2023

DS 7333 | Quantifying the World

CASE STUDY 1

JOEY HERNANDEZ
DANIEL CHANG

Table of Contents

Introduction

Background

Superconductivity is a fascinating quantum mechanical phenomenon where a material, under certain conditions, can exhibit zero electrical resistance, allowing electrical current to flow without loss of energy. The temperature at which a material transitions into a superconducting state is termed its "Critical Temperature". This parameter is crucial for applications involving superconductors, including energy storage, transportation, and advanced computing systems. Accurate prediction of the critical temperature for a given superconducting material can offer invaluable insights into material design, facilitate optimization processes, and accelerate technological innovations. However, predicting Critical Temperature has remained a significant challenge, given the complex interplay of variables such as material composition, structural features, and external conditions.

Objective and Scope

The objective of this case study is to develop a robust predictive model for the critical temperature (Tc) of superconducting materials. Specifically, we aim to construct a Linear Regression model optimized with L1 or L2 regularization techniques—or a combination of both, known as Elastic Net regularization—to predict Tc as accurately as possible. Regularization methods like L1 (Lasso) and L2 (Ridge) can prevent overfitting, thereby making the model generalizable to unseen data. Additionally, these techniques can assist in feature selection, an essential aspect when dealing with high-dimensional data sets commonly encountered in materials science.

Data Source

This case study will utilize two datasets, “train” and “unique_m”. The data was given to us in the Case 1 Study Module and is in the form of two separate csv files. When combined the data contains 21263 observations and 168 features.

Data Inspection

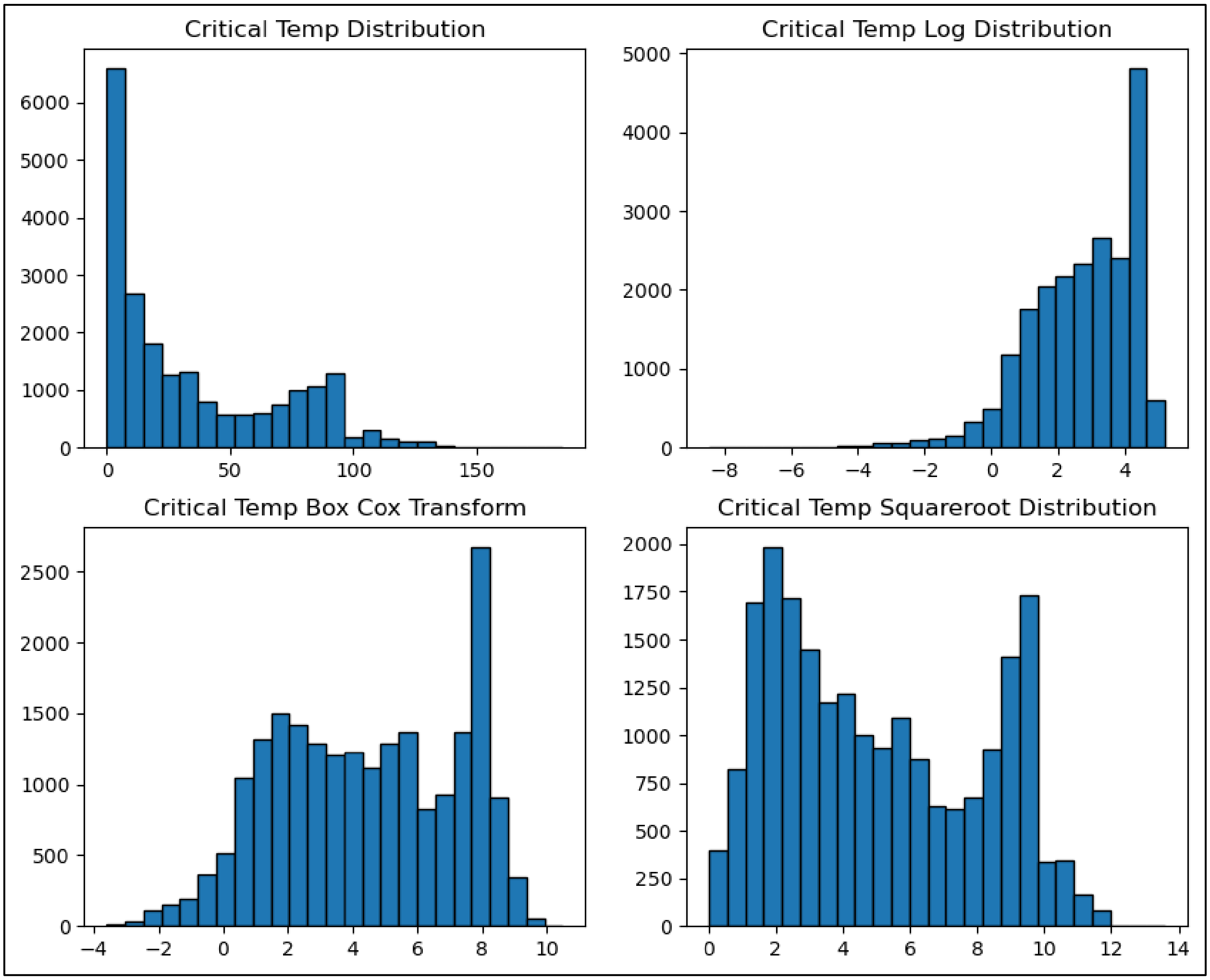
Before creating any models or analysis with the data our first step was to inspect our data to better understand data types (such as int, cat, object, etc.), distributions of values, identification of missing values, duplicated data, and outliers. This step is vital in understanding how we should approach any types of transformations or adjustments to the modeling and analysis process of our data.

