NUS | Engineering
National University of Singapore

UNDERGRADUATE RESEARCH OPPORTUNITIES PROGRAMME (UROP)

PROJECT REPORT

Using machine learning to predict bulk modulus

for Heusler Alloys

Liang Rongjian

Project Supervisor: Shen Lei

Department of Mechanical Engineering, Faculty of Engineering

National University of Singapore

Semester 1 / Academic Year 2020-2021

Abstract:

With the progress of machine learning and the development of computing hardware, recent years have seen an increasing trend of utilizing various machine models in material science. These models including random forest and neural networks are helping materials scientists in screening stable materials from tens of thousands of candidates and predicting properties for new substances. In this study, several machine learning models are trained to predict the bulk modulus of Heusler alloys and have achieved some success on the current data available. These models, while focusing on a small subset of special materials, are hoped to shed new light on the possible issues and inspire innovative methods in training more sophisticated models to predict mechanical properties of a wider range of materials, so that they could be used in pro-active screening for potential candidates with expected mechanical traits.

Preface and acknowledgments

This project is initiated by Prof. Shen Lei who has kindly offered many guidance and advice on how to approach the project aims and provided access to many key resources along with the development progress.

All original data and packages used in this project shall be credited to the Material Project database, the matminer, and pymatgen development teams as well as Scikit-Learn, NumPy, and Pandas who have provided many useful tools for this field of research. Without them, everything would have to be implemented from scratch.

Table of Contents

1. Introduction

This project is to utilize machine learning models to predict the mechanical properties of Heusler alloys. Similar to most machine learning projects, it is divided into several key steps including data retrieval, data cleaning, feature engineering, model selection, tunning of the model to improve performance, and finally the evaluation of the models.

As the performance of the models is of central concern, the most frequently asked questions throughout the project are, “how well does the model perform on unseen datasets” and “what could be done the improve the models performance”. Depending on the complexity of fitting models used and the training samples available, the answers to these questions would be very different from time to time.

Since the concepts and implementation of those machine learning models have been discussed and implemented by many others, the very idea this project would be shifted to the idea of underfitting and overfitting which will be used constantly in tuning the models.

2. Design of the machine learning project

2.1 Data Retrieval

The structural information of Heusler alloys is acquired from the Material Project database by inputting the formula from the Heusler magnetic datasets currently available in the matminer datasets. There are 1153 entries in the Heusler magnetic dataset, but not all of them have mechanical properties such as elasticity, Poisson's ratio, and crystal structure data documented in the Material Project database. These data will be foundation on which the models will be trained and built on.

2.2 Data cleaning and the feature engineering

With all the retrieved entries stored in a pandas array format, all the rows containing null or empty values are dropped. This leaves only 306 rows with full structure and elasticity data on which the model could be trained.

First the material strings are converted to composition columns containing elements in the alloys. Then using the composition data element properties could be derived by the matminer conversion packages. The oxidation composition is also converted from the composition data. Finally, density features are also added to the datasets by using the structure of the materials. After the feature engineering the dataset is far more complex than original and might pose a potential threat of over-fitting since the sample size is relatively small.

In this project, the bulk modulus namely the K_VRH is used as target output. The other mechanical properties that are highly relevant including the Poisson's ratio, G_VRH, and anisotropy elasticity as well as other non-numerical columns are dropped from the training data.

2.3 Model selection and evaluation strategies

Considering the complexity of the dataset, a rather simple and unsophisticated model could provide a relatively good benchmark for the performance evaluation of more complex models in future steps. Therefore, a ridge regression model with regularization parameter is used as primitive fitting method.

More sophisticated models, namely Random Forest regressor and Gradient Boosting Regression Trees are then used. A common problem of training models on small dataset is the tendency of getting relatively high variance and low bias. Both models are ensemble methods in order to trade smaller bias for lower variance and achieve a good balance between these two.

Preprocessing of training data is done based on the nature of each model. For linear model like Ridge regression, the data is centralized using the standard scaler. As of the more sophisticated two, unsupervised dimensionality reduction techniques are used to control the dimension of feature space and curb over-fitting. Hyperparameter tunning via exhaustive grid-search is used to determine the optimal parameters used in these methods.

