Superconductor Critical Temperature Analysis

Braden Anderson

September 8, 2022

I Introduction

Superconductors are materials that, when cooled below some material specific critical temperature T_c, have zero electrical resistance. To begin understanding the value in identifying such materials, consider the equations below which govern the relationships between current (I), voltage (V), resistance (**R**) and power (P) in a direct current (DC) circuit:

$$(1)\ V = IR, \qquad (2)\ P = IV, \qquad (3)\ P = I^2R$$

Equation (1) is ohms law and states that the voltage drop across a component (or a length of conductor) is the product of the conductor's resistance and the current flowing through it.

Equation (2) states that the instantaneous power supplied or absorbed by an electrical component is the product of the voltage across the component and the current flowing through it. A positive sign indicates power being absorbed by the element, and a negative sign indicates power being supplied by the element.

By replacing the voltage in (2) with the definition of voltage provided by (1), we arrive at equation (3) which states the power dissipated by a resistive component is the product of the component's resistance and the square of the current flowing through it. Since both resistance and current squared are always non-negative quantities, the result of (3) will also be non-negative which indicates power dissipation (absorbed). Considering these results for the special case of a superconductor provides intuition regarding the value in identifying materials with these properties. Specifically, for the case of DC traveling through a component with zero resistance (i.e., a superconductor), equation (3) implies that zero power loss would occur.

Despite numerous high value theoretical use cases for superconductors, such as the loss-less power transmission example described above, very few have been realized due to challenges associated with identifying materials which make implementation feasible.

One line of research aimed at overcoming these challenges involves the use of machine learning to understand what features of the material are most predictive of a superconductor's critical temperature. This approach was used in [1], where the authors constructed a dataset of 21,263 superconductors with known critical temperatures. For each superconductor, a set of 8 fundamental material properties (e.g., atomic mass, first ionization energy) were put through a feature extraction process which produced a total of 81 attributes for modeling the superconductors critical temperature.

In this paper, we leverage the dataset produced in [1] to construct linear models that predict the critical temperature of a superconductor. Our goal is to replicate and extend the findings in [1] and gain a better understanding of the relationship between the chemical properties of a superconductor and its critical temperature.

II Methods

Data Preprocessing

The data used in this analysis was also the subject of prior research described in [1], which included a significant amount of work to preprocess and prepare the data for modeling. This prior work ensured the data was free of missing and duplicate observations, which we were able to quickly verify. The only additional preprocessing steps performed were:

1. Merged the train.csv and unique_m.csv data sources into a single pandas data frame.
2. Removed the nine columns which contained all zeros. Each of these nine columns ('He', 'Ne', 'Ar' 'Kr', 'Xe', 'Pm', 'Po' 'At', 'Rn') corresponds to a single chemical element and indicates how much of that element is present in the superconductor. For example, the superconductor with chemical formula:

$$Ba_{0.2}La_{1.8}Cu_1O_4$$

Contains the values 0.2, 1.8, 1 and 4 in the "Ba", "La", "Cu" and "O" columns respectively, and zeros in all other chemical element columns. The presence of all zeros in these nine columns indicates that no superconductors in the dataset contain any of the nine corresponding elements, which is not information that would be useful to an inferential or predictive model.

