Predicting the Thermodynamic Stability of Perovskite Oxides Using Machine Learning Models

Siddharth Sawant MM22B001, Rajat Maliwal MM22B050, Karthikayini S MM22B035, Vidit M Jain MM22B068, Arun Barwa MM22B019

Abstract

Perovskite oxides have emerged as a promising and progressively favored material for solar cell technology due to their remarkable efficiency, cost-effectiveness, and scalability. Our team delves into the realm of predicting the thermodynamic stability of perovskite oxides through the application of machine learning. Our objective centers on employing machine learning algorithms to forecast the stability of perovskite oxides, using computed data derived from Density Functional Theory. The dataset utilized for our analysis originates from Kaggle.

1. Introduction

Perovskites with their varied composition,controllable structure and affordability are gaining popularity in catalysis and thermoelectrics.Some perovskites hold promise for solar cells and LEDs because of their tunable properties.To understand these materials better, researchers use high-throughput methods(DFT) and databases like OPTIMADE to gain a deeper understanding of these materials but DFT is computationally expensive.Therefore researchers are now bending towards machine learning which offers a faster way to predict material stability without needing tons of experiments.[3]

In this project we are trying to find the thermodynamic stability of these materials to get a better insight on material reliability, safety, and performance across a wide range of applications.For this purpose we are using the already available DFT data as a training dataset.We are using various ML models like Linear Regression , Neural Networks,XGBoost to determine which model gives the best output.The underlying principle behind this is energy above the convex hull energy.If the Ehull$<$40 meV/atom then it is stable and unstable if Ehull>40 meV/atom.

2. Material Science

Perovskite oxides are ternary oxides of general formula ABO3 and the same crystallographic structure as perovskite mineral (CaTiO3). We can describe the perovskite oxide lattice arrangement as a large atomic or molecular cation (positively charged) of type A in the center of a cube. The corners of the cube are then occupied by atoms B (also positively charged cations) and the faces of the cube are occupied by a smaller oxygen atom, O with negative charge (anion). The A ions are typically large ions such as Sr2+, Ba2+, Rb+, or a lanthanide 3+ ion, and the B ions are smaller transition metal ions such as Ti4+, Nb5+, Ru4+, etc. The perovskite structure has simple cubic symmetry, but is related to the fcc lattice in the sense that the A site cations and the three O atoms comprise a fcc lattice. The B-site cations fill 1/4 of the octahedral holes and are surrounded by six oxide anions.[1]

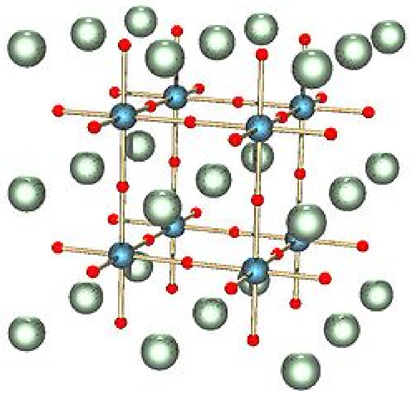


Figure 1: ABX3 perovskite structure. A, B, and X are white, blue, and red, respectively.

Perovskites can exhibit a wide range of stoichiometry (where A and B are one or more cations). Hence it is challenging to predict the thermodynamic stability of these compounds. E.g. BaFe0.25V0.75O3.

Basically, a convex hull is a plot of formation energy with respect to the composition which connects phases that are lower in energy than any other phases at that composition. Phases that lie on the convex hull are thermodynamically stable and the ones above it are either metastable or unstable.

