgboost models permit both linear and non-linear correlations to be studied. All these three models were tested to track material descriptors to the target property, in this case the formation energy. The other models, such as linear regression, were the worst performers and are not mentioned herein. Catboost model performed much better than other ML models with higher R^2 values, lower MSE after 5-fold CV.

When predicting material properties, machine learning on its own demands, no understanding of the physical principles. The phase of transforming element-derived features into a prediction is managed by the algorithm utilizing strictly statistical techniques. The key features are expected to match the physical principles of the energy of formation. Three critical features in the energy prediction of formation energy are:

i. Average - maximum electron negativity.

ii. Variance - electronegativity Pauling.

iii. Sum - average d valence.

All these features are properties usually related to molecular bonding. This discovery is quite well linked to the common interpretation of formation energy using the vibrational frequencies in the solid because bond type and force may affect the vibrational frequencies. Therefore, the observed attributes confirm the understanding of the physical process of formation energy.

3.8 Web-Tool For Formation Energy Prediction

We have launched a publicly available web application for the battery community to predict the formation energy and energy above the hull of new electrodes in seconds with minimal primary data. The only data needed to forecast formation energy and energy above the hull is the chemical formula (e.g., $LiMnO_2$). Catboost model is used as a back-end ML model for our online formation energy and energy above the hull prediction. The web tool can be utilized to forecast the formation energy and energy above the hull of any battery electrodes containing Li-ion (e.g., $LiNi_3Mn_{0.2}$, Li_2MnO_3, etc.) because the catboost model is only trained on materials containing Li-ion. The tool is open on `https://material-properties-prediction.herokuapp.com`.

In comparison, the community studying the electrodes' contents experimentally and using DFT may want to understand the web tool performance. As evidence of the idea, we relate predicted formation energy with experimental formation energy.

Table 3.2: Materials DFT and predicted formation energies using ML web-tool we developed. The material is more stable when, $E_f > 0$.

Electrodes	**DFT calculation,** E_f (eV)	**ML predicted,** E_f (eV)
$Li_9Mn_{12}Ni_3O_{32}$	-1.96	-1.9536
$Li_3Mn(NiO_2)_4$	-1.553	-1.4966
$Li_3MnCoNiO_6$	-1.818	-1.7599
$Li_4MnCo_2NiO_8$	-1.695	-1.7410
$LiMnO_2$	-2.171	-2.1279

We choose a few electrode materials from MPD and predicted their formation energy and energy above the hull using the web-tool we developed. Table 3.2 tabulates the findings where we identify a strong agreement between DFT calculated and the ML predicted formation energy.

Chapter 4

Thermodynamic Stability

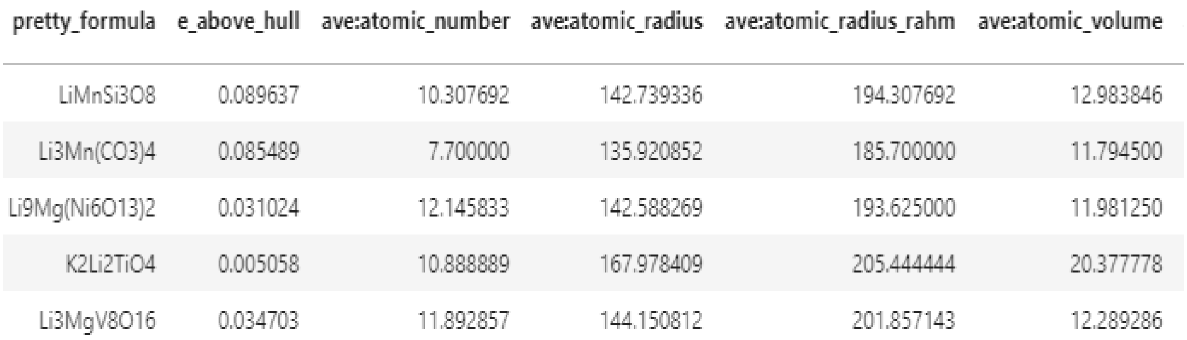
There is an urgent need for cathode materials with high energy density, capacity, and voltage. Oxide materials such as $LiNi_{1-x-y}Mn_xCo_yO_2$ (NMC) and $LiNi_{1-x-y}Co_xAl_yO_2$ (NCA), rich in Ni have been proven to have higher energy density and cathode capacity compared to conventional $LiCoO_2$. The technique for optimizing the Li-ion battery's energy density is to improve the operating voltage of cathode materials by using high-voltage cathode materials such as olivine ($LiNi_{0.5}Mn_{1.5}O_4$) and spinel ($LiNi_{0.5}Mn_{1.5}O_4$). However, their low thermodynamic stability and heavy electrolyte reactivity is the main challenge in utilizing these high energy density cathode materials. Hence, a need to discover a new material that demonstrates good electrochemical stability while maintaining high energy density during the electrochemical cycling [76].

4.1 Thermodynamic Stability Classification

Classification is a machine learning method for separating the dataset into some groups. Since the thermodynamic stability (stable and unstable) is defined in the model, such a model is considered a supervised learning method.

4.2 Data Collection and Data Preparation

As discussed in chapter 3, the same dataset was used to predict Li-ion battery materials' thermodynamic stability. We presume that the materials are unstable or metastable in the repository when energy above the convex $E_{hull} < 0$ eV. Dataset appears, as shown in Figure 4.1 after being prepared for ML algorithms to learn from it. The E_{hull} characterizes the variation between the zero-temperature energy of all phases and the most stable phase [23].



pretty_formula	e_above_hull	ave:atomic_number	ave:atomic_radius	ave:atomic_radius_rahm	ave:atomic_volume
LiMnSi3O8	0.089637	10.307692	142.739336	194.307692	12.983846
Li3Mn(CO3)4	0.085489	7.700000	135.920852	185.700000	11.794500
Li9Mg(Ni6O13)2	0.031024	12.145833	142.588269	193.625000	11.981250
K2Li2TiO4	0.005058	10.888889	167.978409	205.444444	20.377778
Li3MgV8O16	0.034703	11.892857	144.150812	201.857143	12.289286

Figure 4.1: Part of materials we selected to train our algorithms.

4.2.1 Dataset Split

Now that we have a readable dataset to ML algorithms, we partition our data into a train and test split. The training set will be used to train and optimize the model, while the test set will be reserved until the end to ensure that the model can accurately predict properties beyond the data used for training. Since $E_{hull} > 0$ implies unstable compounds and stable E_{hull} <0 are stable [23], we created a target label called thermodynamic stability. From the dataset, we have 8940 materials of which, only 1616 are stable, and the rest are unstable materials. These lead us to a term called imbalance classes or target.

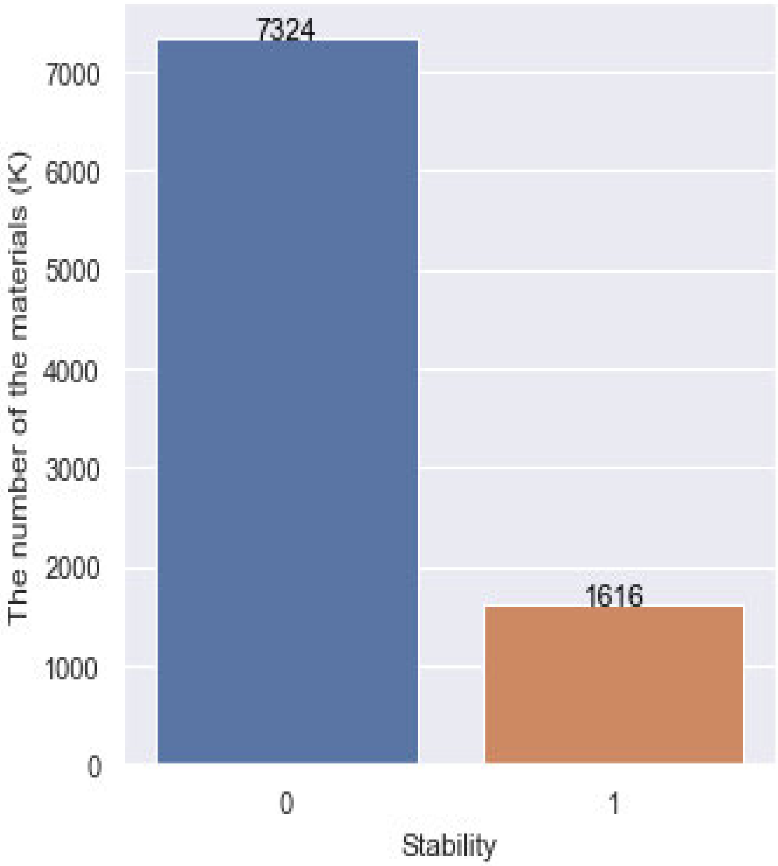


Figure 4.2: Label distribution. Here we have 7324 unstable and 1616 stable compounds.

Imbalance classes are a typical issue in machine learning classification wherein there are many observations of each class. Class imbalances can be observed in many different fields, including medical care, spam detection, and fraud identification. This section will look at one way to deal with an imbalanced class issue using material data. Our goal is to accurately identify the minority class of stable material. It can be seen from Figure 4.2 plot that we have a very imbalanced class; only 16% of our dataset belongs to the stable class. This is a challenge since many machine learning models are built to optimize overall accuracy, which may not be the right metric to use, particularly for imbalanced classes. For instance, if we predicted that all materials are not stable, we will have a classification accuracy of more than 82%, which is misleading. Classification accuracy is estimated as the amount of correctly predicted over total predictions times 100.

4.3 Model Selection

Xgboost classifier seems to perform better than all the other ML models tested as observed in Figure 4.3. Hence, we selected xgboost to model the data and for further investigations.

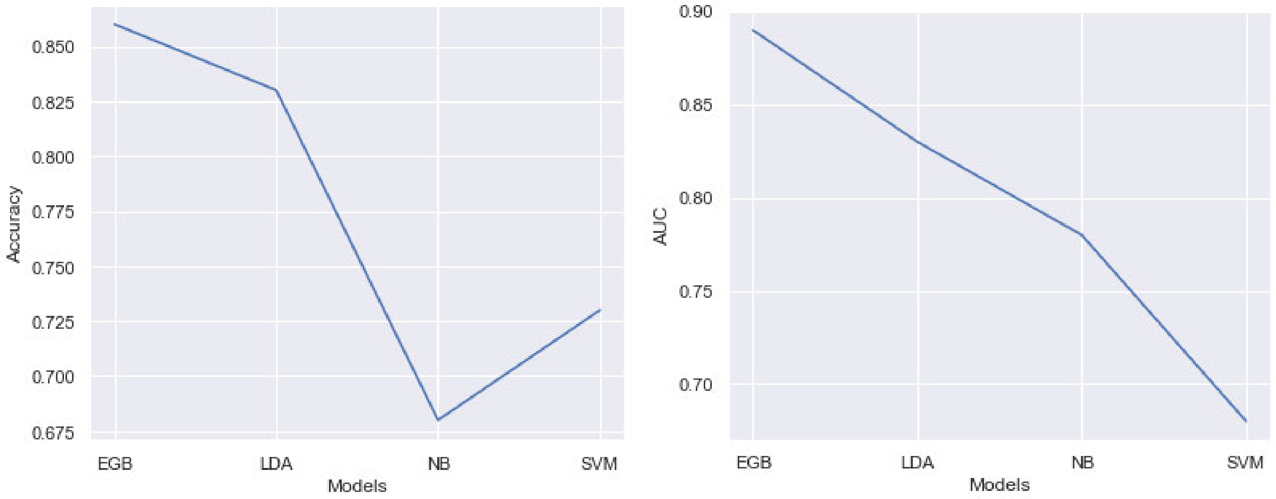


Figure 4.3: Comparison of various ML algorithms to predict properties of materials from the MPD. Extreme gradient boosting (EGB or xgboost), linear discriminant analysis (LDA), naive bayes (NB), and support vector machine (SVM).

