PREDICTING THE FEATURES OF INORGANIC MATERIALS

COLBY WIRTH, BENJAMIN FRANKLIN, LUCAS MATHESON

ABSTRACT. Launched in 2013 as part of the Materials Genome Initiative, the Materials Project is an open-source collaboration in which computer and materials scientists aimed at making advances in materials science. Traditional methods such as measuring properties in laboratories and Density Functional Theory are computationally intensive. The Materials Project seeks to expedite research by using machine learning (ML) and artificial intelligence (AI), in order to circumvent these traditional means. The paper focuses on predicting the properties of inorganic compounds using ML and AI models. We pursued two primary objectives: (1) training models on experimental datasets—containing non-theoretical, lab-measured data—to predict specific features of elements, and (2) evaluating model performance against theoretical datasets derived from computational estimations (e.g., DFT or ML-generated data). Despite rigorous data cleaning and tuning hyper-parameters, our results show that achieving high accuracy across all attributes is not always guaranteed.

Keywords: Machine Learning, Artificial Intelligence, Inorganic Chemistry, Materials Project

1. INTRODUCTION

Since the 1970s, Density Functional Theory has been a widely used technique for estimating the properties of chemical compounds. This method is based on the Universal Functional, a computational approach used to simulate a molecule's quantum mechanical properties. However, the Universal Functional is limited by its computational complexity, as calculating exact results for complex systems can be computationally intensive. In order for DFT to produce exact or highly accurate results, it is believed that the computational complexity of such calculations would require quantum computation. These problems are called the Quantum Merlin Arthur (QMA) class of problems. These QMA problems are analogous to the NP class of problems in classical computing, which are defined to involve non-deterministic computation for verifying solutions. https://www.nature.com/articles/nphys1370.pdf

Machine Learning (ML) offers an alternative approach to DFT. ML models provide predictions in relatively low-intensity computation. In this project, our goal was to obtain predictions with low degrees of variance and high degrees of accuracy through the implementation various ML and AI models with Python Sci-Kit Learn libraries. At the beginning of the project, the level of achievable variance and accuracy scores was not accurate. Through rigorous data cleaning and data aggregation, we found that in some cases we could achieve R^2 scores as high as 0.991. This figure is apparently arbitrary, as with predicting other features, our the models were less performative with scores R^2 scores as low as 0.3xxx. In the

2000 *Mathematics Subject Classification.* 68T01.

following sections, we outline our pipeline process, the data, and key insights that can be drawn from the data.

2. Dataset Overview and Preprocessing Pipeline

From the official website *https://next-gen.materialsproject.org/*.org, the dataset can be accessed via a public API. At the time of the writing of this paper, the dataset contains 155,361 compounds with 70 features represented as a table with tuples and columns respectively. From the 70 features, 12 contained either duplicated data from another column, or data related to the status of the database, therefore they could be removed from the set. From the columns, there included categorical data that could converted to numerical data with one-hot encoding functions. Al through the random forest models can handle categorical data, the binary data created from one-hot is much more computationally efficient as it each data point is represented as a bit, rather than a String or other data type. This was done for nested data as well, as we could one hot encode data from regarding the crystal structure from the nested dictionaries.

After converting the categorical data to numeric, there were four attributes that contained null values: two were binary attributes and each contained 8 null values. These were filled with the value that was more prevalent in the dataset. The other two attributes eached contained 64254 missing data points. These were to be left as null in order to not artificially influence the model.

To finalize the data pre-processing, the data was split into two subsets: synthesized and hypothetical datasets. The hypothetical molecules contained mostly synthetic data, as the molecules had not been synthesized in a lab, and tested. In order to stray the farthest from the real results, the model was to only be trained on the synthesized molecules dataset. We would later compare the performance of our model against the synethic molecules dataset.

3. Main Content

Feature Selection:

The following features were selected to train predictive models with: 1. Band-Gap 2. X-ray Absorption Spectroscopy (XAS)....
Band-gap is the property associated with the amount of energy required to move an electron from an atom's valence band to its conductive band. It is a continuous value, measured in electron vols (eV).