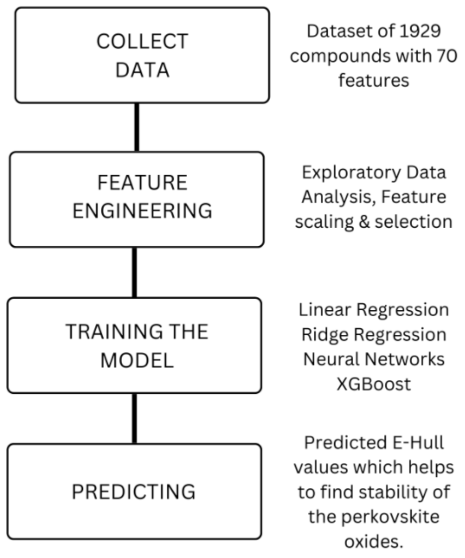
Formation energy is the energy with respect to some reference states. This is used for example with DFT results, since the absolute energy output by the program is not very useful without a reference. Energy above the hull is referring to the distance above the convex hull, which is formed by the lowest calculated states. In DFT, anything on the convex hull is considered a ground state, i.e. it is a stable state at OK. For a simple binary system, you can calculate the formation energy by $E_f = E - (1-x)E_1 - xE_2$ where E is the energy of the material and E_1 is the energy of the reference at $x = 0$ composition and E_2 is the energy of the reference at $x = 1$ composition. To get the energy above the hull, you have to know what the nearby stable phases are.[2]

3. Methods

The development and validation of our machine learning models for predicting perovskite stability comprised four key stages: (i)Collecting data based on the thermodynamic characteristics of perovskite oxides. (ii) Identification of pertinent features exhibiting strong correlations with stability through feature selection and engineering. (iii) Selection of the optimal machine learning model from a pool of 4 Machine learning models. (iv) Forecasting thermal stability for perovskites beyond the dataset and comparing these predictions against DFT calculations. In subsequent sections, we elaborate on each of these steps essential to constructing our machine learning models.

In this study, we used the Python library scikit-learn for all machine learning models, feature selection techniques, and model assessments. Scikit-learn is an open-source machine learning toolkit distributed under the BSD license. A comprehensive summary of all scikit-learn methods and function calls utilized in this project is mentioned in this report in Brief . The training dataset comprising compositions of perovskite oxides and corresponding DFT-calculated Ehull values, along with the project source code and optimal models, are also made available in the report in Brief.

4. Pipeline


COLLECT DATA
Dataset of 1929 compounds with 70 features
FEATURE ENGINEERING
Exploratory Data Analysis, Feature scaling & selection
TRAINING THE MODEL
Linear Regression
Ridge Regression
Neural Networks
XGBoost
PREDICTING
Predicted E-Hull values which helps to find stability of the perkovskite oxides.

5. Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is a crucial initial step in any data science project. It involves a comprehensive examination of the data to understand its characteristics, identify patterns, and assess its suitability for modeling. During EDA, we focus on the features (columns) within the dataset and their potential contribution to the desired outcome.

In this project, we employed the pandas-profiling library to conduct a detailed analysis of the data. This library provided a wealth of information, including:

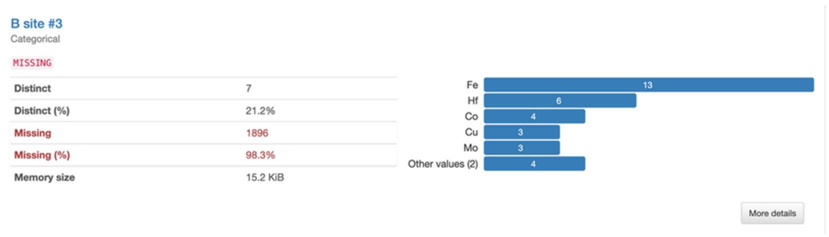
- Total Count: The number of data points present in each column.
- Frequency: The distribution of unique values within each column.
- Mean and Median: Measures of central tendency for numerical columns.

- Variance: A statistical measure of how spread out the data is for numerical columns.

- Empty Values: The number of missing entries within each column.

Through careful analysis of these metrics, we identified and addressed two key data quality issues: missing values and low variance features.