4.4 Feature Importance

Xgboost provides a way to determine the value of any feature in the dataset. The top 18 descriptors are shown in Figure 4.4. Variance metallic valence, range atomic weight, the sum of atomic concentration are among the top descriptors. As alluded in the previous chapter, a good learning model will enable the user to define a materials' properties easily and reliably, without the need for extensive testing or simulation.

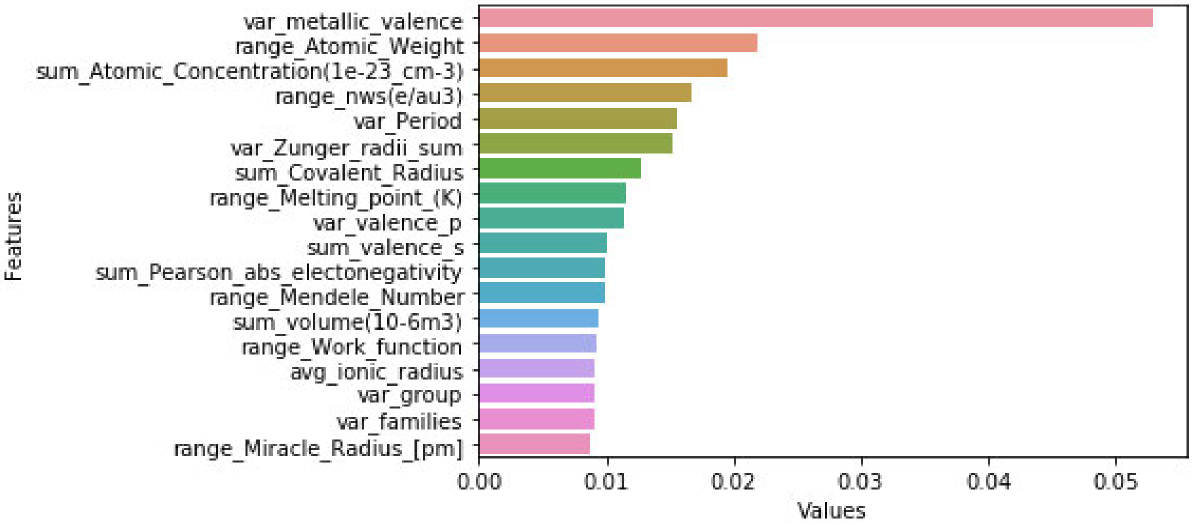


Figure 4.4: The best 18 features and their values from the xgboost model classifier.

4.5 Model Tuning

We used a grid search technique to tune xgboost model. A grid search is done using a cross-validation scheme and the parameter space we want to search over was defined. Since xgboost has many parameters, we only focused on essential parameters related to this particular model. Below are the parameters that were tuned:

i. Max depth - is the maximum tree depth allowed. Tree depth is the length of the longest path from the root node to a leaf node. Making this too high will give our model more variance, or more potential to overfit. Similar to number of trees, the more we increase this, the longer our training period will be. Max depth range $[0, \infty]$ [77].

ii. Gamma - "the gamma parameter specifies the minimum loss reduction required to add a new split in a tree. A larger value for gamma has the effect of pre-pruning the tree, by making harder to add splits". "Gamma range $[0, \infty]$. "This decides whether a node will split based on the expected loss reduction after the split". Gamma represents the minimum loss reduction required for a node to split [77].

iii. Subsample - "subsample ratio of the training instances. Setting it to 0.5 means that xgboost would randomly sample half of the training data prior to growing trees. and this will prevent overfitting. Subsampling will occur once in every boosting iteration". Subsample range $[0, 1]$ [77].

Initially we had max depth 6, gamma 0, and subsample 1 as default parameter values as shown in Table 4.1. After tuning with 10 fold CV we found max depth 3, gamma 0.07, and subsample 0.4 to be generally effective values for the models. Accuracy and Area under the curve (AUC) score were optimized from 85% to 86% and 0.84% to 0.89%, respectively.

Table 4.1: Tuned model parameters from training set.

Xgboost	Max depth (default=6)	Gamma (default=1)	Subsample (default=1)	Accuracy	AUC
Without CV	6	0	1	85%	0.84%
After 10 fold CV	3	0.07	0.4	86%	0.89%

4.6 Results and Discussions

A confusion matrix is an important tool to evaluate performance of the classifier. It provides a clear picture of the performance of the classifier. It creates a matrix where the frequency of hits and misses of each label are forecast. In order to evaluate the confusion matrix, there must be forecasts to link them with actual targets.

Table 4.2: TP, FP, FN, and TN mean true positive, false positive, false negative, true negative, respectively.

	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

We can show the confusion matrix built for the prediction of our model as follows:

The matrix with the following parameters was obtained, TP 254, FP 113, FN 231 and TN 2085. It can be seen from Table 4.3 that 113 materials were classified incorrectly as stable and 231 as unstable. According to the results, the best algorithm (xgboost) achieves 86% and 89% AUC and accuracy respectively,

Table 4.3: TP, FP, FN, and TN mean true positive, false positive, false negative, true negative, respectively.

	Predicted 0	Predicted 1
Actual 0	2085	113
Actual 1	231	254

classifying 254 materials as stable and 2085 as unstable incorrectly.

The precision, recall, specify and f1-score can be calculated as follows:

i. **Classification Accuracy**

$$\text{Accuracy} = \frac{X}{Y} \tag{4.1}$$

where X TP+TN, and Y TP+TN+FP+FN

Classification accuracy is calculated as the amount of correctly predictions times 100 divide by the total amount of the sample. In ML, we use the accuracy metric when the equivalent number of a sample belongs to each class essentially accuracy metric when just one class holds a greater part of the sample. Notably, suppose the training data had 99% category X and 1% category Y instances. In that case, the model can then hit 99% by predicting every sample of the category is X. If a similar model is evaluated with 60% samples from category X and 40% from category Y, then the accuracy of the test will be reduced to 60%. Hence, classification accuracy will provide a misleading idea that high accuracy is achieved.

ii. **Precision**

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.2}$$

iii. **Recall/Sensitivity**

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.3}$$

To minimize FN, the recall should be as close to 100%. To minimize FP, the precision should be as close to 100%

iv. **Specificity/True Negative Rate (TNR)**

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{4.4}$$

v. **F1 Score**

F1 score is the harmonic mean between precision and recall. It indicates how accurate the classification algorithm is (how many cases it accurately classified) and how stable it is (no large number of instances it classifies correctly). The higher the F1 score, the best is model performance. F1 score range between [0, 1] and is calculated as follows:

$$F1 = 2 * \frac{1}{\frac{1}{\text{precision}} + \frac{1}{recall}} \tag{4.5}$$

vi. **Area under the curve (AUC)**

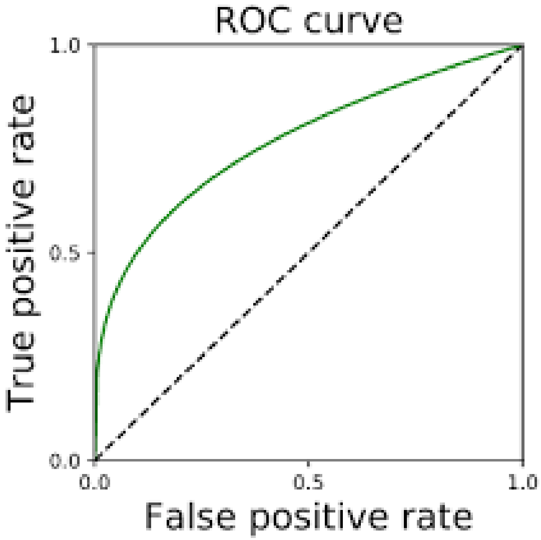


Figure 4.5: Area under the curve (AUC)

It is very different to interpret a receiver operating characteristic curve (ROC) plot than a regular line plot. We get a line that tracks the likelihood cut from 0 to 1 on the top right to left bottom. This is an overview of the sensitivity and specificity performance of the entire likelihood cutoff intervals ranging from 0 to 1. The region under the ROC is one for a perfect model. The cut-off values drop from 1 to 0 when the curve is drawn from the bottom left. When the model is correct, it should be predicted that more actual events will result in high sensitivity and low FPR. The greater the region in the ROC curve, the more influential the model is. The ROC curve is the best indicator for the performance of the model for various probability cutoff values. AUC is the percentage of the ROC plot that is underneath the curve. The AUC describes the ability of a model to differentiate the negative classes from the positive classes. AUC is helpful even when there is a high-class imbalance (unlike classification accuracy)

4.7 Band Gap Prediction

The electronic conductivity in a battery must be minimum across the electrolyte and maximum across the cathode [23]. We predict the band gap of the materials to guarantee high electronic conductivity in the cathode. The same dataset and methodology as discussed in chapter 3 was used to predict the energy band gap.

4.7.1 Results and Discussions

Catboost regression still achieves the best results compared to other ML algorithms. Catboost predicted band gap achieve MSE 0.53 eV and R^2 0.88 on training set and MSE 0.75 and R^2 0.64 on testing set. It is noted that the Catboost model performed much better on the training set and overfitted a bit on the test set. The model tends to optimize overfitting when increasing the number of trees parameter. The model optimized to MSE 0.63 eV and R^2 0.70 on the test set by only changing the number of trees from default value of 1000 to 700 trees. The performance of Catboost is displayed in Figure 4.6 below:

We decided not to deploy the model since we want to achieve at least R^2 of >0.90. Hence, future work will improve the model by assembling catboost and other algorithms and finding best model parameters since it is time-consuming.

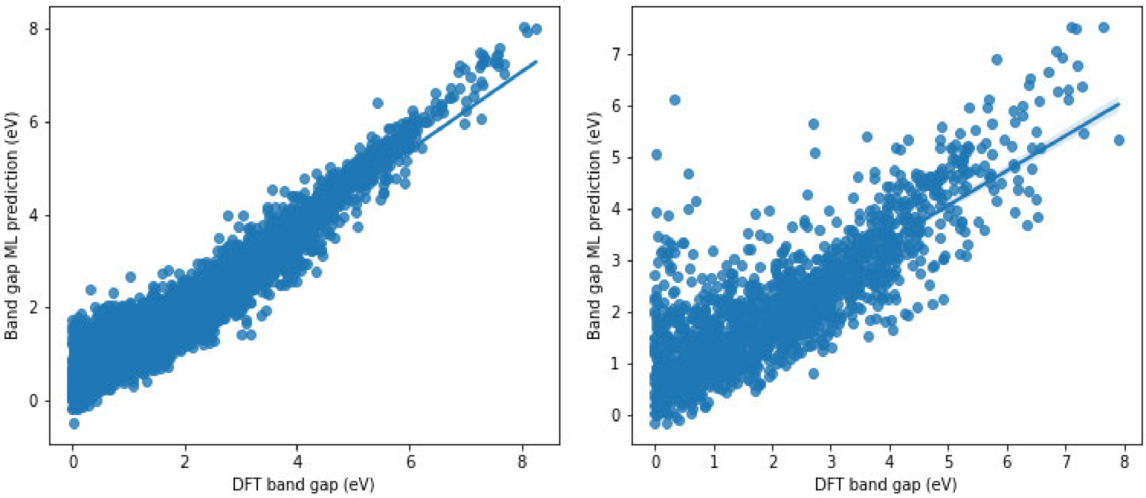


Figure 4.6: Performance of Catboost model in training set (left) and in testing set (right)

Chapter 5

Conclusion

5.1 Conclusion

In this dissertation, we developed detailed ML algorithms for predicting the properties of lithium ion battery materials. We illustrated complete application of machine learning in materials science by presenting two different material problems: finding new possible cathode material for LIBs and identifying stable materials. The approach operates by using ML techniques to create models that predict the materials' properties to estimate chemical effects in a wide range of attributes. Furthermore, our models' performance is improved by tuning the models' parameters. This dissertation shows that indeed machine learning models can provide reliable solutions in material science. To build new models through strategy that was developed in this study, only required understanding of different learning algorithms. From the study, it was established which of the machine learning algorithms enhances accuracy of the models. The three algorithms, namely Catboost Regressor, Light Gradient Boosting Machine and Extreme Gradient Boosting were found to be the most accurate in predicting the formation energy, thermal stability and band gap of LIB materials. The Catboost Regressor acheived an accuracy of over 0.95 as measured by coefficient of determination (R^2) for the formation energy. In addition, the study was able to automate data splitting strategies, which plays an important role in model accuracy. This approach was made to be flexible to cater for various problems in other various applications and made easier to deploy machine learning in the design of new materials faster and more efficiently.