Model Selection

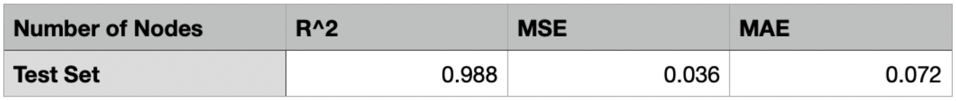
For all target features, it was presumed that its relationship to its predictor features was non-linear. Therefore, a Decision Tree was always trained first. Then a Random Forrest Regressor (RFR) was trained. Generally, the RFR models will produce less bias as they make predictions off of the average of a predetermined number of trees. For attributes that did not achieve acceptable evaluation metrics Deep Neural Networks (DNNs) were after implemented.
For all target features, a Decision Tree first. Then a Random Forrest Regressor (RFR) was trained. Generally, the RFR models will produce less bias as they make

predictions from the average of a predetermined number of Decision Trees. For attributes that did not achieve acceptable evaluation metrics Deep Neural Networks (DNNs) were after implemented. For each target feature, whichever model performed the best against the test set was tested against the Synthetic Dataset.

For all target features, a Decision Tree first. Then a Random Forrest Regressor (RFR) was trained. Generally, the RFR models will produce less bias as they make predictions off of the average of a predetermined number of trees. For attributes that did not achieve acceptable evaluation metrics Deep Neural Networks (DNNs) were after implemented.

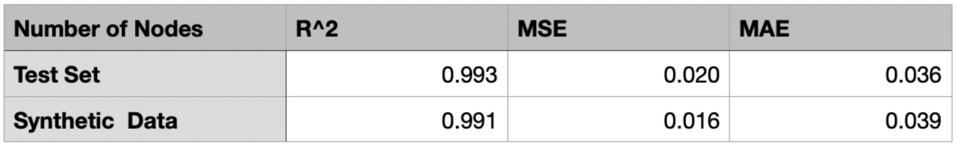
Band Gap

The Band Gap was the first feature to be investigated and predicted. Our standard approach for feature selection was to use the top ten correlated features. The top five positively correlated and top five negatively correlated features were selected. Sci-Kit Learn's standard Recursive Feature Elimination tool could not be used as we had null values in the dataset. Therefore, they were selected by evaluating a correlation matrix with respect to the goal feature.



Number of Nodes	R^2	MSE	MAE
Test Set	0.988	0.036	0.072

Figure 1. Band Gap with Decision Tree



Number of Nodes	R^2	MSE	MAE
Test Set	0.993	0.020	0.036
Synthetic Data	0.991	0.016	0.039

Figure 2. Band Gap with Random Forrest Regressor

When evaluated against the test set, the Decision Tree produced R^2, MSE and MAE scores of $0.988, 0.46,$ and 0.72 respectively. These scores show that the model performs incredibly well. However, we believed that we could improve our results with a Random Forrest as they generally produce a lower bias than the Decision Tree.

These presumptions were validated as the model produced an R^2 score of 0.993 against the test set. Additionally, with this being the highest performing model, it was tested against the Theoretical dataset. With R^2 of 0.991, the model predicted the data 99.8% as effectively as it did against the testing set. This indicates that the model exhibits little bias, and performs exceptionally well overall.

XAS

The performances from models trained on XAS data were very poor in comparison.

To begin with, a Decision Tree was trained, and then a Random Forrest Regressor. The following hyper-parameters were tuned rigorously: 'Max Depth', 'Min Samples Split', 'Min Samples Leaf', 'Max Features', and 'Criterion'. We were able to achieve an R^2 score of 0.298 on the testing set, and an R^2 score on the Theoretical set of -0.66. This indicates that the Tree displayed extreme levels of bias.

Various hyper-parameters were tuned, beginning with

Model Tuning

Results

4. Conclusion

References

Department of Computer Science, University of Southern Maine, Portland, ME
Email address: colby.wirth@maine.edu