1. Addressing Missing Values: We encountered columns like ”B-site#3” that contained a substantial amount of missing data. To ensure our model is trained on complete and reliable information, we opted to remove such columns entirely. This approach safeguards the model from the potential inaccuracies introduced by missing data imputation techniques.

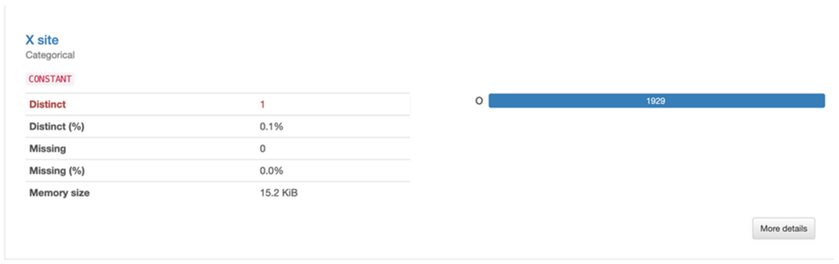


B site #3
Categorical
MISSING

Distinct	7
Distinct (%)	21.2%
Missing	1896
Missing (%)	98.3%
Memory size	15.2 KiB

Figure 2: Missing data in Column “B site #3”

2. Eliminating Features with Low Variance: Features with minimal variation, such as ”x-site”, were identified. These features hold minimal predictive power because the values are nearly identical across most, if not all, data points. Including such features in the model can introduce noise and potentially hinder its performance. Consequently, we made the decision to remove features exhibiting low variance.
3. Feature Distribution and Range Analysis: Beyond addressing missing data and low variance, we delved deeper into the characteristics of each remaining feature. This involved analyzing the distribution (e.g., normal, skewed) and range of values for features like ”E-hull”. This analysis provided valuable insights into the spread and potential relationships between these features and the target variable.



X site

Categorical

CONSTANT

Distinct	1
Distinct (%)	0.1%
Missing	0
Missing (%)	0.0%
Memory size	15.2 KiB

Figure 3: Low Variance of data in the column “X site”

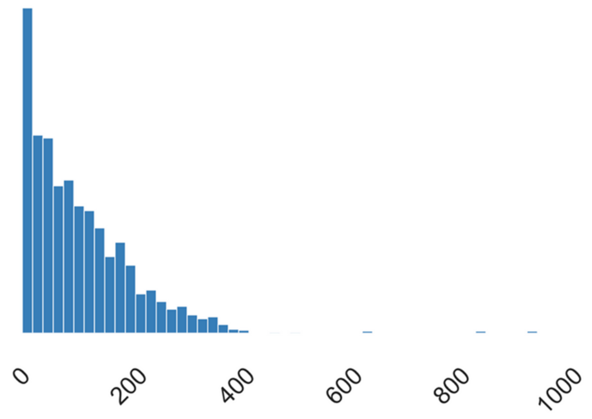


Figure 4: Distribution of E-hull values

6. Feature Engineering: Correlation Analysis and Scaling

Following data cleaning and initial feature selection, we sought to further optimize our data for model training. This involved two key techniques: based on correlation and feature scaling.

1. **Correlation analysis:** Correlation analysis helps visualize the relationships between features within a dataset. A correlation matrix depicts these relationships numerically, with values close to 1 or -1 indicating strong positive or negative correlations, respectively. Conversely, values close to 0 suggest minimal to no correlation between features.

   To identify and eliminate redundant features, we implemented code that automatically detects features with a correlation coefficient exceeding 0.95. These highly correlated features likely contain overlapping information, potentially hindering model performance. Removing such redundant features simplifies the model and potentially improves its efficiency and generalizability.