Models are all assessed by ten-fold cross-validation to evaluate their variance and bias. In order to see if the model is over-fitting or under-fitting, learning curve is used where the r squared score is plotted against the number of training samples. During the plotting process, a given number of training data is further split into the training data and validation data to perform a ten-fold cross-validation. The mean scores are then represented in the plot as solid curves while the standard deviation of the score for each fold are then used to plot the shaded area around the line, indicating the variance of score across all the fold. This will help paint a better picture about the performance of the model.

The r squared score indicates the strength of linear relationship between variables, in this case, the predicted bulk modulus and the true values in the dataset.

Finally, the true values are plotted against the predicted values to help give an intuitive picture of the dispersion about those value as well as spotting some outliers.

ww

3. Results evaluation and discussion

3.1 Ridge regression model performance and analysis

For linear model, the inputting data is all standardized and centralized around origin to improve the model's performance. The only hyperparameter to tune would be the regularization strength.

After the hyperparameter tuning, the best model is then fitted using all the training data and passed to the ten-fold cross-validation whose evaluation metrics are r squared score and root-mean-squared-error.

The scores for Ridge regression with the alpha value set to 10 is as followed:

```
Cross-validation results:
Folds: 10, mean R2: 0.750
Folds: 10, mean RMSE: 18.870
```

Figure 1:ridge regression cross-validation results

The simple model performs decently from the cross-validation scores. And the learning curve is as followed:

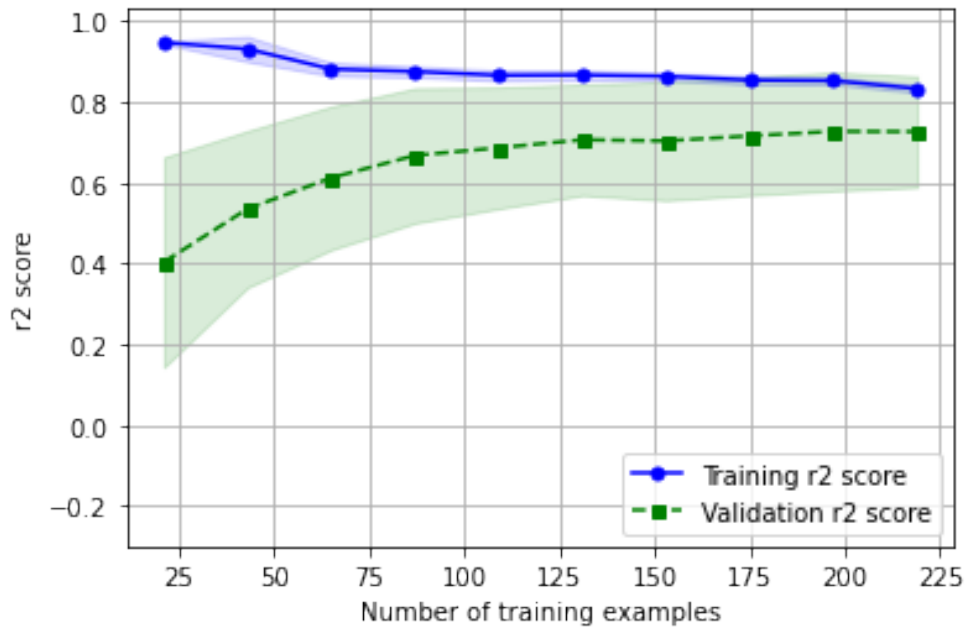

Figure 2: Learning curve for ridge regression

The curve is plotted on the training data, which is 90% of the dataset. The shaded area beside each line indicated the possible range of the value. Since the r squared score is

a scaled mean-squared-error in nature, this plot also shows the relative prediction accuracy of the model.

As training samples pour in, the difference between training and validating accuracy is shrinking, and both values are closing up to around 0.8 as all samples have been used. The rather small difference between training and validation accuracy (around 0.10 at the end of the training) indicate that the model is not over-fitting too much.