5.2 Future Studies

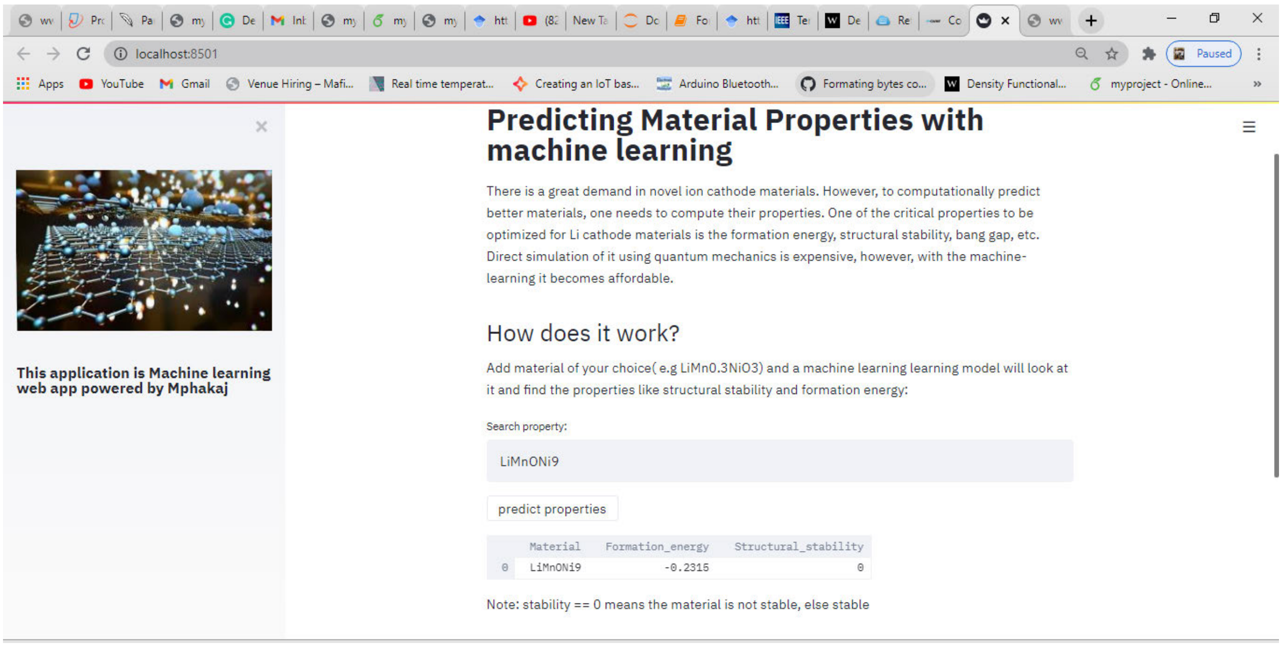
The findings obtained in this study can be strengthened further by exploring other properties that are directly linked to battery performance, e.g. voltage, energy density, power density, etc. Elementary descriptors gave a much more exact representation of materials and seemed more effective when used in ML.

Since Catboost, in most cases, showed better results than other algorithms, it should be given greater attention in further studies to optimize the performance. In future studies, ways of automatically tuning the model could be explored since with the grid search technique is becoming impossible due to expensive computation.

Appendix A

Developed Web Application

Shown in appendix A is a snapshot of the web-app that was developed as part of this study. The app is capable of predicting formation energies of lithium-ion battery materials using the GUI.



This application is Machine learning web app powered by Mphakaj

Predicting Material Properties with machine learning

There is a great demand in novel ion cathode materials. However, to computationally predict better materials, one needs to compute their properties. One of the critical properties to be optimized for Li cathode materials is the formation energy, structural stability, bang gap, etc. Direct simulation of it using quantum mechanics is expensive, however, with the machine-learning it becomes affordable.

How does it work?

Add material of your choice(e.g LiMn0.3NiO3) and a machine learning learning model will look at it and find the properties like structural stability and formation energy:

Search property:

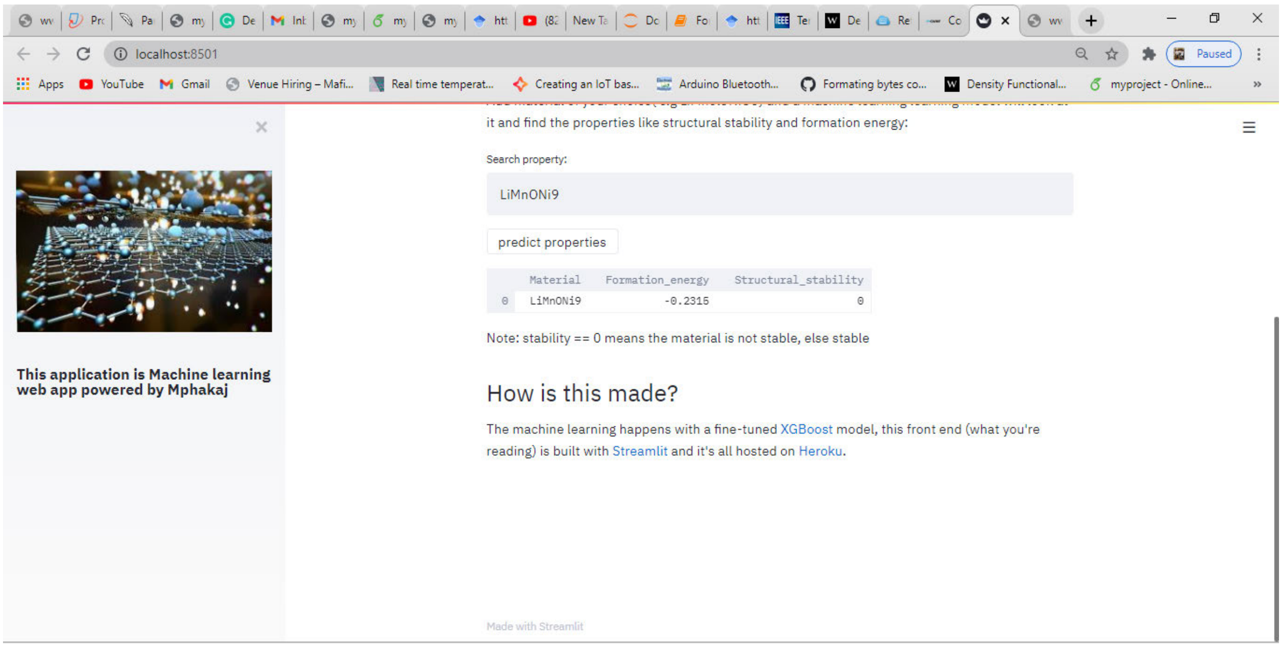
LiMnONi9

predict properties

	Material	Formation_energy	Structural_stability
0	LiMnONi9	-0.2315	0

Note: stability == 0 means the material is not stable, else stable

Figure A.1: Software tool view page



This application is Machine learning web app powered by Mphakaj

it and find the properties like structural stability and formation energy:

Search property:

LiMnONi9

predict properties

	Material	Formation_energy	Structural_stability
0	LiMnONi9	-0.2315	0

Note: stability == 0 means the material is not stable, else stable

How is this made?

The machine learning happens with a fine-tuned XGBoost model, this front end (what you're reading) is built with Streamlit and it's all hosted on Heroku.

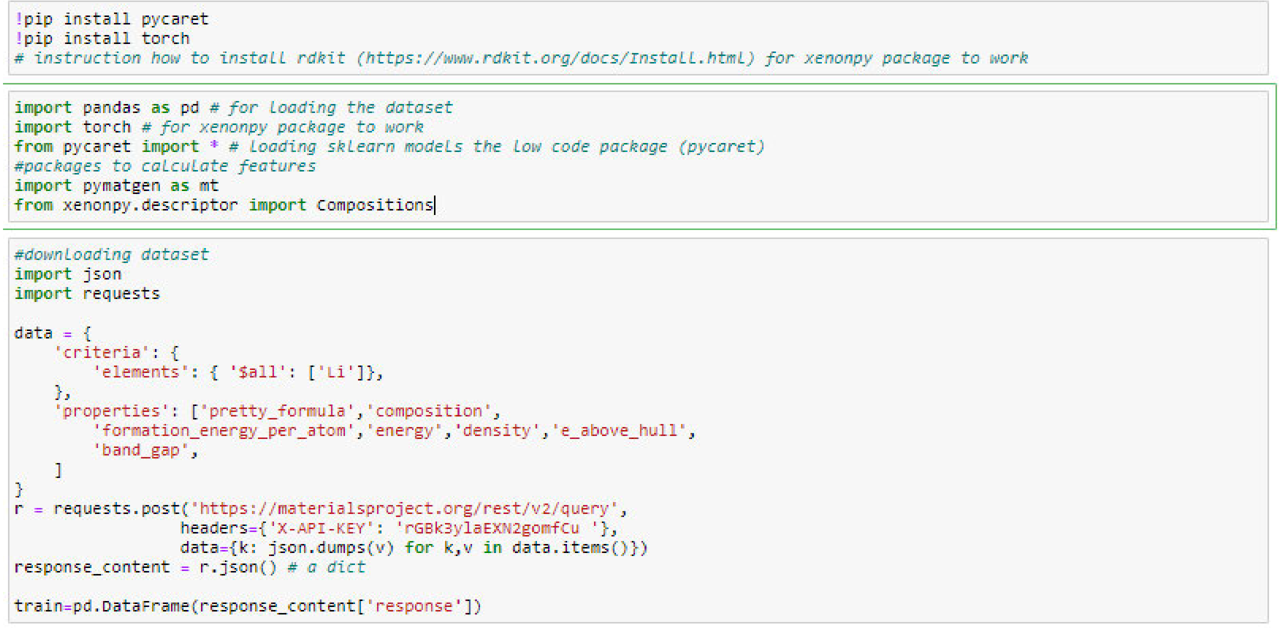
Made with Streamlit

Figure A.2: Software tool view page

Appendix B

Code Details

Presented in appendix B are the details of the code that was developed to build and validate the machine learning models. The snapshots are presented for demonstration only, and will not be discussed since key features were discussed in the dissertation.