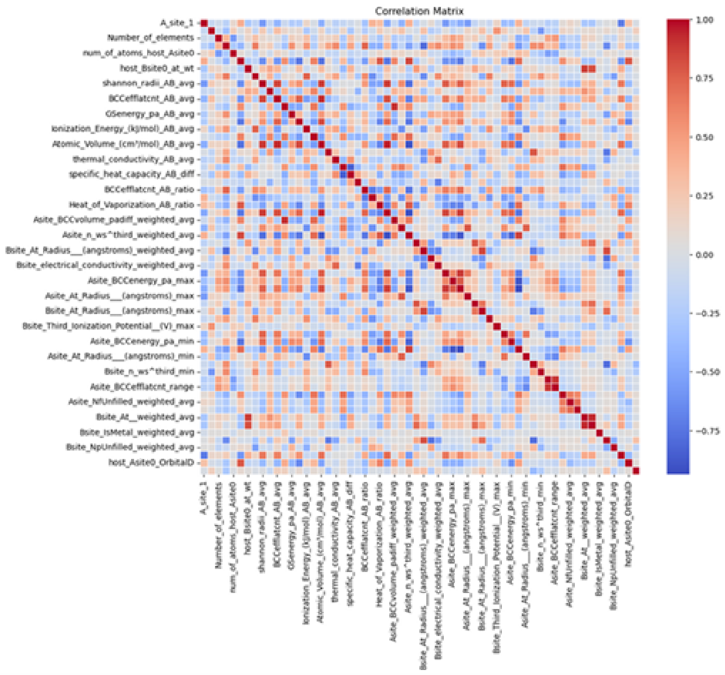


Figure 5: Correlation Matrix

Furthermore, to verify the redundancy of removed features, we analyzed the adjusted R-squared score after each removal. This metric helps assess the model's explanatory power, and a negligible change in R-squared after feature removal reinforces the redundancy of the eliminated feature.

2. **Feature Scaling:** Feature scaling plays a crucial role in optimizing data for machine learning models. We employed a standardization technique to scale the features within our dataset. Standardization transforms the features to have a mean of 0 and a standard deviation of 1. This ensures all features contribute equally to the model's learning process, preventing features with larger ranges from dominating the

training phase.

The effectiveness of feature scaling is evident when comparing the

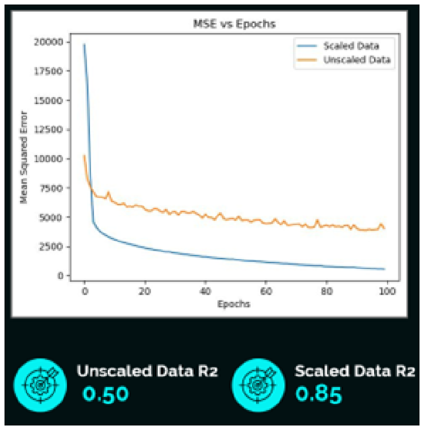


Figure 6: R2 score of unscaled and scaled data

performance of models trained on scaled versus unscaled data. As illustrated in the provided graph, a neural network model trained on scaled data converges significantly faster and achieves a lower overall Mean Squared Error (MSE) compared to the model trained on unscaled data. This demonstrates how scaling levels the playing field for features, leading to faster training and potentially improved model performance.

By implementing these feature engineering techniques, we ensure our model leverages the most informative and efficient data representation, ultimately contributing to a robust and accurate model.

7. Model Training and Evaluation

7.1. Linear Regression

7.1.1. Data Splitting:

To evaluate the performance of our chosen model, we used the train_test_split function from the scikit-learn library to partition the preprocessed data into two distinct sets: a training set and a testing set in the ratio 80:20.

7.1.2. Model:

We implemented a linear regression model to uncover the underlying relationship between the independent and dependent variables within our dataset. Linear regression establishes a linear equation to best approximate the relationship between these variables. The model is trained using the training data, allowing it to learn the coefficients of the linear equation.

7.1.3. Evaluation:

The model achieved an RMSE of 49.8meV/atom. The R^2 score of 0.77 suggests that the model explains 77% of the variance in the dependent variable. While these results show promise, further exploration of models or feature engineering might be beneficial.