Exploratory Data Analysis

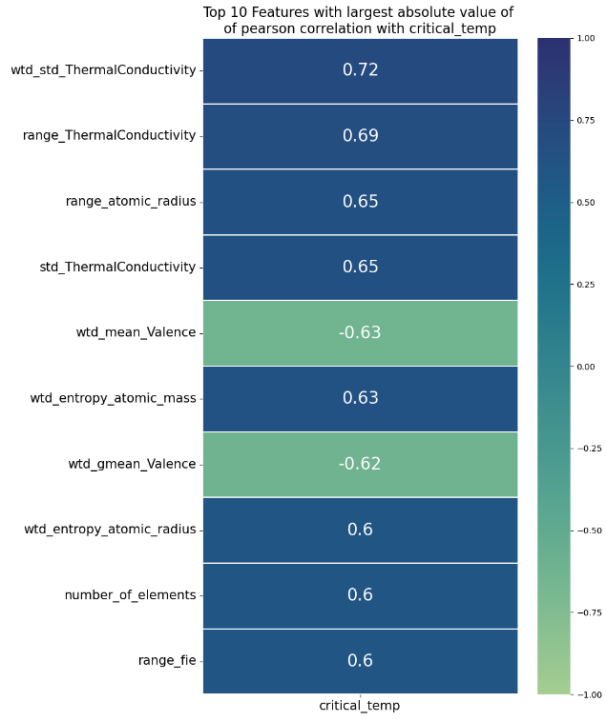

Figure 1: Pearson Correlations

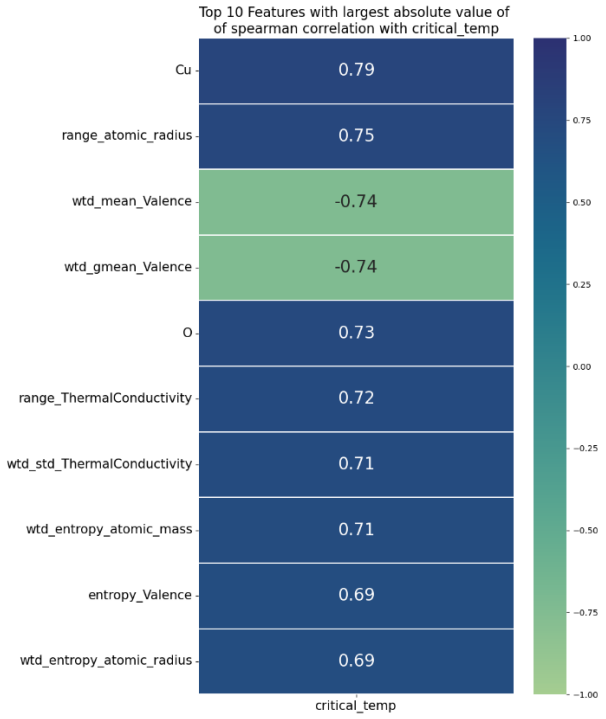

Figure 2: Spearman Correlation

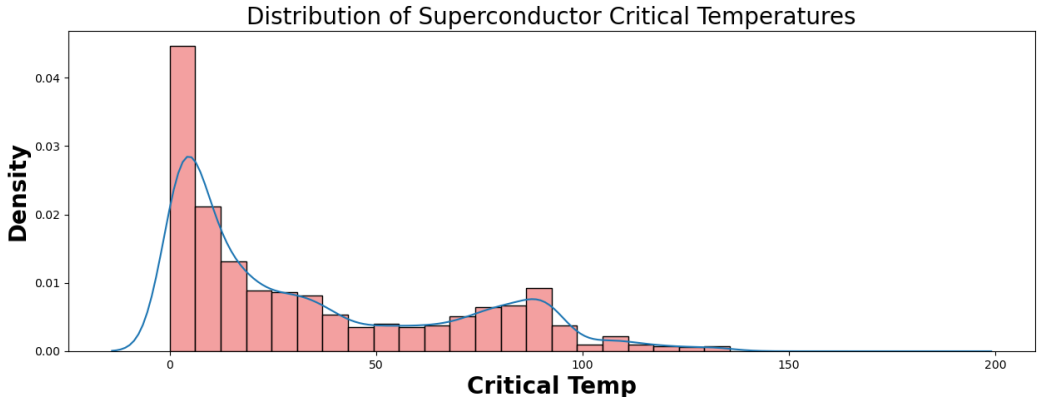

Figure 3: Distribution of Superconductor Critical Temperatures

We explored the relationships between the chemical properties of the superconductor and its associated critical temperature through the use of correlation statistics. The 10 features that have the strongest correlation (largest absolute value) with critical temperature according to the Pearson and Spearman correlation measures are shown in **Figure 1** and **Figure 2** respectively. Since the Spearman correlation coefficient is a non-parametric measure based on ranks, it has the ability to detect non-linear associations where as the Pearson measure will only identify linear relationships. Note that in **Figure 2** the feature with the largest Spearman correlation, **Cu**, is not listed in the top 10 of the **Figure 1** Pearson table. This observation becomes relevant in the model evaluation section where we show that after all of the strictly linear models failed to utilize **Cu**, it becomes the most important feature in a model with non-linear terms.

In **Figure 3** we show the distribution of our target variable, critical temperature, and note that it has a significant right skew. On the right-hand side of Figure 3 we see a large area of white space and an exaggerated tail on the blue density curve, both of which are being caused by the material H_2S_1 which has the highest critical temperature in the dataset at 185. Considering the second largest critical temperature is 143, and the 95^{th} percentile of all critical temperatures is 94, the material H_2S_1 is a significant outlier. While overall model performance would be improved by excluding H_2S_1, there is no justification for its removal other than its inconveniently large critical temperature and therefore the decision was made to include it in the models.