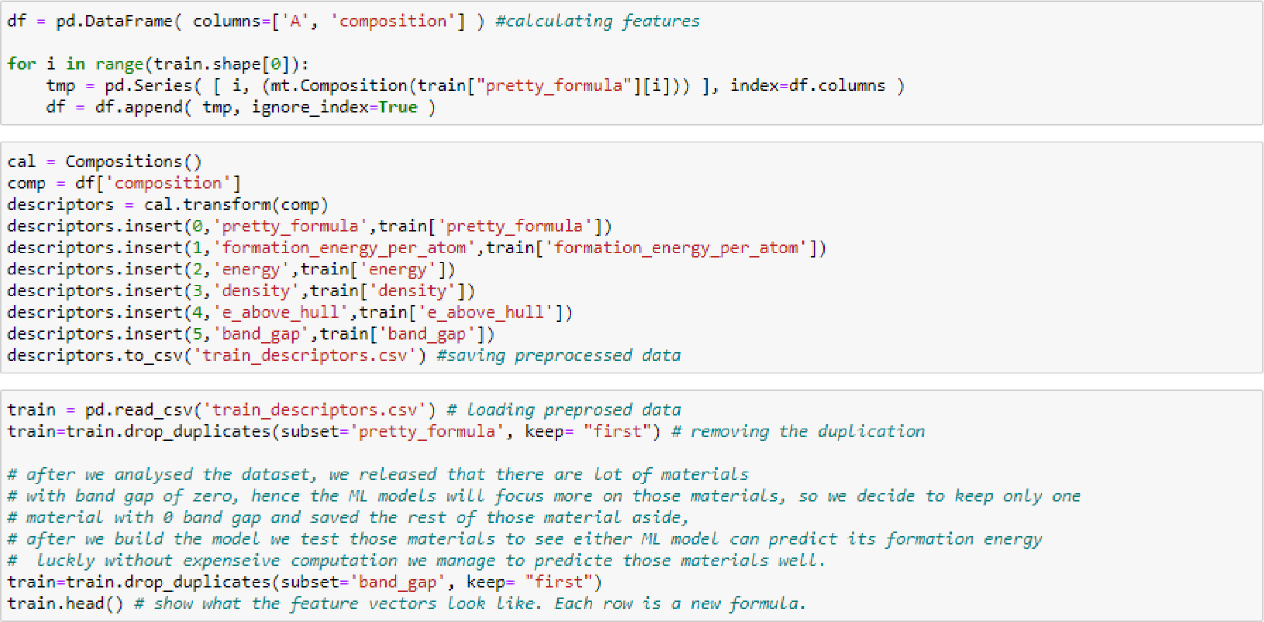


```
!pip install pycaret
!pip install torch
# instruction how to install rdkit (https://www.rdkit.org/docs/Install.html) for xenonpy package to work
```

```
import pandas as pd # for loading the dataset
import torch # for xenonpy package to work
from pycaret import * # loading sklearn models the low code package (pycaret)
#packages to calculate features
import pymatgen as mt
from xenonpy.descriptor import Compositions
```

```
#downloading dataset
import json
import requests

data = {
    'criteria': {
        'elements': { '$all': ['Li']},
    },
    'properties': ['pretty_formula','composition',
        'formation_energy_per_atom','energy','density','e_above_hull',
        'band_gap',
    ]
}
r = requests.post('https://materialsproject.org/rest/v2/query',
                 headers={'X-API-KEY': 'rGBk3ylaEXN2gomfCu '},
                 data={k: json.dumps(v) for k,v in data.items()})
response_content = r.json() # a dict

train=pd.DataFrame(response_content['response'])
```

Figure B.1: Dataset collection

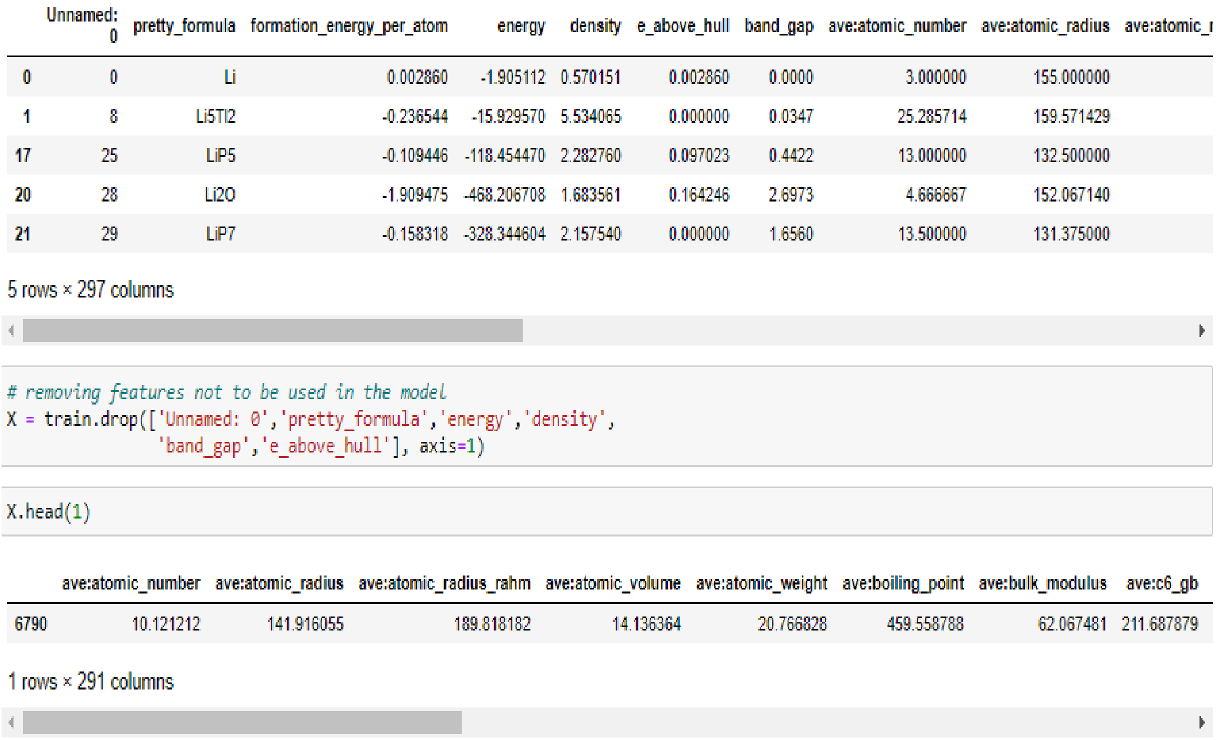


```
df = pd.DataFrame( columns=['A', 'composition'] ) #calculating features

for i in range(train.shape[0]):
    tmp = pd.Series( [ i, (mt.Composition(train["pretty_formula"][i])) ], index=df.columns )
    df = df.append( tmp, ignore_index=True )
```

```
cal = Compositions()
comp = df['composition']
descriptors = cal.transform(comp)
descriptors.insert(0,'pretty_formula',train['pretty_formula'])
descriptors.insert(1,'formation_energy_per_atom',train['formation_energy_per_atom'])
descriptors.insert(2,'energy',train['energy'])
descriptors.insert(3,'density',train['density'])
descriptors.insert(4,'e_above_hull',train['e_above_hull'])
descriptors.insert(5,'band_gap',train['band_gap'])
descriptors.to_csv('train_descriptors.csv') #saving preprocessed data
```

```
train = pd.read_csv('train_descriptors.csv') # loading preprosed data
train=train.drop_duplicates(subset='pretty_formula', keep= "first") # removing the duplication

# after we analysed the dataset, we released that there are lot of materials
# with band gap of zero, hence the ML models will focus more on those materials, so we decide to keep only one
# material with 0 band gap and saved the rest of those material aside,
# after we build the model we test those materials to see either ML model can predict its formation energy
#  luckly without expenseive computation we manage to predicte those materials well.
train=train.drop_duplicates(subset='band_gap', keep= "first")
train.head() # show what the feature vectors look like. Each row is a new formula.
```

Figure B.2: Feature vector calculations



	Unnamed: 0	pretty_formula	formation_energy_per_atom	energy	density	e_above_hull	band_gap	ave:atomic_number	ave:atomic_radius	ave:atomic_r
0	0	Li	0.002860	-1.905112	0.570151	0.002860	0.0000	3.000000	155.000000	
1	8	Li5Tl2	-0.236544	-15.929570	5.534065	0.000000	0.0347	25.285714	159.571429	
17	25	LiP5	-0.109446	-118.454470	2.282760	0.097023	0.4422	13.000000	132.500000	
20	28	Li2O	-1.909475	-468.206708	1.683561	0.164246	2.6973	4.666667	152.067140	
21	29	LiP7	-0.158318	-328.344604	2.157540	0.000000	1.6560	13.500000	131.375000	

5 rows × 297 columns

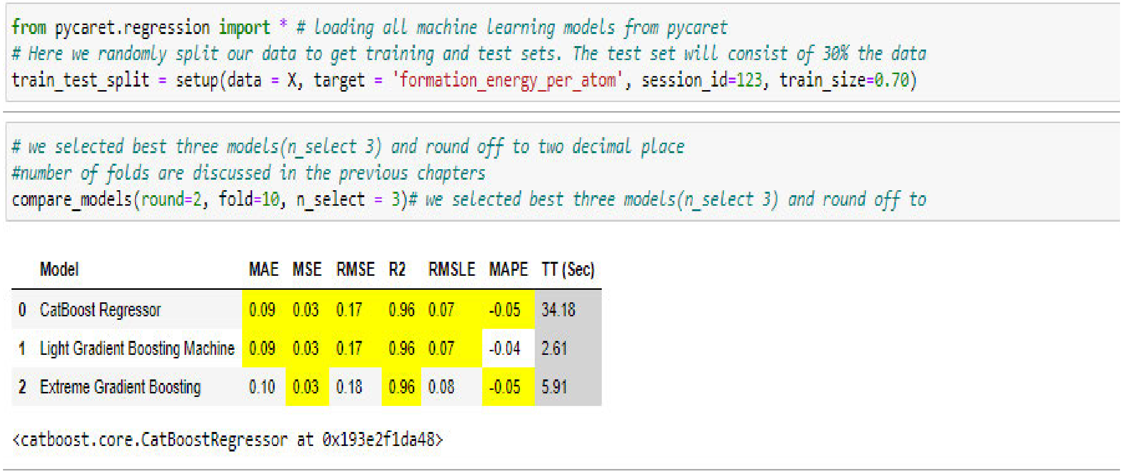
```
# removing features not to be used in the model
X = train.drop(['Unnamed: 0','pretty_formula','energy','density',
                'band_gap','e_above_hull'], axis=1)
```

```
X.head(1)
```

	ave:atomic_number	ave:atomic_radius	ave:atomic_radius_rahm	ave:atomic_volume	ave:atomic_weight	ave:boiling_point	ave:bulk_modulus	ave:c6_gb
6790	10.121212	141.916055	189.818182	14.136364	20.766828	459.558788	62.067481	211.687879