Target Variable Inspection

To better understand the target variable distribution a visual inspection was performed using subplots containing various transformations of our target variable.

By plotting the histogram of the target variable in its original form, we gained insight into its inherent characteristics. However, a prominent right skew prompted the exploration of alternative transformations. The logarithmic transformation, Box-Cox transformation, and square root transformation were applied in attempt to get a more normally distributed target variable. Ultimately while none of the results were a textbook “normal” distribution, we proceeded with both normal data, and a Box Cox Transformation so that we can see performance discrepancies between various scenarios.

Figure 1: *Four-Quadrant Bar Plot Illustrating the Distribution of the Target Variable*



Correlation Plot (Original Target Variable)

Prioritizing the preprocessing steps for the target variable in the step prior to this was important because it allowed us to assess integrity and accuracy of subsequent analyses. By addressing the target variable's distribution and potential transformation needs, the resulting correlation values can be trusted to either accurately reflect the underlying relationships between variables or understand what limitations may arise from the less than desirable target variable distribution. Failure to preprocess and identify data discrepancies in the target variable could lead to misinterpretations, as correlations might be influenced by skewedness, outliers, or nonlinearities within the target data.

Next, performing a correlation heatmap provides a visually informative representation of the relationships between variables within a dataset. By illustrating the strength and direction of linear associations, the heatmap becomes an indispensable tool for uncovering patterns and dependencies that might not be immediately apparent from individual variable analyses. Each cell in the heatmap corresponds to a pair of variables, with the color gradient indicating the magnitude of correlation. This enables the rapid identification of high and low correlation values, highlighting potential areas of interest for further investigation.

Table 1: *Original Target vs. Explanatory Variables: Smallest Correlations*

Feature	Correlation Coefficient
Cs	-0.076822
Tc	-0.075295
std_atomic_radius	-0.071642
S	-0.071229
Er	-0.070134

Table 2: *Original Target vs. Explanatory Variables: Largest Correlations*

Feature	Correlation Coefficient
Pd	0.090037
Sb	0.072646
Ga	0.058372
Be	0.057971
Mg	0.055814

Table 3: *Transformed Target vs. Explanatory Variables: Smallest Correlations*

Feature	Correlation Coefficient
Er	-0.113125
Tc	-0.085807
B	-0.080060
Cs	-0.077309
S	-0.076124