Before discussing preliminary models in the next section, we believe it is worth explicitly stating that despite the use of linear regression models throughout this work a decision was made to not perform any formal analysis of the linear regression model assumptions. The work that follows will not make any claims of statistical significance, or attempt to quantify the uncertainty of either model predictions or parameter estimates. In other words, the analysis performed here will not utilize the statistical inference techniques that the formal assumption verification step is designed to protect.

Establishing a Baseline

Prior to any work aimed at optimizing model performance, we established a baseline by performing 5-fold cross-validation on a selection of four models using default hyperparameter values (defaults per scikit-learn). Unless explicitly stated otherwise, all models generated throughout this analysis were fit on preprocessed features which were scaled to zero mean and unit variance.

The baseline models included an unregularized linear regression, LASSO (L1 regularized) linear regression, Ridge (L2 regularized) linear regression, and ElasticNet (L1 & L2 regularized) linear regression. The average 5-fold cross validation performance of each model is provided below:

	Train RMSE Shuffled, random seed = 42)	Validation RMSE (Shuffled, random seed = 42)
Null Model (Predicts mean)	34.2535	--
Linear Regression	16.57132	22.6167
Lasso ($\alpha = 1$)	18.9137	21.0814
Ridge ($\alpha = 1$)	16.5765	22.4481
Elastic Net ($\alpha = 1, l1\ ratio = 0.5$)	19.3867	21.3114

Table 1: Baseline Model Performance

Linear Regression Baseline:

The lower training root mean squared error (RMSE) (16.57) compared to validation RMSE (22.61) provides evidence that the unregularized linear regression model is overfitting to the training data (i.e., suffering from high variance). This result is not surprising given that this a large linear regression model with 158 parameters, and in general increasing the number of parameters in the model will also increase its capacity to overfit. Some common methods to help reduce overfitting include:

1) Obtaining additional training data
2) Reducing the number of features (i.e., give the model less parameters)
3) Selecting a simpler algorithm
4) Adding regularization

LASSO Baseline:

The difference between training and validation RMSE for the baseline LASSO model is significantly smaller than what was observed for standard linear regression (difference of 2.11 vs 6.04). The relative closeness of LASSO training and validation RMSE is evidence that the L1 regularization successfully reduced the model's error due to variance (i.e., reduced overfitting). Referencing the list of methods for reducing overfitting in the previous section, we note that there

are two properties of the L1 penalty which reduce model complexity and contribute to the observed reduction in variance:

1. Regularization to limit the magnitude of model parameters.
2. Feature selection to reduce the overall number of model parameters.

We found that the LASSO model with default regularization strength ($\alpha = 1$) selected (on average, across 5 folds) only 24 of the original 158 inputs and reduced parameter estimates for the remaining 134 inputs to zero. While this significantly reduced the model's error due to variance, the increased training RMSE (18.97) relative to that of the unregularized model (16.57) indicates that it has also increased the model's avoidable bias.

Ridge Baseline:

The baseline Ridge model achieved an average training RMSE of 16.57, and average validation RMSE of 22.44, a difference of 5.87 RMSE. This gap between training and validation performance indicates that the L2 regularized model is still overfitting to the training data. Comparing to the unregularized regression model we observe an even larger drop in performance when predicting on validation data (6.04 RMSE), which provides evidence that the L2 penalty has slightly reduced the models error due to variance. From the nearly identical training set errors, we see that this slight reduction in variance came with almost no impact to the model's avoidable bias.

Overall, the results from this baseline Ridge model are very similar to those from the standard linear regression. The reason these models are so similar is because at the default value of $\alpha = 1$, the impact of the L2 penalty term is quite small and therefore does not significantly decrease the parameter estimates relative to the unregularized model.