However, this is by no means showing that the ridge regressor is one of the best as the training score decreases with larger sample size. And one may observe the constant width of shaded area beside the validation line, implying that the model produces a rather large range of variance on unseen data and unable to reduce it with increasing number of training samples. This is limited by the potency of the simple linear ridge regression model.

Finally, below is the scatter plot showing the dispersion of the predicted values against the true values. Ideally all the data points shall cluster on the dashed line where predicted values for each point are equal to the true values.

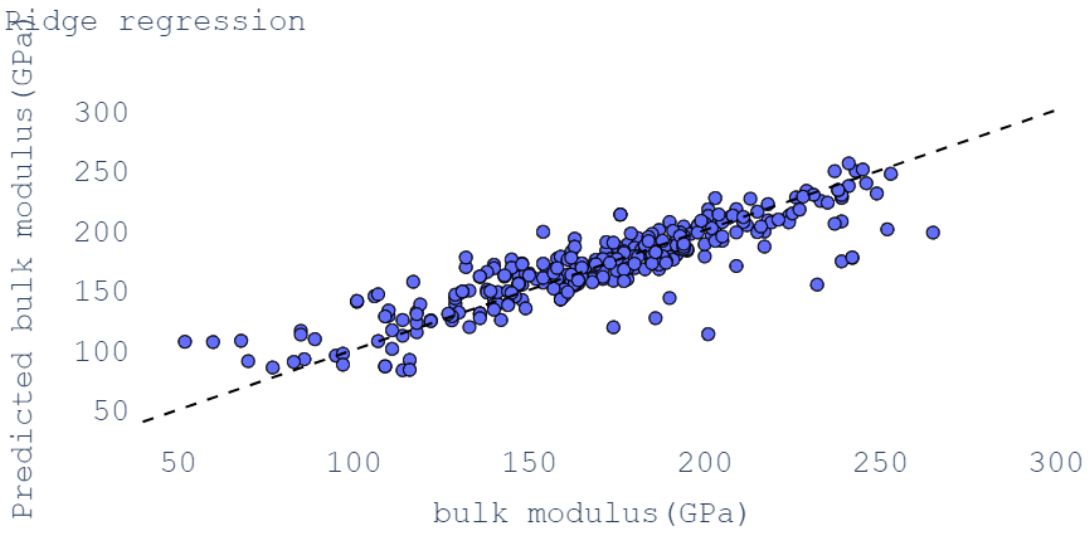

Figure 3: Predicted values plotted against true values

3.2 Random Forest regressor model performance and analysis

Random Forest Regressor is a powerful model which does not assume linear relationship between the features and the target, but it also suffers from the tendency of over-fitting. A Principal-Components-Analysis (PCA) is used to first reduce the dimensionality of the feature space, namely the number of features, before fitting the Random Forest regressor.

These two processes make up a pipeline whose parameters including output components number from PCA, number of trees in the Random Forest and the maximum depth of the trees are tuned in the grid-search.

The grid-search yields following parameter with scoring method set as r squared score, and its performance in cross-validation is also shown as below:

```
{'pca__n_components': 20, 'randomforestregressor__max_depth': 20, 'randomforestregressor__n_estimators': 200}
```

```
Cross-validation results:
Folds: 10, mean R2: 0.644
Folds: 10, mean RMSE: 22.652
```

At first glance, the Random Forest seems to perform poorly compared to the previous one, but the learning curve below will reveal more details.

