Duke University

ME490 – Fall 2025

End-Term Project

Applying ML Techniques to Forecast Material Elasticity Using Composition and Symmetry-based Features

Jaivir Parmar

Due December 10, 2025

Contents

1 Introduction

1.1 Motivation

The elastic modulus is a fundamental mechanical property that determines how materials respond to applied stress and strain. Obtaining accurate modulus values typically requires resource-intensive experimental testing or computationally expensive density functional theory (DFT) simulations. These limitations make the evaluation of large chemical spaces or screening of novel materials for structural applications particularly complicated. A number of engineering industries, ranging from aerospace to energy to advanced manufacturing, show steady growth in demand for high performance materials, motivating the need for faster, data-driven approaches to mechanical property estimations without expensive experimental testing or DFT simulations.

Developments in materials informatics suggest that compositional and symmetry based descriptors may contain enough information to approximate underlying patterns in mechanical stiffness. If true, this suggests that the elastic modulus could be predicted using machine learning techniques on readily accessible descriptor data, significantly reducing the computational cost of large-scale materials discovery.

1.2 Background and Importance of Elastic Modulus

In crystalline solids, the elastic modulus property is fundamentally controlled by atomic bonding strength, crystal symmetry, and micro-structural features. Engineers often rely on accurate elastic modulus estimations to confirm that materials used in load-bearing applications can withstand mechanical loads. The elastic modulus is often used to inform material selection given certain performance requirements. For example, high modulus materials are well suited for stiffness-critical applications (like aerospace panels or cutting tools), whereas lower modulus materials can be selected where flexibility is required (such as polymers or biomedical implants). The rapid growth of materials databases (such as the Materials Project API used here) allows for data-driven approaches for learning relationships between structures and properties. Machine learning algorithms have already shown great promise in their ability to predict band gaps, formation energies, and elasticity using compositional and structural descriptors.

Investigating the ability of machine learning algorithms to use simpler descriptors (such as elemental compositions and crystal system labels) to successfully approximate the elastic modulus could extend ML models to systems where structural data is incomplete or unknown.

1.3 Prior Work in Materials Informatics

Machine learning has become a powerful tool for predicting materials properties, enabled by large open databases such as the Materials Project [1] and feature engineering toolkits like Matminer [2]. Ward et al. demonstrated that compositional descriptors, particularly the Magpie feature set, can accurately predict a broad range of inorganic properties using traditional machine learning models [3]. De Jong et al. provided one of the first large, systematically computed datasets of elastic tensors, enabling data-driven studies of stiffness and related mechanical properties [4]. Benchmark evaluations such as Matbench [5] have shown that tree-based ensemble models often outperform shallow neural networks when trained on composition-only descriptors. More recent work introduced structure-aware deep learning models such as the Crystal Graph Convolutional Neural Network (CGCNN) [6] and the MEGNet architecture [7], both of which achieve higher accuracy for structure-dependent properties by learning directly from atomic coordinates and bonding environments. These studies collectively highlight the strengths of engineered descriptors for baseline prediction and the importance of richer structural representations for accurately modeling elastic moduli.

1.4 Research Question and Hypothesis

The experiment poses the following research question: **”Can elastic modulus be accurately predicted from compositional and crystal-symmetry descriptors without explicit structural information?”** This report hypothesizes that the composition and crystal symmetry features contain enough meaningful data related to stiffness to provide sufficient information to capture major trends in the elastic modulus. It is further hypothesized that machine learning models, particularly Random Forest and Multilayer Perceptron models, are capable of approximating the elastic modulus without full structural descriptors. To this end, the model is expected to have a strong predictive accuracy of $R^2 > 0.8$.

2 Problem Definition

2.1 Objective

The objective of this project is to develop and evaluate machine learning models that can predict the elastic modulus of crystalline materials using only composition and symmetry-based descriptors. This capability enables screening thousands of candidate materials before expensive DFT or experimental characterization, accelerating materials discovery pipelines.

2.2 Scope of the Study

This study focuses on predicting the elastic modulus of inorganic crystalline materials. All data is derived from the Materials Project elasticity dataset via an API call. Features from the dataset are limited to compositional descriptors and crystal-symmetry labels only, which are parsed for each respective material. In the interest of focusing on simplistic features, detailed structural data is deliberately excluded. The performance of two machine learning models, Random Forest and Multilayer Perceptron (MLP), are individually trained, benchmarked, and compared. Interpretability is emphasized (particularly with Random Forest) in the identification of which elemental or symmetry features matter the most.