Table 3: *Transformed Target vs. Explanatory Variables: Largest Correlations*

Feature	Correlation Coefficient
Pd	0.064658
Ga	0.050046
Sb	0.049051
wtd_range_ThermalConductivity	0.046485
Mg	0.045164

***Figure 2:** Example - Correlation Heatmap of Variables against original target*

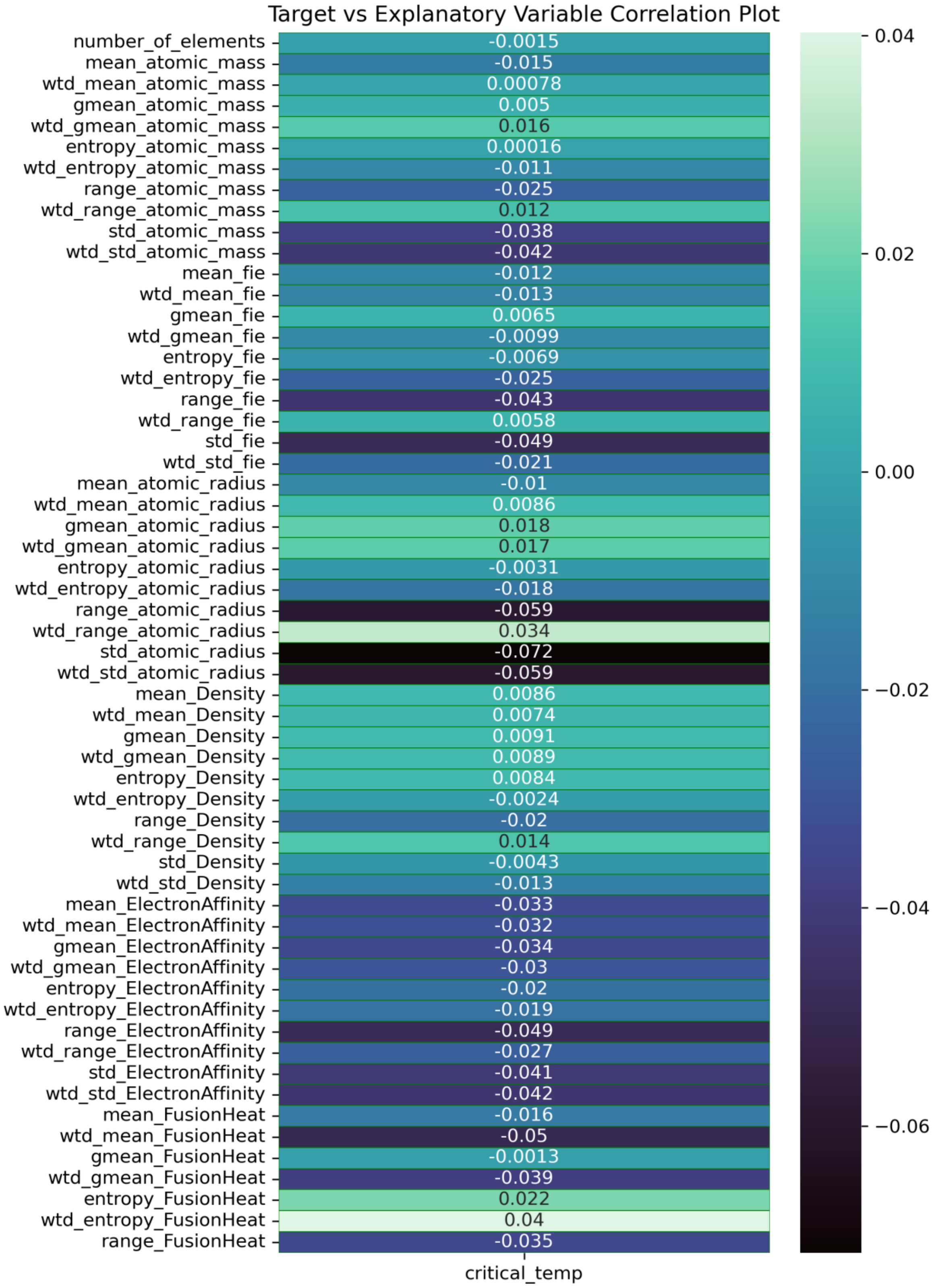