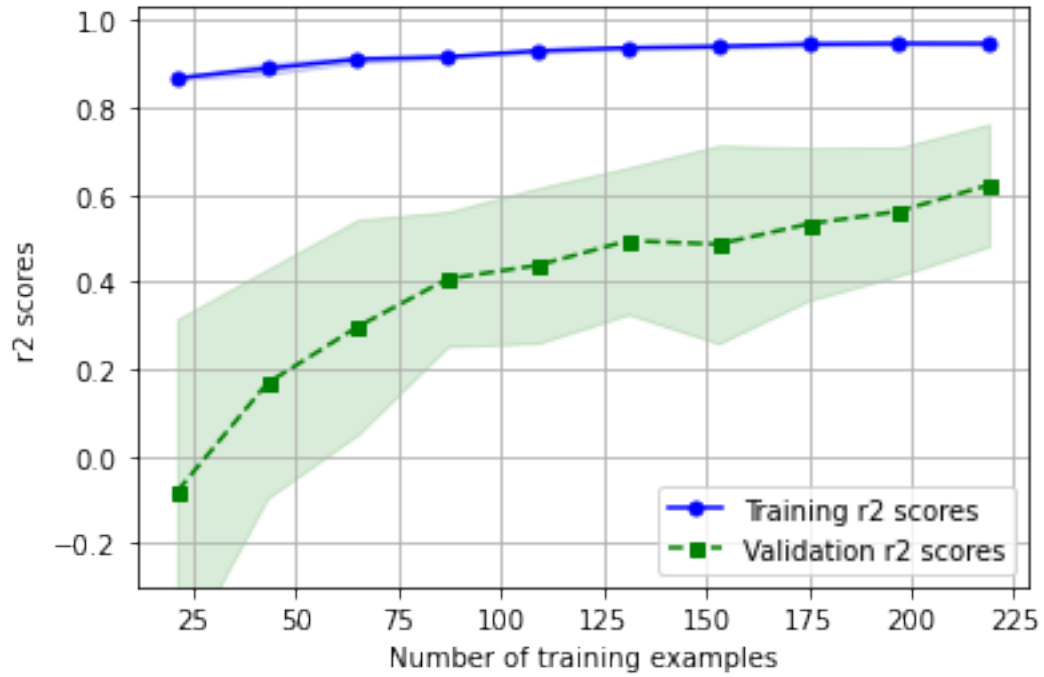

Figure 4: Learning curves of Random Forest Regressor using the best parameters from grid-search

The model achieves near perfect fitting for the training set from the very beginning but fails miserably at the validation set, indicating that the model over-fits the data given small sample sizes. With more data available for training, the performance on the validation sets climbs steadily with slightly shrinking variance as the width of green shaded area deceases, while the scores for training set remains around 0.9 and even falls slightly at the end of the curve.

It seems that this small dataset is against the Random Forest which will perform much better given more data, as seen from the trends in the learning curve. The convergence of training and validating accuracy has not been achieved within the currently available dataset, which is very different from the ridge regression whose converges early but also reaches its possibly performance ceiling. Therefore, while this model is outperformed by the ridge regression, it is still a strong candidate with great potential.