7.2. Ridge Regression

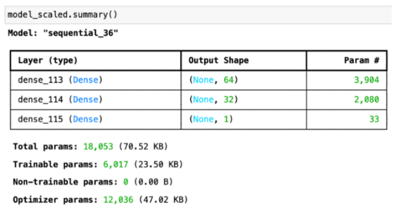
This model performs ridge regression using cross-validation to evaluate the model's performance. For each fold of the cross-validation, the training and validation datasets are created. The Ridge regression model (pr_l2—) is fitted to the training data. Predictions are made on both the training and validation sets. The evaluation metrics, including root mean squared error (RMSE) and coefficient of determination (R2), are calculated for each fold. When evaluated on an unseen test set we got the RMSE as 44.1 meV/atm and R2 score as 0.80.

7.3. ANN

Utilizing TensorFlow and Keras, we constructed a neural network regression model to predict Ehull values. Our model comprises three dense layers with 64, 32, and 1 units respectively, organized in a Sequential structure. ReLU activation functions are employed for the hidden layers, while a linear activation function is applied to the output layer.

For optimization, we compiled the model using the Adam optimizer, employing mean squared error (MSE) as the loss function and a metric during training. Training took place over 100 epochs, utilizing a batch size of 32 and incorporating a validation split of 20%.

To assess model performance, we evaluated it on an unseen test set, calculating Mean Squared Error (MSE) of 39.25 meV/atoms and Coefficient of Determination (R2) of 0.856 as evaluation metrics.



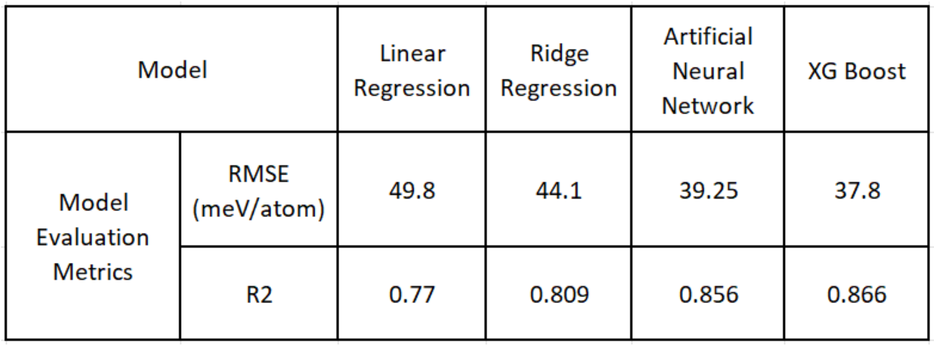
```
model_scaled.summary()
Model: "sequential_36"
┌──────────────────────┬───────────────┬──────────┐
│ Layer (type)         │ Output Shape  │  Param # │
├──────────────────────┼───────────────┼──────────┤
│ dense_113 (Dense)    │ (None, 64)    │    3,904 │
├──────────────────────┼───────────────┼──────────┤
│ dense_114 (Dense)    │ (None, 32)    │    2,080 │
├──────────────────────┼───────────────┼──────────┤
│ dense_115 (Dense)    │ (None, 1)     │       33 │
└──────────────────────┴───────────────┴──────────┘
 Total params: 18,053 (70.52 KB)
 Trainable params: 6,017 (23.50 KB)
 Non-trainable params: 0 (0.00 B)
 Optimizer params: 12,036 (47.02 KB)
```

Figure 7: ANN model summary

7.4. XG Boost

XGBoost is a powerful machine learning algorithm known for its effectiveness in various tasks, including regression problems like ours. Using XGBoost we achieved an impressive R-squared value of 0.866. We also achieved a Root Mean Squared Error (RMSE) of 37.8 meV/atom.Compared to other models we explored, XGBoost delivered the best overall performance.