Elastic Net Baseline:

Elastic Net is a regularized linear regression model where the regularization term in the model's loss function is a weighted sum of the L1 (Lasso) and L2 (Ridge) penalties discussed previously. The training and validation performance of the baseline elastic net model shown in **Table 1** indicates that, relative to the unregularized linear regression, this model has:

1. slightly better validation performance
2. lower error due to variance
3. higher avoidable bias

Modeling

LASSO Model

$$(4) \qquad Lasso\ loss = SSE + L_1 Penalty = \sum_{i=1}^{n} (y_i - \hat{y})^2 + \alpha \sum_{j=1}^{p} |\beta_j|$$

To improve upon the baseline LASSO models performance, a hyperparameter search was performed to tune the model's regularization strength, denoted as α in equation 4. The search

evaluated 100 different regularization strengths, evenly spaced between 0.00501 and 20.8926 on a base 10 logarithmic scale. The logarithmic scale was selected over linear spacing to provide additional coverage of values at the lower end of the search range. For each regularization strength the LASSO model's performance was evaluated using 5-fold cross validation RMSE. The best performing model according to average validation RMSE across the 5-folds utilized a regularization strength of $\alpha = 1.5768$, and achieved an average validation RMSE of 20.7109.

Ridge Model

$$Ridge\ loss = SSE + L_2 Penalty = \sum_{i=1}^{n} (y_i - \hat{y})^2 + \alpha \sum_{j=1}^{p} \beta_j^2 \quad (5)$$

Similar to LASSO, we identified a reasonable value for the ridge model's regularization strength by evaluating the models 5-fold cross validation RMSE at each of 200 candidate α values. The first 50 regularization strengths were evenly spaced on a base 10 logarithmic scale between 0.005 and 1, and the next 150 values were linearly spaced between 1 and 2000. The best performing model utilized a regularization strength of $\alpha = 221.19909$ and achieved an average validation RMSE of 24.0979. Considering the baseline ridge model with default regularization strength of $\alpha = 1$ achieved a 5-fold cross validation RMSE of 22.4481, observing 24.0979 as the smallest RMSE across 200 different alpha values (which also included $\alpha = 1$) was an unexpected result.

Further investigation revealed that this a completely repeatable discrepancy which was the result of the data being shuffled prior to the baseline model cross validation, which was not performed during the grid search cross validation. While this discrepancy makes the result understandable, we believe it's worth quantifying how impactful this simple reordering of observations was on our performance metrics. Reperforming 5-fold cross validation on the model identified by grid search ($\alpha = 221.19909$) with the same shuffling strategy used for the baseline evaluations resulted in an average validation RMSE of 19.822, a decrease of 4.2750 RMSE (17.8227% decrease).

Elastic Net Model

$$ElasticNet\ loss = SSE + L_1 Penalty\ +\ L_2 Penalty = \sum_{i=1}^{n} (y_i - \hat{y})^2 + \alpha \left[l1ratio \sum_{j=1}^{p} |\beta_j| + (1 - l1ratio) \sum_{j=1}^{p} \beta_j^2 \right] \quad (6)$$

Equation 6 shows that the penalty term in the Elastic Net model's loss function is controlled by two hyperparameters, the regularization strength, α, and elastic net mixing parameter, l1 ratio. The regularization strength is used to scale the penalty, where a larger α corresponds to a greater calculated loss for a given set of model parameters. The elastic net mixing parameter is a proportion that controls the relative strengths of the L1 and L2 penalties. An l1 ratio of zero results in a pure L2 (ridge) penalty, while an l1 ratio of one result in a pure L1 (LASSO) penalty. This means the elastic net model has the flexibility to produce any combination of the LASSO and Ridge models discussed previously, α held constant.

We selected values for α and l1 ratio from a search space of 275 regularization strengths and 11 elastic net mixing parameters, for a total of 3025 unique hyperparameter settings. The first 25 regularization strengths were evenly spaced from 0.01 to 1 on a base 10 log scale, and the

remaining 250 were linearly spaced between 1 and 50. Nine of the eleven l1 ratio values were evenly spaced between 0.1 and 0.9, and the remaining two were at the extremes of 0.01 and 0.99.

The 3025 candidate models were evaluated using 5-fold cross validation and ranked according the average validation RMSE. The best performing model utilized an $l1ratio = 0.99$ and $\alpha = 1.689$, and achieved an average validation RMSE of 20.718.

We also note that the best elastic net found by grid search is one that utilized the largest l1 ratio in the search space (l1 ratio = 0.99), which means it is almost entirely a LASSO model. Additionally, the regularization strength of ($\alpha = 1.689$) and average validation RMSE (20.718) are both very similar to the best performing LASSO model which had ($\alpha = 1.5768$), RMSE=20.7109.