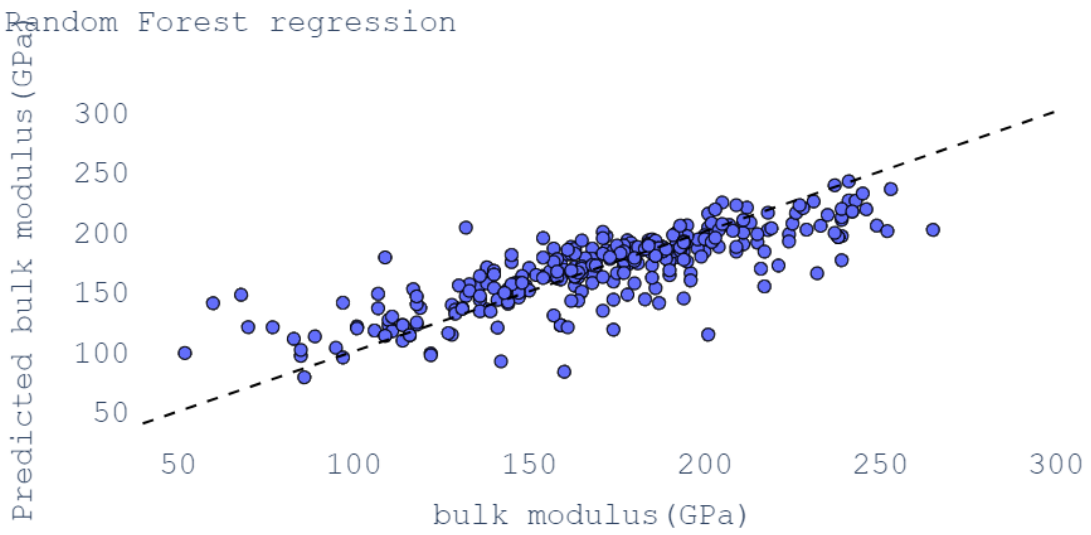

Figure 5: Scatter plot of predicted value against true value for Random Forest

Finally, above is the scatter plot of predicted value against true values. The clustering of the data points is not as dense and close as the one of ridge regression, which is consistent with the scores from cross-validation.

3.3 Gradient Boosted Regression Trees model and evaluation

Gradient Boosted Regression Trees (GBRT) is a powerful ensemble model that improve performance by sequentially adding predictor to an ensemble, each correcting its predecessor. This will help tightening up the clustering as seen in the Random Forest more quickly in this small dataset and possibly make more accurate prediction. It also has a unique hyperparameter called subsample that enable trading a higher bias for a lower variance.

The same dimensionality reduction technique is used, and the cross-validation yields the following result with the parameter set as below:

```
{'gradientboostingregressor__learning_rate': 0.1, 'gradientboostingregressor__n_estimators': 150, 'gradientboostingregressor__subsample': 0.6, 'pca__n_components': 40}
```

```
Cross-validation results:
Folds: 10, mean R2: 0.747
Folds: 10, mean RMSE: 18.748
```

The learning curve is also plotted and shown as below:

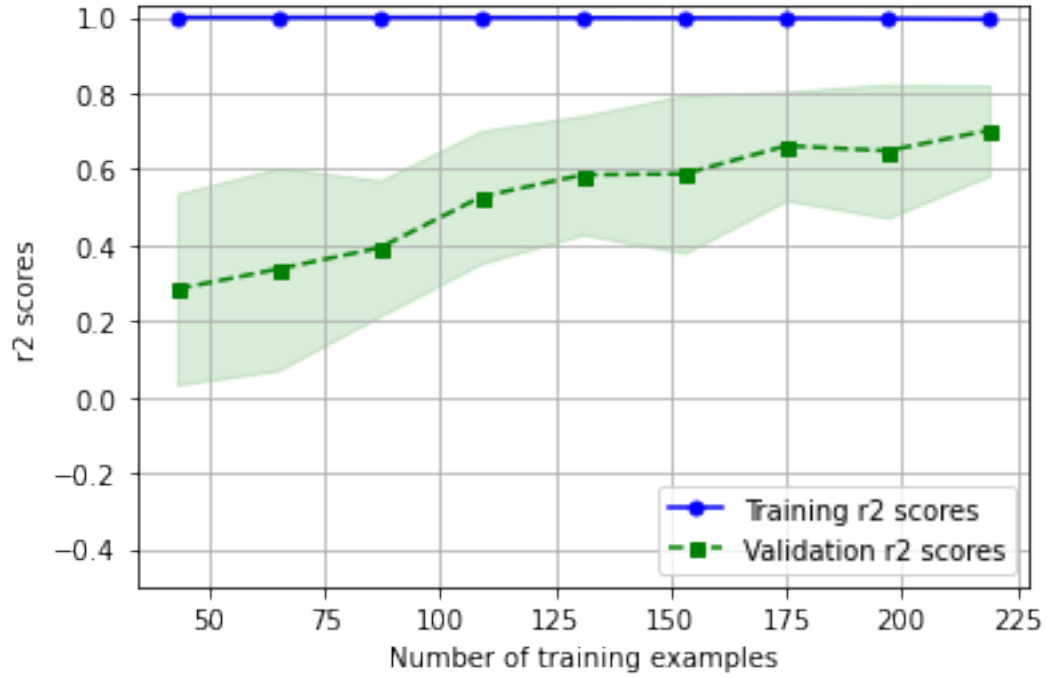

Figure 6: Learning curve for GBRT

One could see that the training accuracy remains perfect across the samples and the validation accuracy climb steadily with a much smaller range of variance compared to the Random Forest regressor.