7.5. Comparison of Model



Model		Linear Regression	Ridge Regression	Artificial Neural Network	XG Boost
Model Evaluation Metrics	RMSE (meV/atom)	49.8	44.1	39.25	37.8
	R2	0.77	0.809	0.856	0.866

Figure 8: Comparison of R2 and RMSE values between four models for prediction of Ehull.

8. Areas of Improvement

In this project, we achieved an R2 score of 0.866 ,however we find this performance to be unsatisfactory. To enhance our model's accuracy and efficiency, we have identified several strategies for improvement.

- Hyperparameters play a crucial role in determining the learning dynamics and convergence of neural networks.So optimizing them is a wise choice if someone wants to improve model accuracy. Among them, learning rate and batch size are highly influential. Learning rate controls the step size during gradient descent and greatly impacts the training process. Too high a learning rate may cause overshooting, whereas too low a rate can lead to slow convergence. Batch size affects the stability and speed of convergence during training. Larger batch sizes can provide computational efficiency but might lead to poorer generalization. Conversely, smaller batches introduce more noise into the gradient estimates but can yield better generalization.

- Thearchitecture of a neural network significantly impacts its capacity to learn and generalize from data. Experimenting with various architectures, including the number of layers, types of activation functions and dropout rates is crucial for achieving optimal performance.

- PCAisadimensionality reduction technique used to simplify high-dimensional datasets by transforming them into a lower-dimensional space while preserving the most critical information. In the context of neural network training, PCA can be applied as a preprocessing step to reduce input dimensionality and remove redundant or less informative features. This not only speeds up training but also helps in reducing overfitting by focusing on the most relevant features. It is five step process:

    1. Step-1: The aim is to standardize the range of the continuous initial variables so that each one of them contributes equally to the analysis.
    2. Step-2: The aim of this step is to understand how the variables of the input data set are varying from the mean with respect to each other, or in other words, to see if there is any relationship between them.To do this we create the covariance matrix.
    3. Step-3: In this step we compute the eigenvectors and eigenvalues of the covariance matrix to identify the principal components.
    4. Step-4: In this step we choose whether to keep all these components or discard those of lesser significance (low eigenvalues) and form with the remaining ones a matrix of vectors that we call Feature vector. So, the Feature vector is simply a matrix that has

as columns the eigenvectors of the components that we decide to keep.

5. Step-5: In this step we aim to use the feature vector formed using the eigenvectors of the covariance matrix, to reorient the data from the original axes to the ones represented by the principal components.This can be done by multiplying the transpose of the original data set by the transpose of the feature vector.

Through these strategies we hope to achieve a higher R2 score than we previously achieved.

9. Conclusions

In this project, we used machine learning techniques to forecast the stability of perovskite oxides. Our machine learning models were trained using a DFT-derived dataset containing a lot of compounds, demonstrating promising capabilities in predicting Ehull values with precision comparable to typical DFT calculation margins. Weused 70 features that enabled optimal stability predictions without encountering significant overfitting.
Among five candidate models evaluated via cross-validation, we discovered XGboost regression as the top performer for regression tasks. The highest attained R2 score for classification reached 0.866 , while the best RMSE value for Ehull regression stood at 37.8 meV/atom.
Furthermore, we are planning to apply our models to newly formulated perovskite compounds and validate their predictions against DFT calculations. The models will demonstrate accurate forecasts, particularly for compounds featuring elements commonly represented in the training dataset. Our models offer significant potential for efficiently screening new candidate materials across vast composition spaces using machine learning methodologies.

References

[1] https://chem.libretexts.org/Courses/Douglas$_{C}ollege/DC$

[2] http://docs.materialsatlas.org/user-guide/formation$_{e}nergy_{e}hull/$.

[3] Li, W., Jacobs, R., and Morgan, D. Predicting the thermodynamic stability of perovskite oxides using machine learning models. *Computational Materials Science 150* (2018), 454–463.