2.3 Key Assumptions and Limitations

This experiment make several core assumptions in its construction. Its core assumption is that compositional and symmetry descriptors contain enough information to approximate stiffness trends. It also assumes that the Materials Project database contains accurate information that is representative of true material properties. Further, an assumption is made that preprocessing steps do not bias the dataset and that both models are sufficiently expressive for this prediction task. Finally, the experiment assumes that train/test splits and cross-validation properly generalize the performance of the models.

Several limitations are presented within the scope of this experiment. The core limitation imposed by the design of the study is that the lack of explicit structural information may cap accuracy considerably. Additionally, the dataset size may limit the model's ability to learn rare chemistries. There may be many more models with higher viability for this task which were not considered. Since this study is interested in crystalline structures, the results may not generalize to non-crystalline systems. Lastly, these estimations are not confirmed by experimental validation to verify the prediction accuracy beyond DFT.

3 Dataset Curation

3.1 Data Source: Materials Project Elasticity Dataset

This project uses the Materials Project (MP) elasticity dataset for training, which provides a list of DFT-computed elasticity values for 10,994 crystalline materials. It also includes structural info such as IDs, formulas, symmetry, density, and volume, along with full elastic tensors and derived properties. Compositional data, such as element fractions and reduced formulas are also included. This study aims to predict VRH-averaged Young's modulus.

3.2 Data Acquisition via API

Data is retrieved using a free API key generated at the Materials Project webpage with MPRester. Two endpoints are used: summary for structural and compositional metadata and elasticity for elastic tensor and moduli. Data is requested in batches of 1,000 materials per request to avoid usage limits. The summmary and elasticity data is merged based on material escaped_id using an inner join for each material, producing a combined CSV of 264 columns with a flattened JSON field. Only materials with full elasticity data are used.

3.3 Preprocessing Steps

1. The target variable, `youngs_modulus_vrh`, was extracted from the elasticity data. In cases where this value was not directly available, it was computed using the Voigt-Reuss-Hill (VRH) relationship:

   $$E = \frac{9B_{\mathrm{VRH}} \cdot G_{\mathrm{VRH}}}{3B_{\mathrm{VRH}} + G_{\mathrm{VRH}}},$$

   where B_{VRH} is the VRH-averaged bulk modulus and G_{VRH} is the VRH-averaged shear modulus.

2. All entries with missing, null, or invalid Young's modulus values were removed to ensure the completeness of the target variable required for supervised regression.

3. Duplicate materials, identified by repeated `material_id` values, were removed. Only the first occurrence of each unique material was retained to prevent duplication biases.

4. Outliers in Young's modulus were filtered using a percentile-based approach. Specifically, materials with Young's modulus values below the 1st percentile or above the 99th percentile were discarded, removing physically unrealistic or erroneous data points.

5. The dataset was reduced to relevant fields for machine learning, including material identifiers, chemical formulas, crystal system labels, space group numbers, and elastic moduli (bulk, shear, and Young's modulus). Unnecessary or highly nested JSON fields were excluded.

6. After filtering and column selection, the dataset was reduced from approximately 10,994 initial entries to about 10,774 cleaned and validated materials. This final dataset was saved as `elasticity_cleaned.csv` and used for subsequent feature engineering and model development.

3.4 Final Dataset Summary

The final refined dataset contains 10,774 materials after a 2 percent removal of disqualifying data. The refined dataset has 7 core columns, expanding it to 147 total features after feature engineering. The feature types include compositional features (132 Magpie-based descriptors), engineered composition metrics, and 12 symmetry features using one hot encoding. The Young's modulus distribution varied heavily from 6 to 511 GPa with a moderate right skew, showing a wide variance in training data. The dataset spans all 7 crystal systems but is dominated by cubic and tetragonal structures.

3.5 Feature Engineering

3.5.1 Compositional Features

Compositional features describe the elemental makeup of each material without using structural information, which is core to the design of this experiment. ElementFraction and Magpie statistical features from matminer were used. ElementFraction generates fractional abundance for each elements, resulting in a sparse set of one feature per element. Magpie statistical features compute the mean, max, min, range, mode, and standard deviation for 132 elemental properties (such as the atomic radius, electronegativity, valence electrons, etc). Magpie features generate a large and information-rich feature set which captures periodic trends and aggregates elemental behavior. These features represent how composition influences stiffness, bonding strength, and other factors tied to the elastic modulus.