1 rows × 291 columns

Figure B.3: Preprocesed data



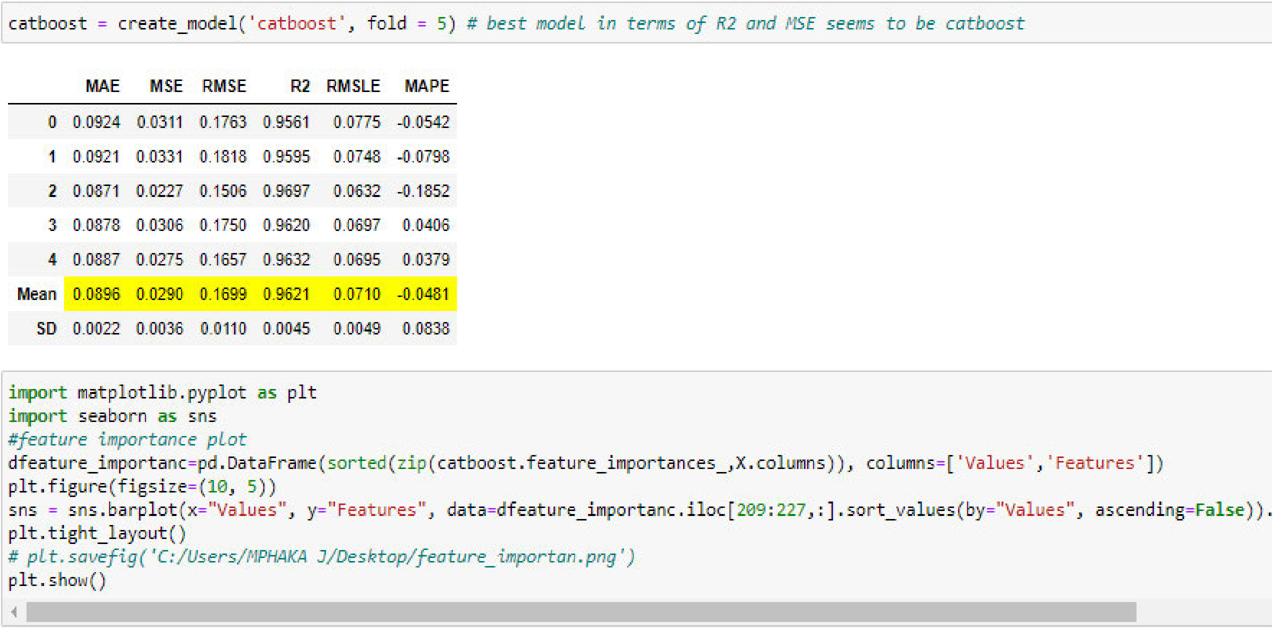
```
from pycaret.regression import * # loading all machine learning models from pycaret
# Here we randomly split our data to get training and test sets. The test set will consist of 30% the data
train_test_split = setup(data = X, target = 'formation_energy_per_atom', session_id=123, train_size=0.70)
```

```
# we selected best three models(n_select 3) and round off to two decimal place
#number of folds are discussed in the previous chapters
compare_models(round=2, fold=10, n_select = 3)# we selected best three models(n_select 3) and round off to
```

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
0	CatBoost Regressor	0.09	0.03	0.17	0.96	0.07	-0.05	34.18
1	Light Gradient Boosting Machine	0.09	0.03	0.17	0.96	0.07	-0.04	2.61
2	Extreme Gradient Boosting	0.10	0.03	0.18	0.96	0.08	-0.05	5.91

```
<catboost.core.CatBoostRegressor at 0x193e2f1da48>
```

Figure B.4: Models performance

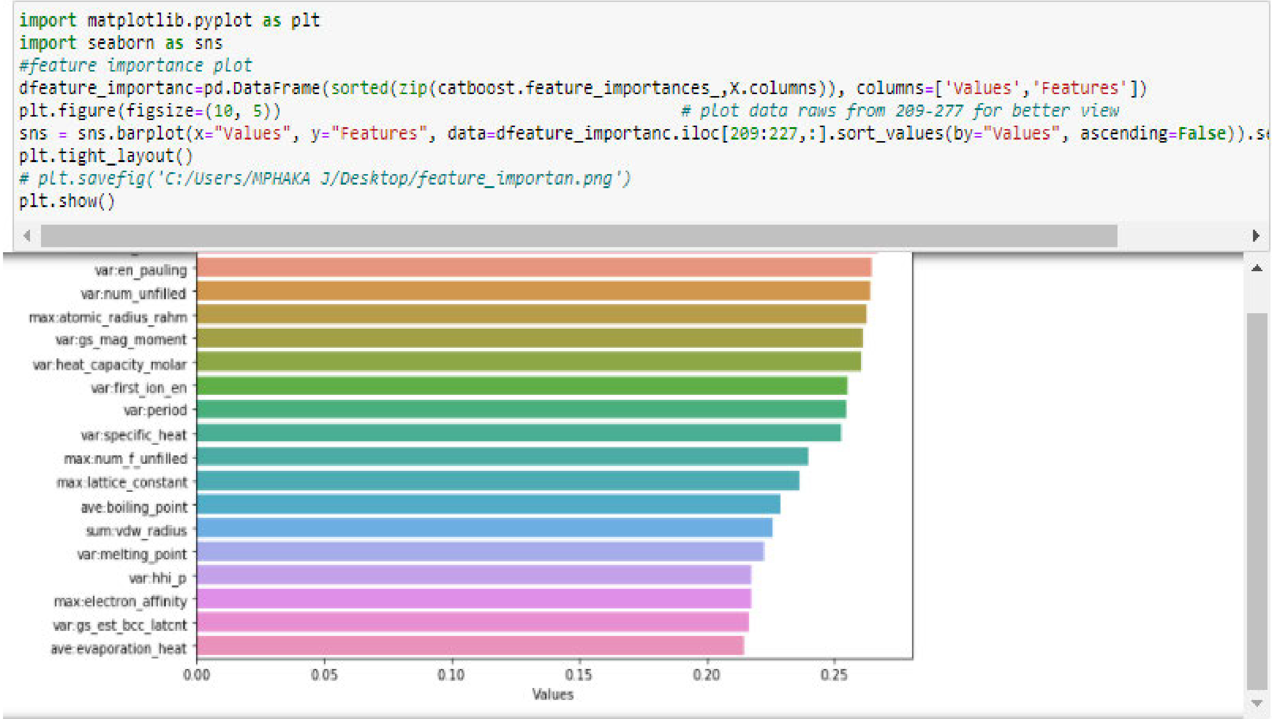


```
catboost = create_model('catboost', fold = 5) # best model in terms of R2 and MSE seems to be catboost
```

	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	0.0924	0.0311	0.1763	0.9561	0.0775	-0.0542
1	0.0921	0.0331	0.1818	0.9595	0.0748	-0.0798
2	0.0871	0.0227	0.1506	0.9697	0.0632	-0.1852
3	0.0878	0.0306	0.1750	0.9620	0.0697	0.0406
4	0.0887	0.0275	0.1657	0.9632	0.0695	0.0379
Mean	0.0896	0.0290	0.1699	0.9621	0.0710	-0.0481
SD	0.0022	0.0036	0.0110	0.0045	0.0049	0.0838

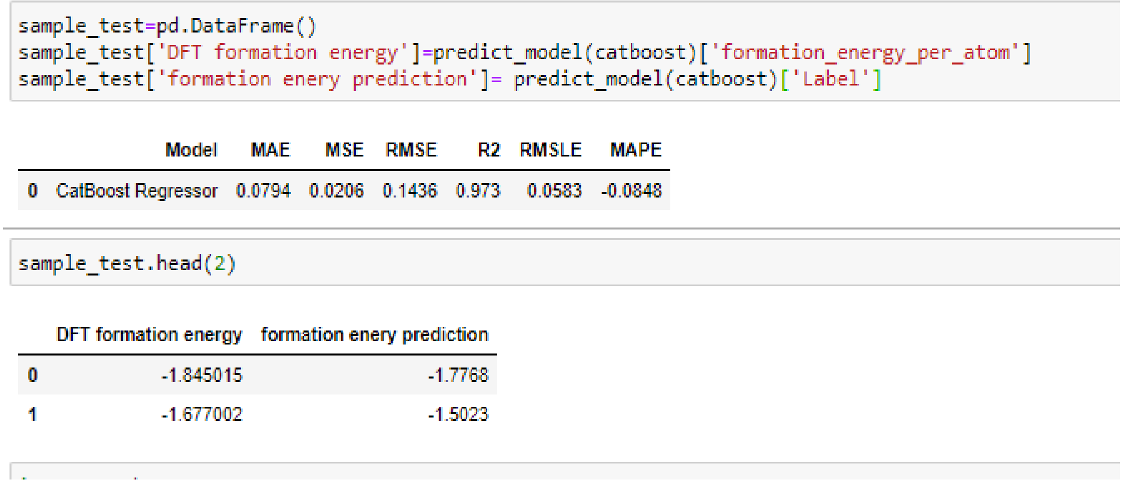
```
import matplotlib.pyplot as plt
import seaborn as sns
#feature importance plot
dfeature_importanc=pd.DataFrame(sorted(zip(catboost.feature_importances_,X.columns)), columns=['Values','Features'])
plt.figure(figsize=(10, 5))
sns = sns.barplot(x="Values", y="Features", data=dfeature_importanc.iloc[209:227,:].sort_values(by="Values", ascending=False)).
plt.tight_layout()
# plt.savefig('C:/Users/MPHAKA J/Desktop/feature_importan.png')
plt.show()
```

Figure B.5: Model build



```
import matplotlib.pyplot as plt
import seaborn as sns
#feature importance plot
dfeature_importanc=pd.DataFrame(sorted(zip(catboost.feature_importances_,X.columns)), columns=['Values','Features'])
plt.figure(figsize=(10, 5))                                    # plot data raws from 209-277 for better view
sns = sns.barplot(x="Values", y="Features", data=dfeature_importanc.iloc[209:227,:].sort_values(by="Values", ascending=False)).se
plt.tight_layout()
# plt.savefig('C:/Users/MPHAKA J/Desktop/feature_importan.png')
plt.show()
```

Figure B.6: Feature importance



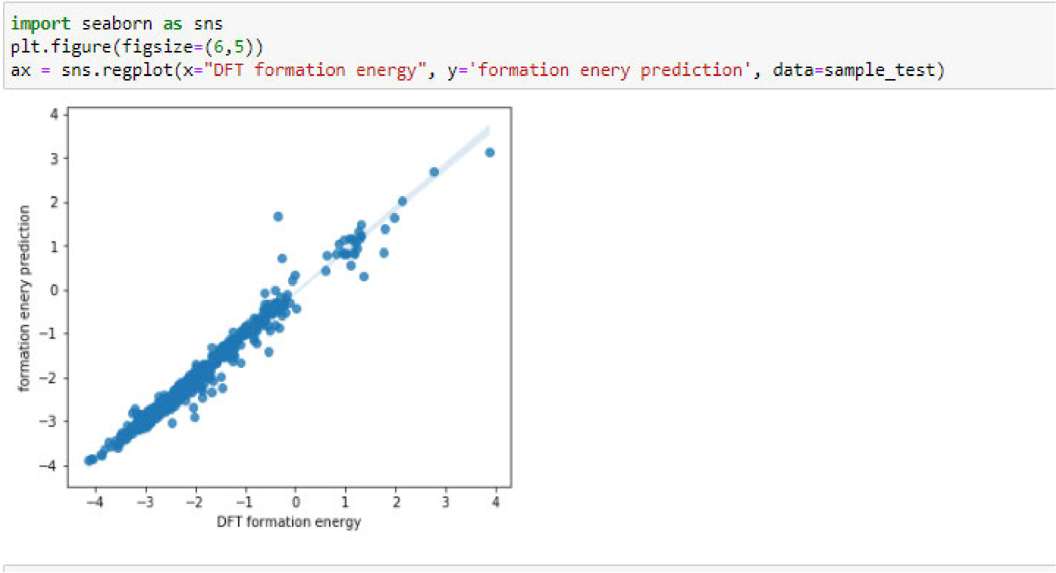
```
sample_test=pd.DataFrame()
sample_test['DFT formation energy']=predict_model(catboost)['formation_energy_per_atom']
sample_test['formation enery prediction']= predict_model(catboost)['Label']
```