Finally, below is the scatter plot with predicted values against the true values:

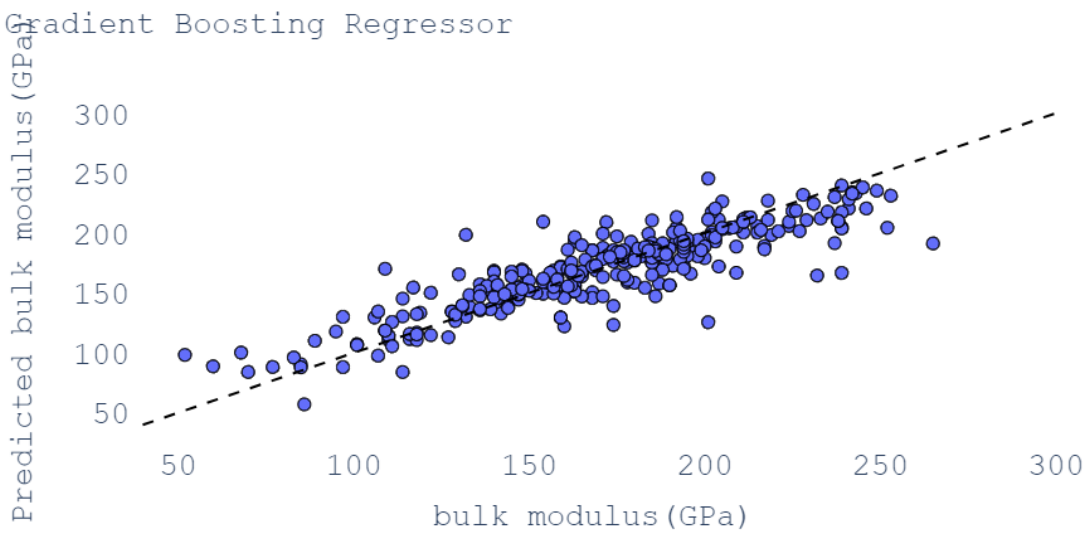

Figure 7: scatter plot of predicted values against true values for GBRT

One could see that despite the “trailing tail” at the end of the upward-going dashed line, the clustering of the data points is rather tight and close around the line, and the overall envelope of points is smaller compared to the one of random forest. Therefore, GBRT seems to be a better candidate than the random forest regressor.

4. Conclusion

There are several limitations about this project including relatively insufficient dataset which hinders the further training of those more potent models and may present a picture biased against those sophisticated machine learning algorithms as they tend to over-fit the data and are penalized in cross-validation.

Although some data preprocessing techniques are used, they may not be perfectly suited for this scenario as they are all unsupervised methods. As for the hyperparameter tunning, due to the limited computation power and time, the searching space is first narrowed down by hand and tuned using constant spacing of hyperparameter or tunning one while setting others as constant, which may not yield exactly the global optimum.

However, despite these limitations there are still some conclusions that could be drawn on the performance of these models, thanks to the evaluations made by plotting learning curves and scatter plots.

In a small dataset, a simple model who has a less tendency to over-fit may outperform those sophisticated ones when evaluated by cross-validation, and learning curve is an important and essential tool in picking up this bias. This is more prominent in this project as there are far more features than samples in the cross-validation, making those potent models to fit perfectly to the training data while fail at the validation data.

Although the linear model takes a lead in cross-validation as compared to the other two, its performance on training and validation data quickly converges and approaches the performance ceiling of the model, while the validation accuracy of the other two are still climbing steadily towards a much better training accuracy. Therefore, linear model is a good starting point for a machine learning project, but it is never a destination. One may say that the ceiling and scope of the linear model is set as it pre-assumes a linear relationship between features and target, which in many cases such relationship might be non-linear. Those model who make less assumption about assumption about the dataset will be more potent and see a much wider scope of application, but also harder to tune since it requires more data to overcome the over-fitting at the beginning.

As for the predictor for bulk modulus of Heusler alloys, this project shows that both random forest and gradient boosted regressor trees are strong candidate but they would need more data to fully exercise their potential.