3.5.2 Symmetry Features

Symmetry features encode crystallographic information that affects elastic behavior. One-hot encoding is used for seven crystal features: cubic, tetragonal, hexagonal, orthorhombic, trigonal, monoclinic, and triclinic. Space group binning is employed into five ranges (1-50, 51-100, 101-150, 151-200, 201-230) and then one-hot encoded. This process reduces the dimensionality versus encoding all 230 space groups on their own. These features help the model learn how symmetry constraints influence stiffness.

3.5.3 Engineered Features

Three simple and interpretable features were menually computed from compositions: mean atomic number, maximum atomic number, and average valence electrons. The mean atomic number is a composition-weighted average of atomic numbers. The maximum atomic number is the highest atomic number present in the formula. Lastly, the average valence electrons is the weighted average of electrons based on common oxidation states. These descriptors summarize high-level chemical trends related to density, bonding, and stiffness.

4 Methods

4.1 Machine Learning Pipeline Overview

The workflow follows a compact, end-to-end pipeline:

1. Elasticity data retrieved from the Materials Project API, restricted to materials with complete elastic tensors.
2. Computation of VRH Young's modulus, removal of missing or invalid entries, deduplication, and percentile-based outlier filtering.
3. Generation of compositional and symmetry descriptors using `matminer`, together with a small set of engineered features.
4. Random Forest and MLP models trained on an 80/20 split, with hyperparameters optimized via randomized search and 5-fold cross-validation.
5. Test-set performance assessed using MAE, RMSE, and R^2, with supplementary diagnostics such as parity plots and residual analysis.

All steps are implemented modularly, with intermediate outputs saved for reproducibility.

4.2 Model Selection Configuration & Rationale

4.2.1 Random Forest Regressor

The Random Forest model includes 400 trees with each one being trained on slightly different data using bootstrapping with random feature subsets. Tuned hyperparameters include number of trees, depth, min samples, and max features. This design is chosen as it works without scaling, handles sparse element features, and is robust to outliers. It also provides feature importance scores for interpretability. Tree based models like Random Forest often outperform linear models and shallow neural networks for composition only descriptors according to Matbench. Additionally, Random Forest performs best in medium-sized datasets like this one. Composition based Magpie features contain many nonlinear relationships that trees naturally capture without feature scaling. Boosting models (such as XBoost and LightGBM) were considered but excluded for simplicity and computational constraints.

4.2.2 Multilayer Perceptron (Neural Network)

The Multilayer Perceptron model is a feedforward neural network with 3 hidden layers. It tests architectures of (256, 128), (128, 128), and (256, 128, 64). This hidden layer size is chosen to balance model capacity with risk of overfitting given the dataset size. ReLU activations are used as it offers stable training behavior and strong performance in regression task. The model is trained with an Adam optimizer for faster convergence on high-dimensional feature spaces compared to SGD. Feature scaling using StandardScaler is performed prior to training, which os required because neural networks are sensitive to feature magnitude. Tuned hyperparameters include hidden layer sizes (2-3), activation, regularization (alpha), and learning rate. Alpha is tuned to avoid overfitting given the large number of input features while the learning rate is tuned as it is critical for stable training (tuning ensures convergence across architectures). The model is trained using mean absolute error (MAE) loss, a popular choice for continuous regression targets. The output layer consists of a single linear neuron appropriate for regression.

This model is chosen as it is good for modeling complex linear relationships in high-dimensional features. MLP is included to tester whether learned representations outperform engineered features, with the caveat that deep learning usually requires more data.

4.3 Data Splitting and Cross-Validation Strategy

Data is split using an 80/20 train/test split using a fixed random state. 5-fold cross validation is used during hyperparameter tuning to avoid overfitting. The cross validation metric is based on minimized MAE. Five folds are chosen for balance of computation and stable validation estimates (commonly used fold number). The test set is untouched during training to avoid biases.

4.4 Training & Hyperparameter Tuning Procedures

RandomizedSearchCV is used to sample from hyperparameter grids because it is faster than a grid search and works better for large spaces. Random search evaluates roughly 20 sampled combinations per model. Cross validation results are saved to a CSV for later analysis. The best models automatically refit on the full training set.

4.5 Implementation Details

All experiments were implemented in Python using `scikit-learn`, `numpy`, `pandas`, and `matminer`. The Materials Project API (`mp-api`) was used for data retrieval, and `pymatgen` handled composition parsing. Reproducibility was ensured by fixing `random_state=42` across data splitting, model initialization, and hyperparameter search. The train/test split and feature matrix were saved to disk so all models were evaluated on identical data.