Model Improvements

The work presented in this section was performed with all model hyperparameters held fixed at the values identified during hyperparameter searches described above. With the individual models held constant, the following strategies were used to improve overall 10-fold cross validation RMSE:

1. Creating ensembles
2. Transforming the existing input features (e.g., yeo-johnson)
3. Creating new inputs from higher order polynomial versions of the original features.

The best overall model in terms of 10-Fold cross validation RMSE was a LASSO and Ridge ensemble. This ensemble utilized 2nd and 3rd degree polynomials and interactions terms for a subset of the original inputs. The polynomial term subset included the 20 features selected by the LASSO model, as well as the top five features according to both the LightGBM split and gain importance metrics, shown in **Table 3**. Finally, all model inputs were preprocessed through a yeo-johnson transformation. This best model achieved average 10-fold training and validation root mean squared errors of 14.47 and 14.59 respectively.

The realitvely small error rate and close agreement between the train and validation metrics indicate that this model has both lower bias and variance than any of the individual models previously considered. A summary table containing the 10-fold cross validation performance of the best individual models and 3 configurations of ensembles is provided in **Table 2**.

III Results

	Train RMSE (10-fold cv)	Validation RMSE (10-fold cv)
LASSO	19.41	20.44
Ridge	17.01	21.82
Elastic Net	19.50	20.39
Ensemble (LASSO, Ridge)	17.76	20.29
Ensemble (LASSO, Ridge) with yeo-johnson transform	16.40	16.44
Ensemble (LASSO, Ridge) with yeo-johnson transform and Polynomial Features	14.47	14.59

Table 2: Best Model Performance Summary

LightGBM Split	LightGBM Gain
Cu	Cu
wtd_mean_ThermalConductivity	Ca
Ca	Ca
wtd_entropy_Density	gmean_Valence
wtd_mean_Valence	wtd_gmean_Density

Table 3: LightGBM Feature Importance

Feature Importance

Best Ensemble Model Coefficients

Based on the parameter estimates for the LASSO and Ridge portions of the ensemble shown in **Figure 4** and **Figure 5**, and the LightGBM feature importance displayed in **Table 3,** we believe that the amount of copper in the material's chemical formula (feature **Cu**) is the most important factor towards generating accurate estimates of a superconductor's critical temperature. Specifically, **Figure 4** shows that three of the five largest magnitude coefficients from the LASSO model ($Ba^2Cu,\ Cu^3, Ca^2Cu$) are utilizing some form of the **Cu** input. In **Figure 5**, we again find **Cu** appearing as part of the second largest parameter in the Ridge model through the interaction

$wstdstdValence^2Cu$. Then, in **Table 3** we see that **Cu** is ranked as the most important feature under the LightGBM [2] split and gain feature importance metrics. Finally, recall from our exploratory data analysis section that **Cu** had the largest association with critical temperature under the Spearman correlation statistic but ranked only 28th among features under the linear Pearson correlation measure. The large Spearman correlation coupled with relatively small Pearson correlation suggested a strong non-linear relationship may exist between **Cu** and critical temperature, which we can now further observe through the large Cu^3 term in the LASSO model.