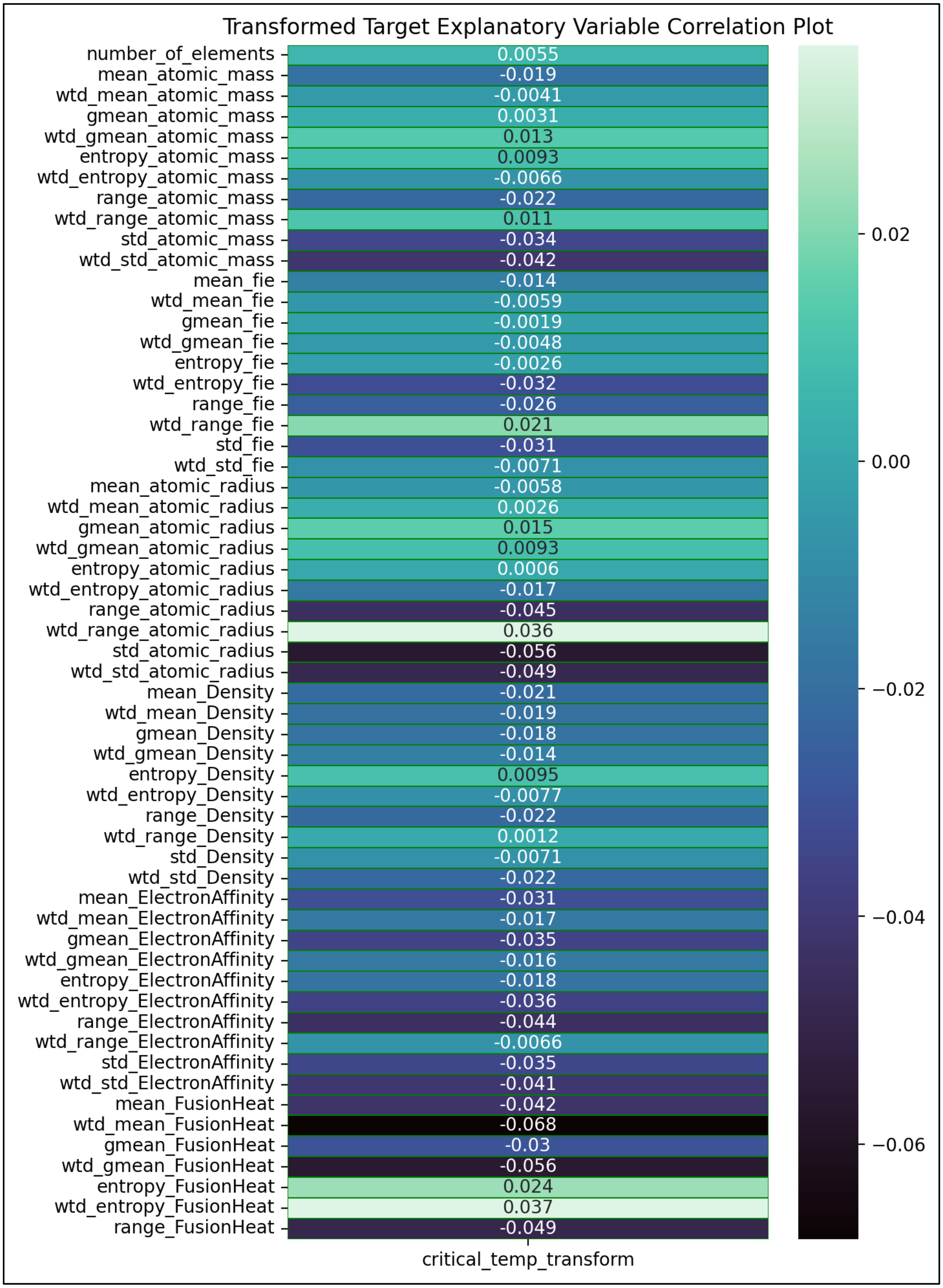


***Figure 3:** Example - Correlation Heatmap of Variables against Transformed Target*



Modeling

Lasso | L1 Regularization

Lasso Regression, also known as L1 regularization, is an extension of linear regression that not only seeks the best-fitting line through the data but also constrains the size of the coefficients.