Random Forest and MLP models were implemented via `RandomForestRegressor` and `MLPRegressor`, with training, tuning, and evaluation performed through `RandomizedSearchCV`. All computations were run in a controlled virtual environment with dependencies managed through a project-specific `requirements.txt`. Intermediate artifacts (cleaned data, features, trained models, and evaluation results) were stored to ensure consistent and repeatable execution of the full pipeline.

5 Model Validation

5.1 Validation Metrics

The mean absolute error (MAE) is used as a primary metric for tuning while the root mean squared error (RMSE) is reported as a complementary metric sensitive to large errors. R^2 is used to measure variance explained by the model and to test the original hypothesis. The final metrics are computed on the held-out test set for an unbiased estimate of the model performance.

5.2 Cross-Validation Results

Five-fold cross-validation was used during hyperparameter tuning for both models. The Random Forest model achieved the strongest validation performance with a mean CV MAE of 25.37 GPa (std. 1.27 GPa). The best hyperparameters selected were 600 trees, max_depth=25, min_samples_split=2, min_samples_leaf=2, and max_features=0.6.

The MLP model reached a higher mean CV MAE of 28.41 GPa (std. 1.10 GPa), with optimal settings including a (256, 128, 64) architecture, ReLU activation, $\alpha = 0.01$, and learning rate of 0.005. Overall, the Random Forest demonstrated better validation performance and was therefore expected to generalize more effectively on the held-out test set.

6 Results

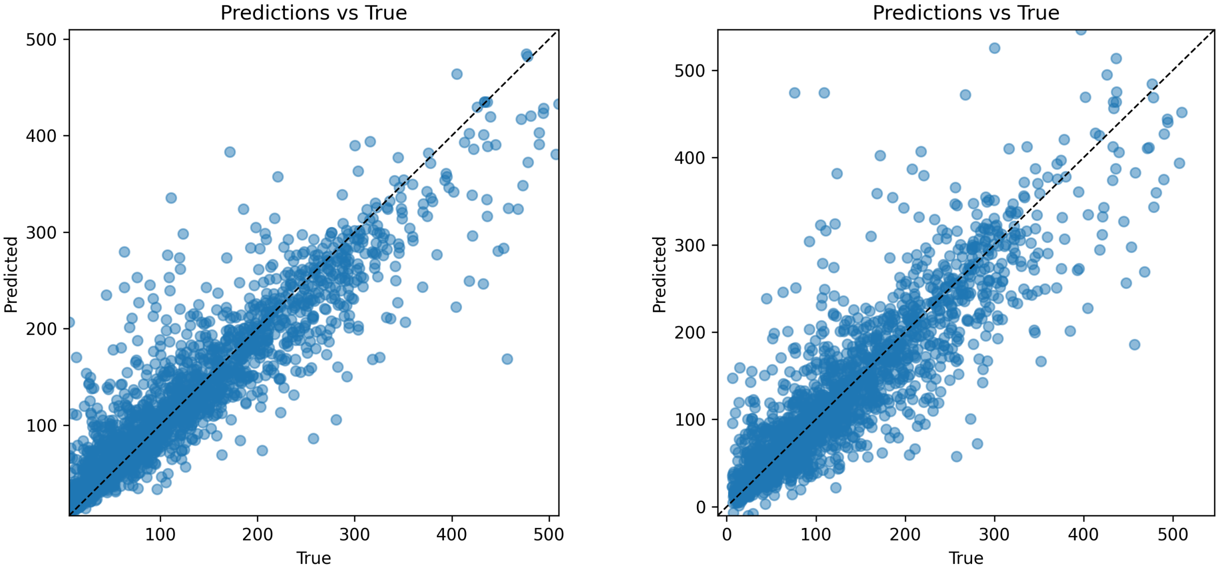
6.1 Test Set Performance & Model Comparison

On the held-out test set, the Random Forest achieved an MAE of 23.80 GPa, RMSE of 38.22 GPa, and $R^2 = 0.836$, outperforming the MLP model. The MLP obtained an MAE of 27.73 GPa, RMSE of 44.63 GPa, and $R^2 = 0.776$. These results indicate that the Random Forest model generalized more effectively and provided the most accurate predictions of Young's modulus. Random Forest consistently outperformed the MLP model on both cross-validation and the test set. Random Forest achieved lower MAE and RMSE with a higher R^2, indicating a stronger overall accuracy and generalization. While Random Forest had a slightly higher variance across CV folds, it maintained better average performance overall. Random Forest also required less preprocessing (due to no scaling needed) and was more stable during training, make it the more robust choice for this dataset.