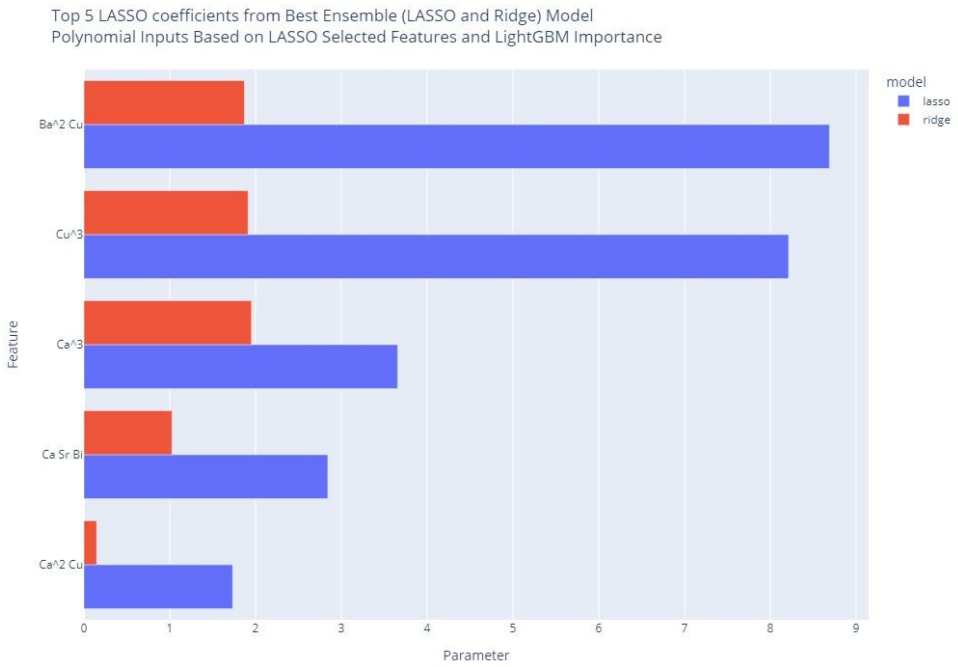

Figure 4: Top Features for LASSO Part of Ensemble

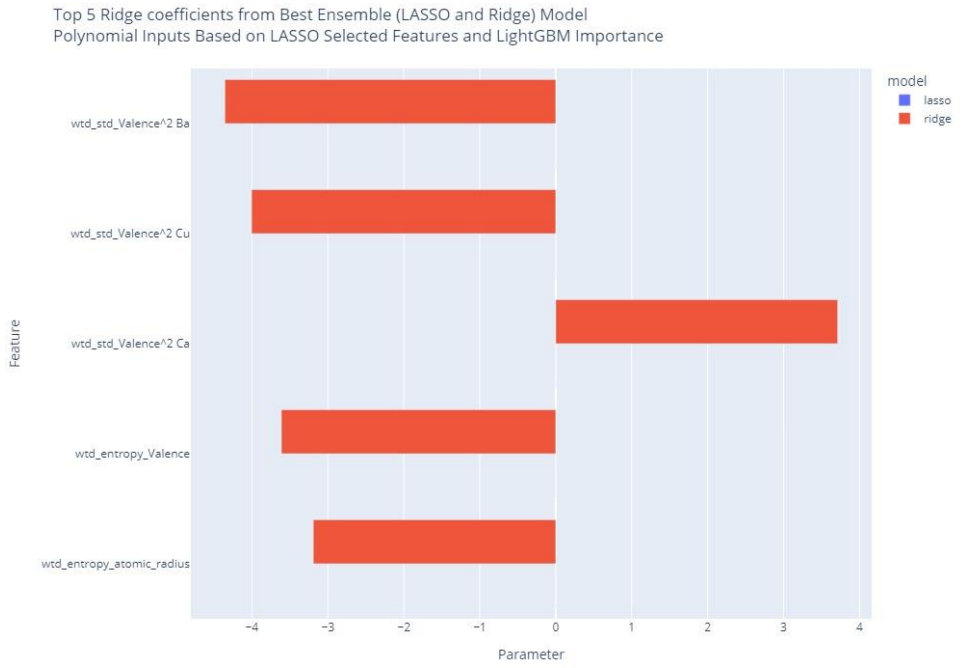

Figure 5: Top Features for Ridge Part of Ensemble

IV Conclusion

In this paper, we leveraged linear models to predict the critical temperature of a superconductor from its material properties. Due to the large feature space, L1 and L2 regularization techniques were used to combat overfitting by reducing both the growth of parameter estimates and the number of overall parameters in the model. Exploratory data analysis and feature importance measures suggested that non-linear associations may exist between the chemical properties of a superconductor and its critical temperature, which was experimentally confirmed through a reduction in both bias and variance for the models which utilized non-linear terms. As a final piece of information to anyone who may wish to continue this work, although the best performing models presented here utilized an ensemble of LASSO and Ridge combining the models is not required to reach the observed performance. In fact, when the LASSO model is removed from the best performing ensemble discussed above, performance improves even further (train RMSE = 13.599, Validation RMSE = 13.826). While this finding does not change our conclusion regarding the importance of the feature **Cu**, we hope it would be useful information to anyone who may wish to further this analysis.

V References

[1] *A data-driven statistical model for predicting the critical temperature of a superconductor*. https://arxiv.org/pdf/1803.10260.pdf. (n.d.)

[2] LightGBM: A Highly Efficient Gradient Boosting Decision Tree (neurips.cc)

V Appendix

Code

Please refer to the attached python source code and python notebook