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	CatBoost Regressor	0.0794	0.0206	0.1436	0.973	0.0583	-0.0848

```
sample_test.head(2)
```

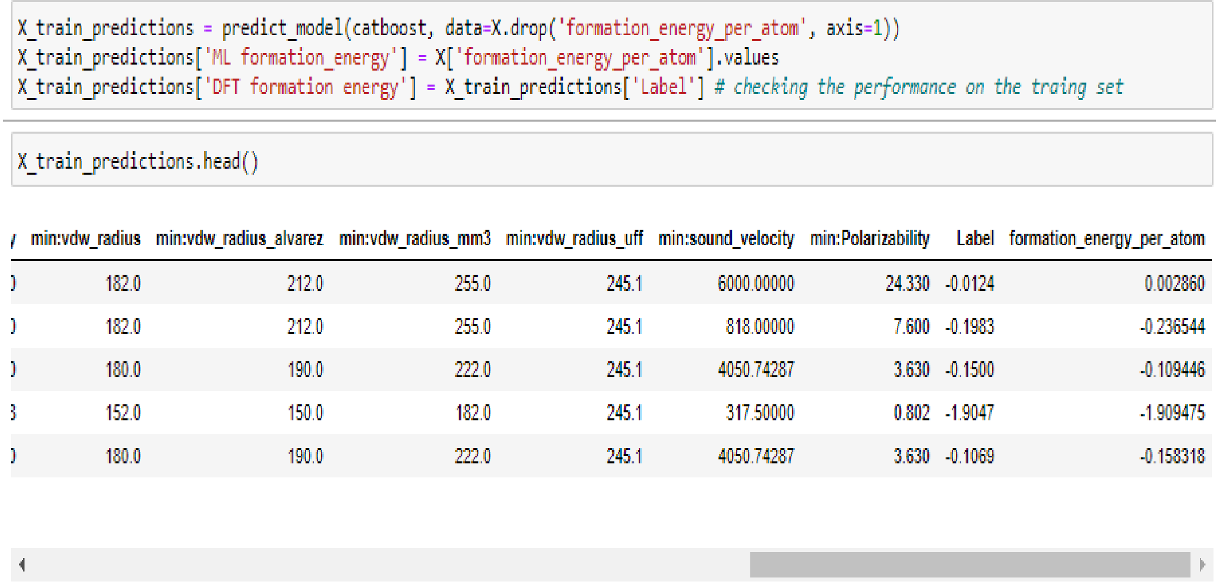
	DFT formation energy	formation enery prediction
0	-1.845015	-1.7768
1	-1.677002	-1.5023

Figure B.7: Performance of catboost model in training set (left) and in testing set (right)



```
import seaborn as sns
plt.figure(figsize=(6,5))
ax = sns.regplot(x="DFT formation energy", y='formation enery prediction', data=sample_test)
```

Figure B.8: Model performance on test data

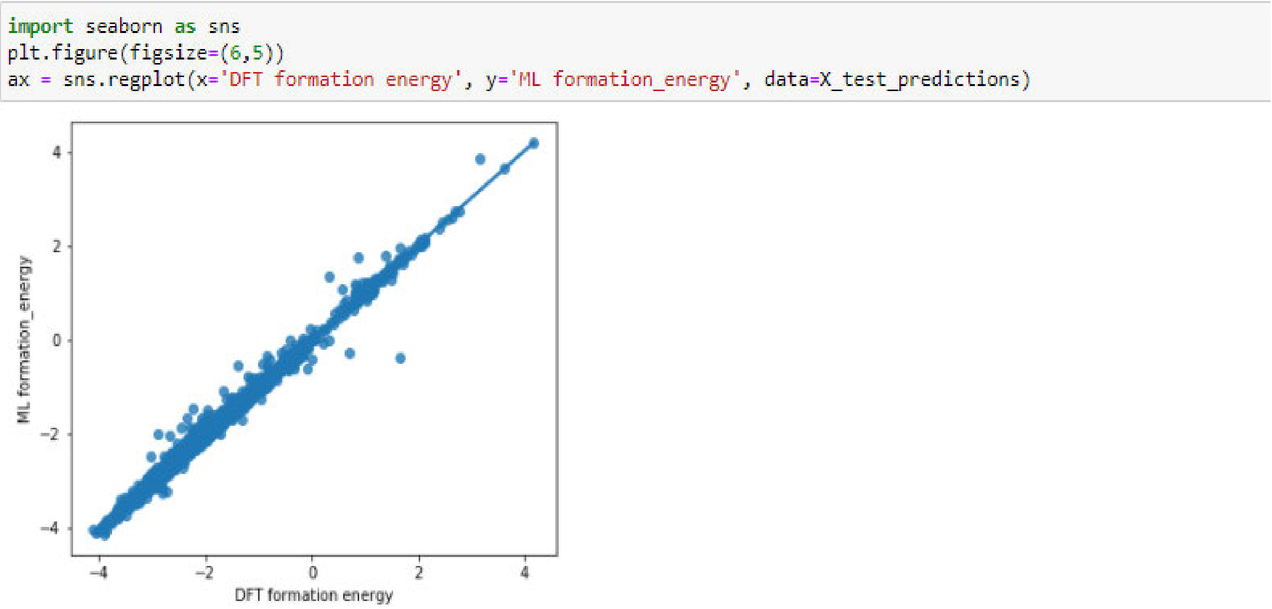


```
X_train_predictions = predict_model(catboost, data=X.drop('formation_energy_per_atom', axis=1))
X_train_predictions['ML formation_energy'] = X['formation_energy_per_atom'].values
X_train_predictions['DFT formation energy'] = X_train_predictions['Label'] # checking the performance on the traing set
```

```
X_train_predictions.head()
```

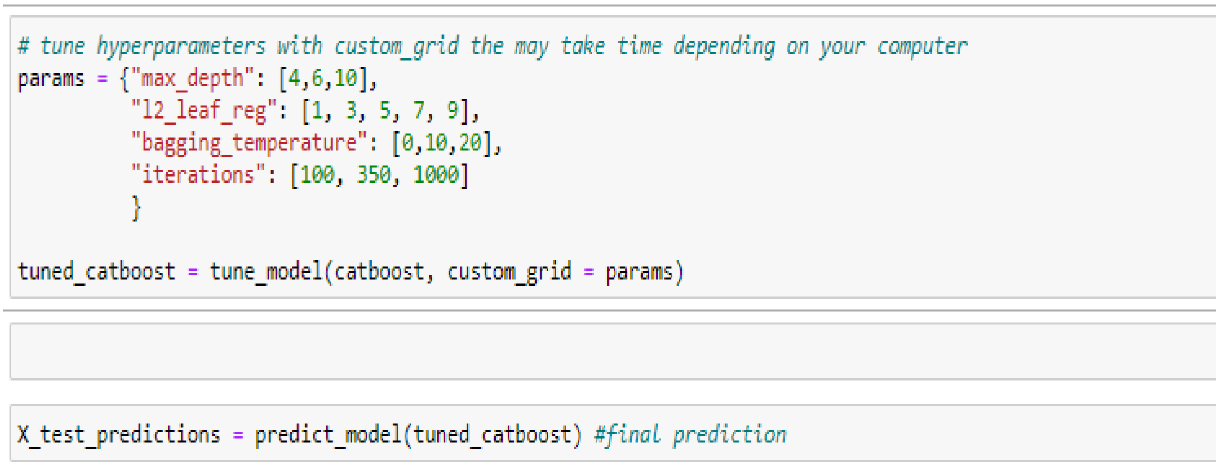
/	min:vdw_radius	min:vdw_radius_alvarez	min:vdw_radius_mm3	min:vdw_radius_uff	min:sound_velocity	min:Polarizability	Label	formation_energy_per_atom
)	182.0	212.0	255.0	245.1	6000.00000	24.330	-0.0124	0.002860
)	182.0	212.0	255.0	245.1	818.00000	7.600	-0.1983	-0.236544
)	180.0	190.0	222.0	245.1	4050.74287	3.630	-0.1500	-0.109446
3	152.0	150.0	182.0	245.1	317.50000	0.802	-1.9047	-1.909475
)	180.0	190.0	222.0	245.1	4050.74287	3.630	-0.1069	-0.158318

Figure B.9: Model predictions on the training data



```
import seaborn as sns
plt.figure(figsize=(6,5))
ax = sns.regplot(x='DFT formation energy', y='ML formation_energy', data=X_test_predictions)
```

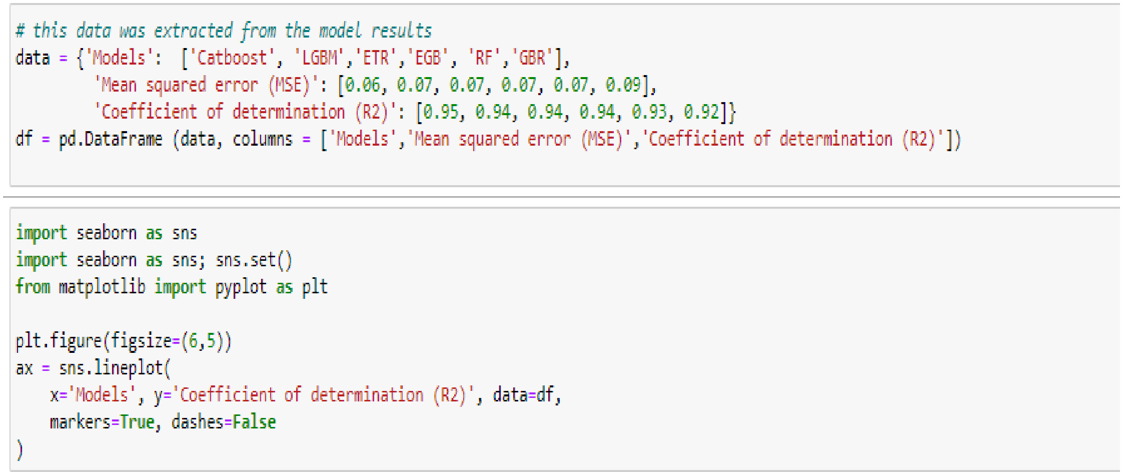
Figure B.10: Model performance



```
# tune hyperparameters with custom_grid the may take time depending on your computer
params = {"max_depth": [4,6,10],
          "l2_leaf_reg": [1, 3, 5, 7, 9],
          "bagging_temperature": [0,10,20],
          "iterations": [100, 350, 1000]
          }

tuned_catboost = tune_model(catboost, custom_grid = params)
```

```
X_test_predictions = predict_model(tuned_catboost) #final prediction
```