6.2 Learning Curve Interpretation

Learning curves (see Appendix A) show both models benefit from increased data in validation runs, with Random Forest saturating earlier. MLP exhibits higher variance and slower improvement, consistent with neural network sensitivity to high-dimensional tabular features. Both curves suggest diminishing returns without richer structural descriptors.

6.3 Error Analysis



(a) Random Forest: Predictions vs. True

(b) MLP: Predictions vs. True

Figure 1: Comparison of predicted vs. true Young's modulus for both models.

Random Forest predictions maintained an error magniuted of approximately $\pm(25-30)$ GPa. MLP indicated higher errors ranging from $\pm(28-35)$ GPa. Both models showed underprediction at high modulus values, especially above 300 GPa. To this end, errors tend to increase at modulus extremes ($<$ 50 GPa or $>$ 300 GPa). Random Forest shows stronger bias at the upper range compared to MLP. Residuals for both models are centered at around zero, with Random Forest residuals slightly skewed-right and MLP more symmetric. Both models exhibit heteroscedasticity with larger errors for stiff materials. Given that the $RMSE > MSE$, outliers and long-tail errors appear to be present. Analyzing structure-related error, cubic materials have lower MAE due to high symmetry, while monoclinic and triclinic systems have higher errors due to more complex structure-property relationships. Hexagonal and tetragonal structures performed moderately. For general model error, Random Forest achieved roughly 3 GPa lower MAE and shows a tighter residual spread, while MLP performed similarly in trend but with less overall accuracy. Again, Random Forest is favored here due to stability, lower error, and interpretability. See Appendix B for the full residual distribution plots supporting these observations.

6.4 Feature Importance and Interpretability

Feature importances from the Random Forest model show that predictive power is dominated by compositional descriptors. Table 1 summarizes the contributions of each feature category.

Table 1: Relative importance of feature groups in the Random Forest model.

Feature Group	**Total Importance**	**Notes**
Magpie statistics	∼60–75%	Largest contributor
ElementFraction	∼15–25%	Secondary influence
Engineered features	∼5–10%	Mean atomic number highest
Symmetry features	∼3–8%	Small but non-negligible

Top individual features included Magpie descriptors such as ground-state volume per atom, melting temperature statistics, and unfilled valence counts. These capture atomic packing and bonding trends, explaining their strong influence. Element fractions of common elements (e.g., O, Al, Fe) contributed moderately, while engineered and symmetry features played smaller roles overall. A full ranking of feature importances from the Random Forest model is provided in Appendix C.

7 Discussion

7.1 Interpretation of Findings

The Random Forest model had consistently outperformed the MLP model, indicating that compositional descriptors and statistical features are better captured by tree-based models than by shallow neural networks. This finding likely happens due to the size of the data and network; MLP tends to perform better with extremely large datasets. The test MAE values (24-28 GPa) and error trends suggest that composition-only models can capture broad stiffness trends but cannot fully reproduce elastic moduli without structural information. Underprediction at high modulus values indicates that extreme stiffness requires structural and bonding detail that compositional features cannot encode on their own. Symmetry features had contributed only marginally, while Magpie descriptors dominate model decisions, confirming that aggregated elemental properties strongly correlate with elastic behavior. Coarse symmetry labels alone seem to proviude limited structural insight compared to full space-group or atomic-coordinate representations.

Performance differences across crystal systems imply that high symmetry structures are easier to model (as likely expected) compared to low symmetry models. Learning curves had shown steady decreasing validation error, indicating that additional data continues to improve model performance (again, as expected), although a performance plateau suggests an intrinsic limit of composition-based predictions.

The proposed hypothesis is therefore supported for the Random Forest model ($R^2 = 0.836$) but narrowly missed for the MLP model ($R^2 = 0.776$) based on the target R^2 value of 0.8. Overall, the results support a common trend in materials machine learning, which is that tree based models with engineered descriptors provide strong baselines, but aren't without inherent limitations in their ability to achieve significantly lower errors without using structure-aware models.

7.2 Potential Sources of Error

Several sources of error may have negatively impacted the performance of either model. Given that only composition and symmetry features were used, full atomic structural information was missing, limiting the scope of the data. The elastic modulus also depends heavily on bonding geometry and coordination, which the descriptors cannot fully capture. Additionally, DFT calculated modulus values in the Materials Project may contain noise or inconsistencies.

Certain design choices, such as the space-group bins and engineered features, may have resulted in the loss of finer structural detail. Random Forest averaging can also smoothen out extreme predictions, while MLP performance is affected by limited data and sensitivity to feature scaling.

7.3 Improving Feature Representations

Future experiments can build upon this one in several ways. They may elect to incorporate full structural information, such as atomic coordinates or local bonding environments, rather than relying only on composition. Symmetry one-hot encoding can also be replaced with richer descriptors. Graph based representations can be used to learn structure property relationships. Information loss can be reduced by including higher order statistics or domain specific property sets. Lastly, additional machine learning models should be tested to explore their efficacy in comparison to the ones here.

8 Conclusion

The experiment revealed that a Random Forest model using composition and symmetry-based features is capable of producing well-correlated elastic modulus predictions ($R^2 = 0.836$). A similarly tested MLP model showed similar overall performance ($R^2 = 0.776$) but with consistently lower prediction accuracy than the Random Forest model. Given its high interpretability and relatively simple architecture, Random Forest models are best suited for the purpose of this experiment. For scenarios involving significantly larger datasets, MLP models should still be considered as their performance is heavily dependent on the size of the input dataset (more sensitive than Random Forest). These results indicate that machine learning techniques are capable of predicting highly correlated elastic modulus estimations for materials using accessible compositional data. However, despite this predictive strength, the average magnitude of error ($\pm 25 - 35 GPa$) makes these models unreliable for situations where precise predictions are required. As materials datasets continue to grow, it can be expected that the prediction accuracy of machine learning models in this context will improve substantially, acting as a promising tool in the realm of material science.

References

[1] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, G. Kirklin, A. Tiwari, J. Redfern, and K. A. Persson, "The Materials Project: A materials genome approach to accelerating materials innovation," *APL Materials*, vol. 1, no. 1, p. 011002, 2013.

[2] L. Ward, A. Dunn, A. Foley, K. Dunn, A. Choudhary, and C. Wolverton, "Matminer: An open source toolkit for materials data mining," *Computational Materials Science*, vol. 152, pp. 60–69, 2018.

[3] L. Ward, K. Dunn, A. Foley, A. Choudhary, and C. Wolverton, "A general-purpose machine learning framework for predicting properties of inorganic materials," *npj Computational Materials*, vol. 2, p. 16028, 2016.

[4] M. De Jong, W. Chen, R. Notestine, K. Persson, K. A. Ceder, M. Gamst, A. Tacoma, and M. K. Asta, "Charting the complete elastic properties of inorganic crystalline compounds," *Scientific Data*, vol. 2, p. 150009, 2015.

[5] A. Dunn, Q. Wang, A. Ganose, D. D. Chen, and A. Jain, "Benchmarking materials property prediction methods: Matbench," *npj Computational Materials*, vol. 6, p. 138, 2020.

[6] F. A. Faber, A. Lindmaa, O. Aquilante, and R. L. J. Malmqvist, "Crystal graph convolutional neural networks for materials property prediction," *International Journal of Quantum Chemistry*, 2017.

[7] C. Chen, W. Ye, Y. Zuo, C. Wang, and K. A. Persson, "Graph networks as a universal machine learning framework for molecules and crystals," *npj Computational Materials*, vol. 5, p. 42, 2019.