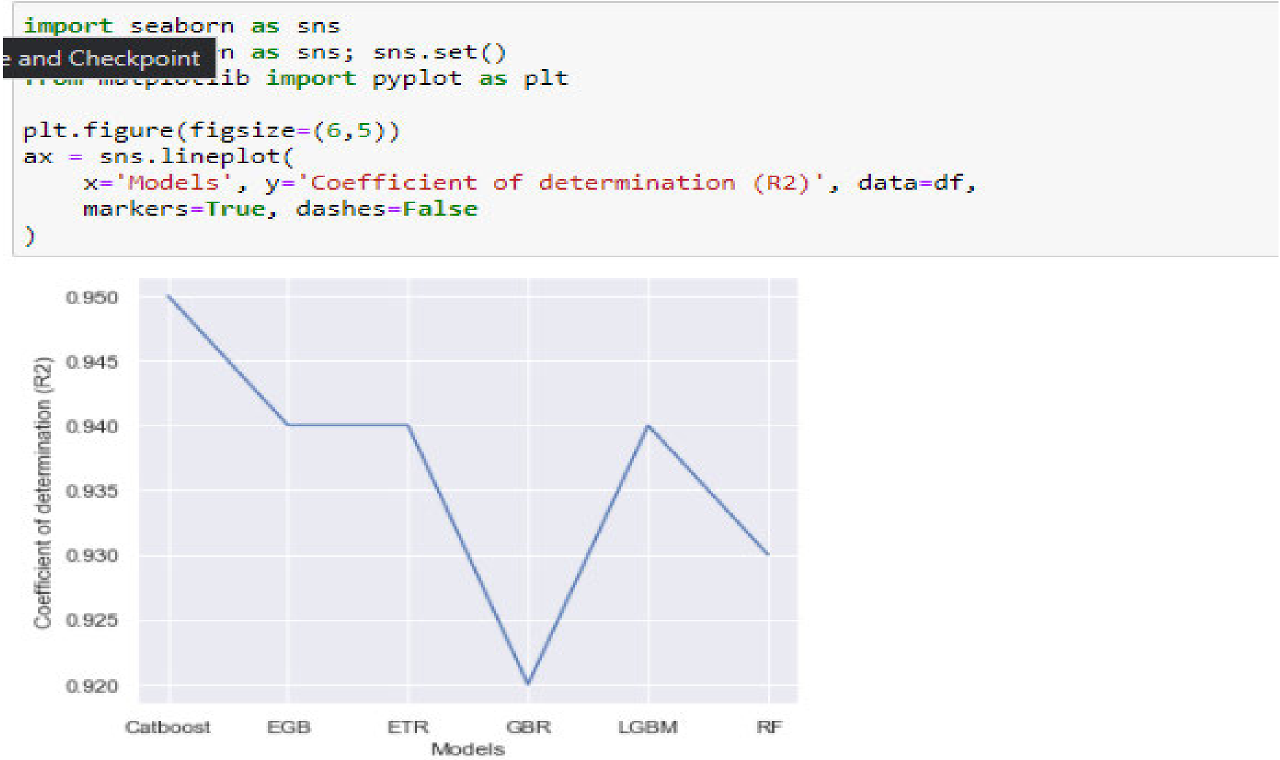
Figure B.11: Model tuning



```
# this data was extracted from the model results
data = {'Models':  ['Catboost', 'LGBM','ETR','EGB', 'RF','GBR'],
        'Mean squared error (MSE)': [0.06, 0.07, 0.07, 0.07, 0.07, 0.09],
        'Coefficient of determination (R2)': [0.95, 0.94, 0.94, 0.94, 0.93, 0.92]}
df = pd.DataFrame (data, columns = ['Models','Mean squared error (MSE)','Coefficient of determination (R2)'])
```

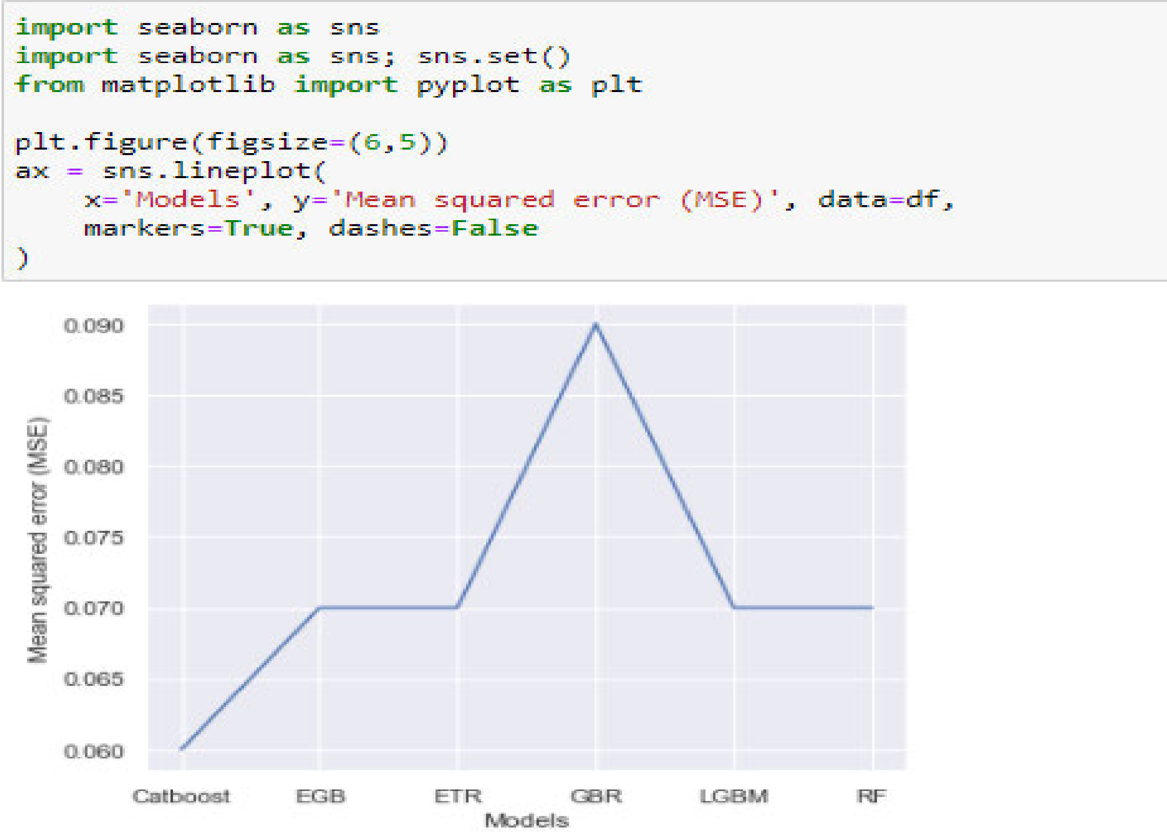
```
import seaborn as sns
import seaborn as sns; sns.set()
from matplotlib import pyplot as plt

plt.figure(figsize=(6,5))
ax = sns.lineplot(
    x='Models', y='Coefficient of determination (R2)', data=df,
    markers=True, dashes=False
)
```

Figure B.12: Model results dataframe



```
import seaborn as sns
e and Checkpoint n as sns; sns.set()
from matplotlib import pyplot as plt

plt.figure(figsize=(6,5))
ax = sns.lineplot(
    x='Models', y='Coefficient of determination (R2)', data=df,
    markers=True, dashes=False
)
```

Figure B.13: Model performance in terms of R^2



```
import seaborn as sns
import seaborn as sns; sns.set()
from matplotlib import pyplot as plt

plt.figure(figsize=(6,5))
ax = sns.lineplot(
    x='Models', y='Mean squared error (MSE)', data=df,
    markers=True, dashes=False
)
```

Figure B.14: Model performance in terms of MSE

Bibliography

[1] Lin Zhu, Ya-Ru Zeng, Jing Wen, Lin Li, and Tai-Min Cheng. Structural and electrochemical properties of Na_2FeSiO_4 polymorphs for sodium-ion batteries. *Electrochimica Acta*, 292:190 198, 2018.

[2] Yu-rong An, Xiao-li Fan, Shi-yao Wang, Zhi-fen Luo, Yan Hu, and Zhen-hai Xia. Pmma-XO (X C, Si, Ge) monolayer as promising anchoring materials for lithium sulfur battery: a first-principles study. *Nanotechnology*, 30(8):085405, 2018.

[3] Rotem Marom, S Francis Amalraj, Nicole Leifer, David Jacob, and Doron Aurbach. A review of advanced and practical lithium battery materials. *Journal of Materials Chemistry*, 21(27):9938 9954, 2011.

[4] Rapela R Maphanga, Tshepiso Mokoena, and Mahlatse Ratsoma. Estimating dft calculated voltage using machine learning regression models. *Materials Today: Proceedings*, 2020.

[5] Gregory B Olson. Designing a new material world. *Science*, 288(5468):993 998, 2000.

[6] Alán Aspuru-Guzik and Kristin Persson. Materials acceleration platform: Accelerating advanced energy materials discovery by integrating high-throughput methods and artificial intelligence. *Mission Innovation*, 2018.

[7] Christopher C Fischer, Kevin J Tibbetts, Dane Morgan, and Gerbrand Ceder. Predicting crystal structure by merging data mining with quantum mechanics. *Nature materials*, 5(8):641 646, 2006.

[8] M Attarian Shandiz and R Gauvin. Application of machine learning methods for the prediction of crystal system of cathode materials in lithium-ion batteries. *Computational Materials Science*, 117:270 278, 2016.

[9] Prasanna V Balachandran, Benjamin Kowalski, Alp Sehirlioglu, and Turab Lookman. Experimental search for high-temperature ferroelectric perovskites guided by two-step machine learning. *Nature Communications*, 9(1):1668, 2018.

[10] Bryce Meredig, Erin Antono, Carena Church, Maxwell Hutchinson, Julia Ling, Sean Paradiso, Ben Blaiszik, Ian Foster, Brenna Gibbons, Jason

Hattrick-Simpers, et al. Can machine learning identify the next high-temperature superconductor? examining extrapolation performance for materials discovery. *Molecular Systems Design & Engineering*, 3(5):819 825, 2018.

[11] Dezhen Xue, Deqing Xue, Ruihao Yuan, Yumei Zhou, Prasanna V Balachandran, Xiangdong Ding, Jun Sun, and Turab Lookman. An informatics approach to transformation temperatures of niti-based shape memory alloys. *Acta Materialia*, 125:532 541, 2017.

[12] Motoaki Nishijima, Takuya Ootani, Yuichi Kamimura, Toshitsugu Sueki, Shogo Esaki, Shunsuke Murai, Koji Fujita, Katsuhisa Tanaka, Koji Ohira, Yukinori Koyama, et al. Accelerated discovery of cathode materials with prolonged cycle life for lithium-ion battery. *Nature Communications*, 5(1):1 7, 2014.

[13] Chuhong Wang, Koutarou Aoyagi, Pandu Wisesa, and Tim Mueller. Lithium ion conduction in cathode coating materials from on-the-fly machine learning. *Chemistry of Materials*, 32(9):3741 3752, 2020.

[14] Languang Lu, Xuebing Han, Jianqiu Li, Jianfeng Hua, and Minggao Ouyang. A review on the key issues for lithium-ion battery management in electric vehicles. *Journal of Power Sources*, 226:272 288, 2013.

[15] J-M Tarascon and Michel Armand. Issues and challenges facing rechargeable lithium batteries. In *Materials for sustainable energy: a collection of peer-reviewed research and review articles from Nature Publishing Group*, pages 171 179. World Scientific, 2011.

[16] Zhi-Hui Xie, Zimin Jiang, and Xueyuan Zhang. Promises and challenges of in situ transmission electron microscopy electrochemical techniques in the studies of lithium ion batteries. *Journal of The Electrochemical Society*, 164(9):A2110, 2017.

[17] Kang Xu. Nonaqueous liquid electrolytes for lithium-based rechargeable batteries. *Chemical Reviews*, 104(10):4303 4418, 2004.

[18] Fredrik Larsson, Petra Andersson, and Bengt-Erik Mellander. Lithium-ion battery aspects on fires in electrified vehicles on the basis of experimental abuse tests. *Batteries*, 2(2):9, 2016.

[19] Yihan Xiao, Lincoln J Miara, Yan Wang, and Gerbrand Ceder. Computational screening of cathode coatings for solid-state batteries. *Joule*, 3(5):1252 1275, 2019.

[20] Kai Liu, Yayuan Liu, Dingchang Lin, Allen Pei, and Yi Cui. Materials for lithium-ion battery safety. *Science Advances*, 4(6):eaas9820, 2018.