A Learning Curve Plots

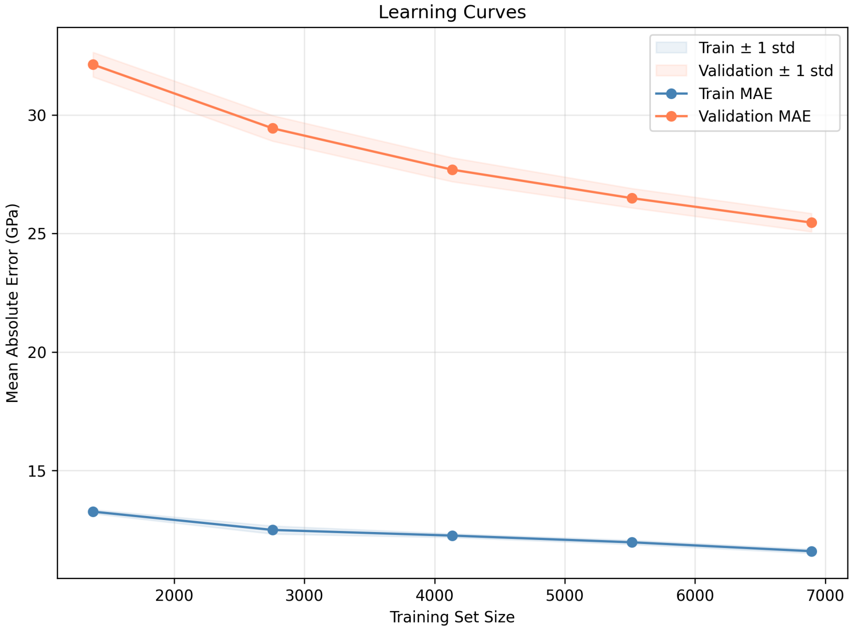


Figure 2: Learning curves for the Random Forest model showing training and validation MAE as a function of training set size.

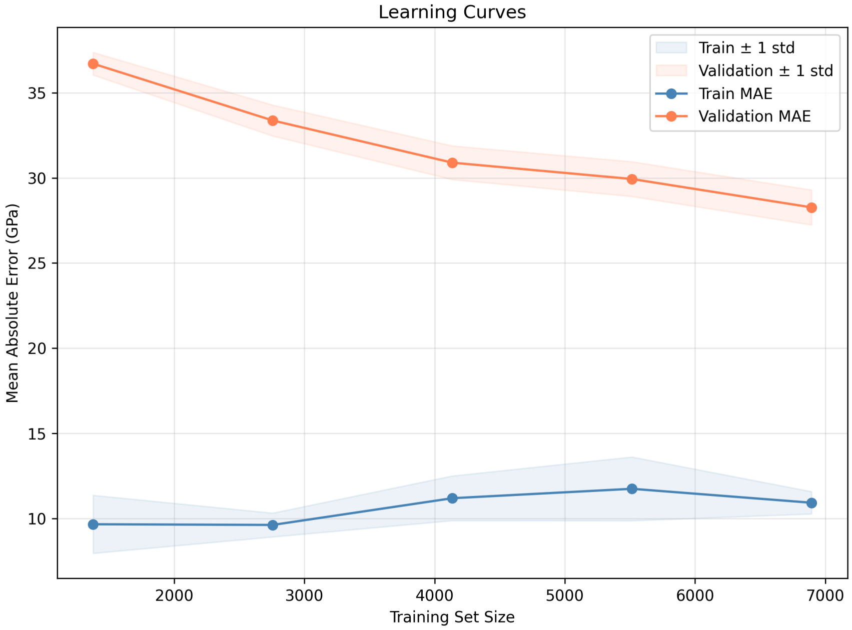


Figure 3: Learning curves for the MLP model showing training and validation MAE as a function of training set size.