[21] Yihan Xiao, Yan Wang, Shou-Hang Bo, Jae Chul Kim, Lincoln J Miara, and Gerbrand Ceder. Understanding interface stability in solid-state batteries. *Nature Reviews Materials*, 5(2):105 126, 2020.

[22] Y-H Chen, C-W Wang, X Zhang, and Ann Marie Sastry. Porous cathode optimization for lithium cells: Ionic and electronic conductivity, capacity, and selection of materials. *Journal of Power Sources*, 195(9):2851 2862, 2010.

[23] Austin D Sendek, Qian Yang, Ekin D Cubuk, Karel-Alexander N Duerloo, Yi Cui, and Evan J Reed. Holistic computational structure screening of more than 12000 candidates for solid lithium-ion conductor materials. *Energy & Environmental Science*, 10(1):306 320, 2017.

[24] Christian M Julien, Alain Mauger, Karim Zaghib, and Henri Groult. Comparative issues of cathode materials for li-ion batteries. *Inorganics*, 2(1):132 154, 2014.

[25] Steven K Kauwe, Trevor David Rhone, and Taylor D Sparks. Data-driven studies of li-ion-battery materials. *Crystals*, 9(1):54, 2019.

[26] Weike Ye, Chi Chen, Zhenbin Wang, Iek-Heng Chu, and Shyue Ping Ong. Deep neural networks for accurate predictions of crystal stability. *Nature Communications*, 9(1):1 6, 2018.

[27] Qiu He, Bin Yu, Zhaohuai Li, and Yan Zhao. Density functional theory for battery materials. *Energy & Environmental Materials*, 2(4):264 279, 2019.

[28] Lei Fang, Xinrui Cao, and Zexing Cao. Covalent organic framework with high capacity for the lithium ion battery anode: insight into intercalation of li from first-principles calculations. *Journal of Physics: Condensed Matter*, 31(20):205502, 2019.

[29] Martin A Green. Intrinsic concentration, effective densities of states, and effective mass in silicon. *Journal of Applied Physics*, 67(6):2944 2954, 1990.

[30] Atsuto Seko, Tomoya Maekawa, Koji Tsuda, and Isao Tanaka. Machine learning with systematic density-functional theory calculations: Application to melting temperatures of single-and binary-component solids. *Physical Review B*, 89(5):054303, 2014.

[31] Henry Wu, Aren Lorenson, Ben Anderson, Liam Witteman, Haotian Wu, Bryce Meredig, and Dane Morgan. Robust fcc solute diffusion predictions from ab-initio machine learning methods. *Computational Materials Science*, 134:160 165, 2017.

[32] Zeyu Liu, Meng Jiang, and Tengfei Luo. Leverage electron properties to predict phonon properties via transfer learning for semiconductors. *Science advances*, 6(45):eabd1356, 2020.

[33] Anjana Talapatra, Shahin Boluki, Thien Duong, Xiaoning Qian, Edward Dougherty, and Raymundo Arróyave. Autonomous efficient experiment design for materials discovery with bayesian model averaging. *Physical Review Materials*, 2(11):113803, 2018.

[34] Daniel P Tabor, Loïc M Roch, Semion K Saikin, Christoph Kreisbeck, Dennis Sheberla, Joseph H Montoya, Shyam Dwaraknath, Muratahan Aykol, Carlos Ortiz, Hermann Tribukait, et al. Accelerating the discovery of materials for clean energy in the era of smart automation. *Nature Reviews Materials*, 3(5):5 20, 2018.

[35] Pavel Nikolaev, Daylond Hooper, Frederick Webber, Rahul Rao, Kevin Decker, Michael Krein, Jason Poleski, Rick Barto, and Benji Maruyama. Autonomy in materials research: a case study in carbon nanotube growth. *npj Computational Materials*, 2(1):1 6, 2016.

[36] Benjamin P MacLeod, Fraser GL Parlane, Thomas D Morrissey, Florian Häse, Loïc M Roch, Kevan E Dettelbach, Raphaell Moreira, Lars PE Yunker, Michael B Rooney, Joseph R Deeth, et al. Self-driving laboratory for accelerated discovery of thin-film materials. *Science Advances*, 6(20):eaaz8867, 2020.

[37] Vasilios Duros, Jonathan Grizou, Weimin Xuan, Zied Hosni, De-Liang Long, Haralampos N Miras, and Leroy Cronin. Human versus robots in the discovery and crystallization of gigantic polyoxometalates. *Angewandte Chemie International Edition*, 56(36):10815 10820, 2017.

[38] Andrew Sparkes, Wayne Aubrey, Emma Byrne, Amanda Clare, Muhammed N Khan, Maria Liakata, Magdalena Markham, Jem Rowland, Larisa N Soldatova, Kenneth E Whelan, et al. Towards robot scientists for autonomous scientific discovery. *Automated Experimentation*, 2(1):1, 2010.

[39] Geoffroy Hautier, Anubhav Jain, and Shyue Ping Ong. From the computer to the laboratory: materials discovery and design using first-principles calculations. *Journal of Materials Science*, 47(21):7317 7340, 2012.

[40] John Maddox. Crystals from first principles. *Nature*, 335(6187):201 201, 1988.

[41] Tim Mueller, Aaron Gilad Kusne, and Rampi Ramprasad. Machine learning in materials science: Recent progress and emerging applications. *Reviews in Computational Chemistry*, 29:186 273, 2016.

[42] Jonathan Schmidt, Mário RG Marques, Silvana Botti, and Miguel AL Marques. Recent advances and applications of machine learning in solid-state materials science. *npj Computational Materials*, 5(1):1 36, 2019.

[43] Geoffroy Hautier, Christopher C Fischer, Anubhav Jain, Tim Mueller, and Gerbrand Ceder. Finding nature's missing ternary oxide compounds using machine learning and density functional theory. *Chemistry of Materials*, 22(12):3762 3767, 2010.

[44] Dominique Luzeaux. Process control and machine learning: Rule-based incremental control. *IEEE transactions on automatic control*, 39(6):1166 1171, 1994.

[45] Danton S Char, Nigam H Shah, and David Magnus. Implementing machine learning in health care addressing ethical challenges. *The New England Journal of Medicine*, 378(11):981, 2018.

[46] László Monostori, András Márkus, Hendrik Van Brussel, and E Westkämpfer. Machine learning approaches to manufacturing. *CIRP annals*, 45(2):675 712, 1996.

[47] Yue Liu, Tianlu Zhao, Wangwei Ju, and Siqi Shi. Materials discovery and design using machine learning. *Journal of Materiomics*, 3(3):159 177, 2017.

[48] Maryam Sabzevari. Ensemble learning in the presence of noise. 2019.

[49] Xindong Wu, Xingquan Zhu, Gong-Qing Wu, and Wei Ding. Data mining with big data. *IEEE Transactions on Knowledge and Data Engineering*, 26(1):97 107, 2013.

[50] Gang-Zhi Fan, Seow Eng Ong, and Hian Chye Koh. Determinants of house price: A decision tree approach. *Urban Studies*, 43(12):2301 2315, 2006.

[51] Steven K Kauwe, Jake Graser, Antonio Vazquez, and Taylor D Sparks. Machine learning prediction of heat capacity for solid inorganics. *Integrating Materials and Manufacturing Innovation*, 7(2):43 51, 2018.

[52] Carl Kingsford and Steven L Salzberg. What are decision trees? *Nature Biotechnology*, 26(9):1011 1013, 2008.

[53] Sreerama K Murthy. Automatic construction of decision trees from data: A multi-disciplinary survey. *Data Mining and Knowledge Discovery*, 2(4):345 389, 1998.

[54] S Rasoul Safavian and David Landgrebe. A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(3):660 674, 1991.

[55] Wei-Yin Loh. Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):14 23, 2011.

[56] J. Ross Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81 106, 1986.

[57] Leo Breiman. Random forests. *Machine Learning*, 45(1):5 32, 2001.

[58] Jeffrey Strickland. *Predictive analytics using R*. Lulu. com, 2015.

[59] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3 42, 2006.

[60] Pierre Geurts and Gilles Louppe. Learning to rank with extremely randomized trees. *Proceedings of Machine Learning Research*, 14:49 61, 2011.

[61] Robert E Schapire. The boosting approach to machine learning: An overview. In *Nonlinear Estimation and Classification*, pages 149 171. Springer, 2003.

[62] Haihao Lu, Sai Praneeth Karimireddy, Natalia Ponomareva, and Vahab Mirrokni. Accelerating gradient boosting machines. In *International Conference on Artificial Intelligence and Statistics*, pages 516 526, 2020.

[63] Haihao Lu and Rahul Mazumder. Randomized gradient boosting machine. *SIAM Journal on Optimization*, 30(4):2780 2808, 2020.

[64] Leo Guelman. Gradient boosting trees for auto insurance loss cost modeling and prediction. *Expert Systems with Applications*, 39(3):3659 3667, 2012.

[65] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, and Yuan Tang. Xgboost: extreme gradient boosting. *R Package Version 0.4-2*, pages 1 4, 2015.

[66] Fei Li, Li Zhang, Bin Chen, Dianzhu Gao, Yijun Cheng, Xiaoyong Zhang, Yingze Yang, Kai Gao, Zhiwu Huang, and Jun Peng. A light gradient boosting machine for remainning useful life estimation of aircraft engines. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3562 3567. IEEE, 2018.

[67] Jason Brownlee. Machine learning mastery with python. *Machine Learning Mastery Pty Ltd*, pages 100 120, 2016.

[68] Jason Brownlee. *XGBoost With Python: Gradient Boosted Trees with XGBoost and scikit-learn*. Machine Learning Mastery, 2016.

[69] Franziska Horn, Robert Pack, and Michael Rieger. The autofeat python library for automated feature engineering and selection. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 111 120. Springer, 2019.

[70] Abhishek Thakur. *Approaching (almost) any machine learning problem*. Abhishek Thakur, 2020.

[71] Jessica Vamathevan, Dominic Clark, Paul Czodrowski, Ian Dunham, Edgardo Ferran, George Lee, Bin Li, Anant Madabhushi, Parantu Shah, Michaela Spitzer, et al. Applications of machine learning in drug discovery and development. *Nature Reviews Drug Discovery*, 18(6):463 477, 2019.

[72] Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, et al. Commentary: The materials project: A materials genome approach to accelerating materials innovation. *Applied Materials*, 1(1):011002, 2013.

[73] Shyue Ping Ong, William Davidson Richards, Anubhav Jain, Geoffroy Hautier, Michael Kocher, Shreyas Cholia, Dan Gunter, Vincent L Chevrier, Kristin A Persson, and Gerbrand Ceder. Python materials genomics (pymatgen): A robust, open-source python library for materials analysis. *Computational Materials Science*, 68:314 319, 2013.

[74] Minoru Kusaba, Chang Liu, Yukinori Koyama, Kiyoyuki Terakura, and Ryo Yoshida. Recreation of the periodic table with an unsupervised machine learning algorithm. *arXiv preprint arXiv:1912.10708*, 2019.

[75] N Yandex. Catboost is a high-performance open source library for gradient boosting on decision trees. *https: // catboost. ai/ docs/ concepts/ parameter-tuning. html* , 2017.

[76] Adelaide M Nolan, Yunsheng Liu, and Yifei Mo. Solid-state chemistries stable with high-energy cathodes for lithium-ion batteries. *ACS Energy Letters*, 4(10):2444 2451, 2019.

[77] T Chen. Xgboost documentation. *https: // xgboost. readthedocs. io/ en/ latest/ parameter. html* , 2014.