B Residual Plots

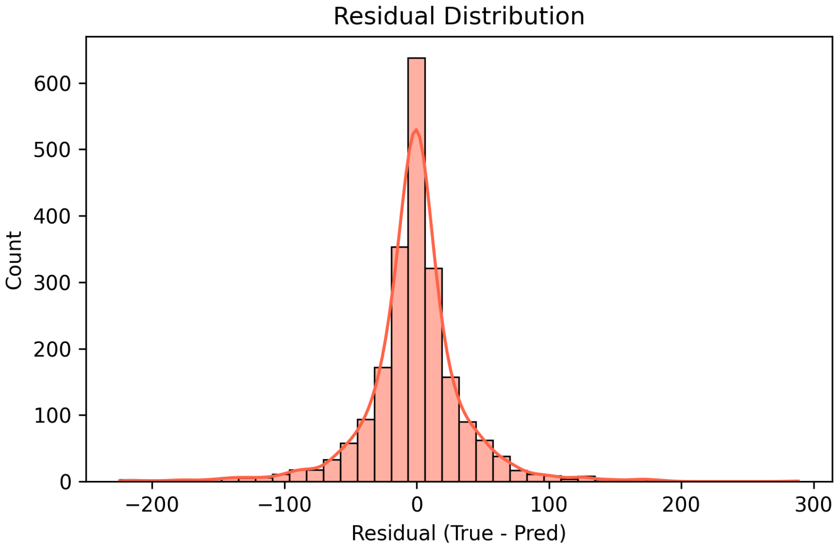


Figure 4: Residual distribution for the Random Forest model

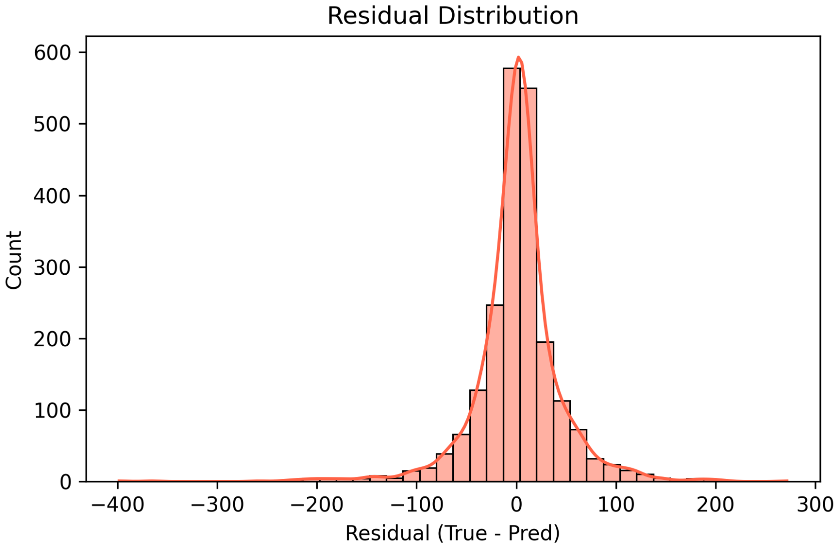


Figure 5: Residual distribution for the MLP model

C Feature Importance Chart

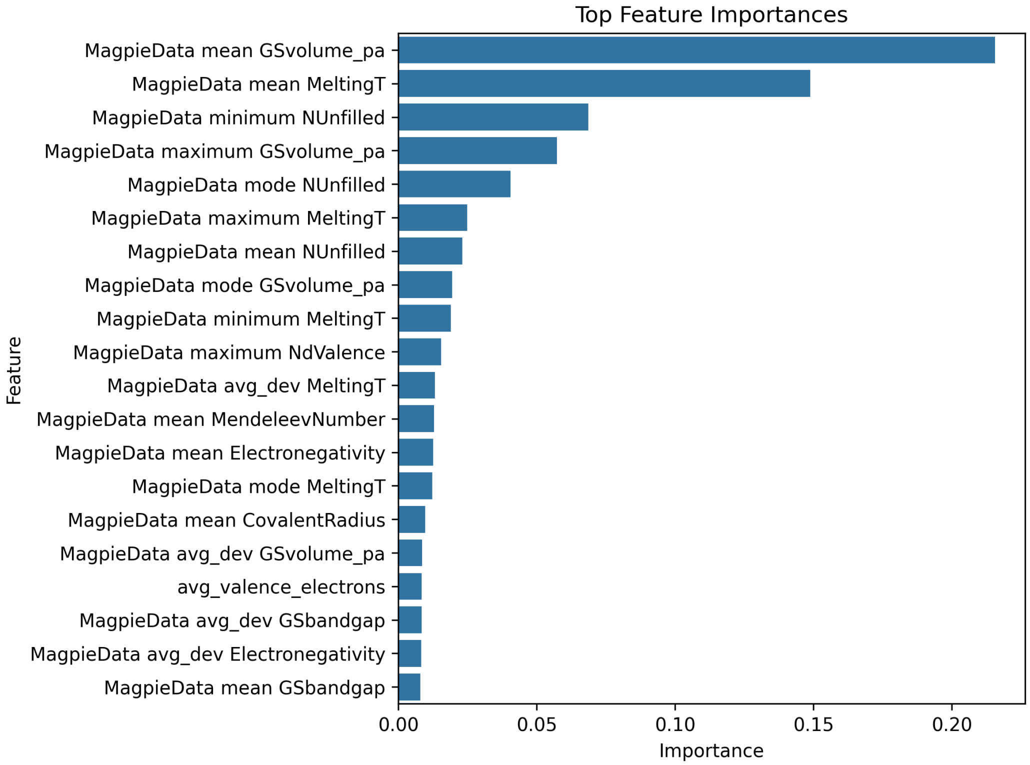


Figure 6: Feature importance rankings from the Random Forest model.

D Code Repository and Implementation

The full source code for data acquisition, preprocessing, feature engineering, model training, and evaluation used in this work is publicly available at the following repository:

`https://github.com/Jaiparmar940/ME490-Final-Project`