In our mission to create the most accurate model to predict the Critical Temperature, we will be using the cross_val_score and cross_val_predict functions. These functions perform cross-validation, a technique that helps us understand how well the model will perform on unseen data. It does this by splitting the dataset into training and testing sets multiple times and averaging the performance across all splits. Additionally, we will utilize LassoCV, a tool designed to find the best "alpha" value, which is the tuning parameter that balances between fitting the data well and keeping the model simple. By identifying the optimal alpha, we aim to make our Lasso model as accurate and generalizable as possible.

Results

The findings of this analysis are highlighted in the results, including the best alpha value of 0.01. The cross_val_score - Cross-Validation Root Mean Squared Error (RMSE) - scores ranged from 16.203 to 18.756, with an average RMSE of approximately 17.593. This shows how the model performs across different dataset splits. Furthermore, the RMSE score derived from cross_val_predict is very close to the average RMSE, at around 17.616.

However, when we predict on the holdout set, we get a higher RMSE of 23.508161878655947.

Assumption Checks:

- Normal Distribution: In the section, Target Variable Inspection, we found that the distribution of the aforementioned variable is heavily right-skewed. Below in Figure 4, we can see that the points on the end of each side of the line are pretty far from the line, indicating that there is evidence of a few outliers.
- Independence: For this project, we will assume that the data is independent.
- Constant Variance: Below in Figure 5, we can see from that residual plot that the data points are all packed together when they should be scattered evenly throughout the plot. Therefore, we can conclude that the Constant Variance Assumption has been violated.
- Linearity: Looking at Figure 5, in order for this assumption to be met, the residuals should be randomly scattered around the horizontal line at zero. This means that there should be no clear pattern or trend in the residuals. However, we do not see this in the plot, so we can conclude that the assumption has been violated.

Figure 4: *QQ-Plot of Residuals(LASSO)*

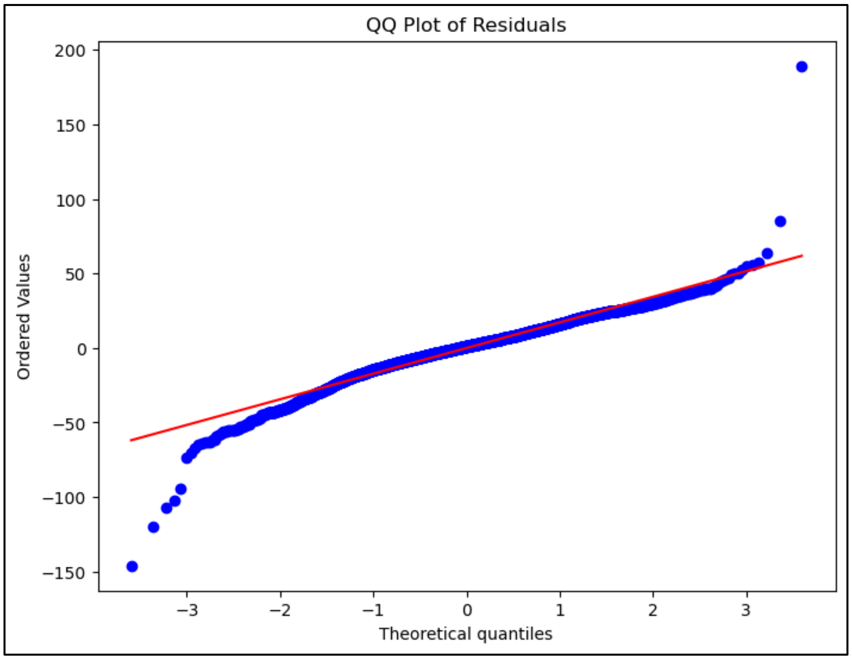
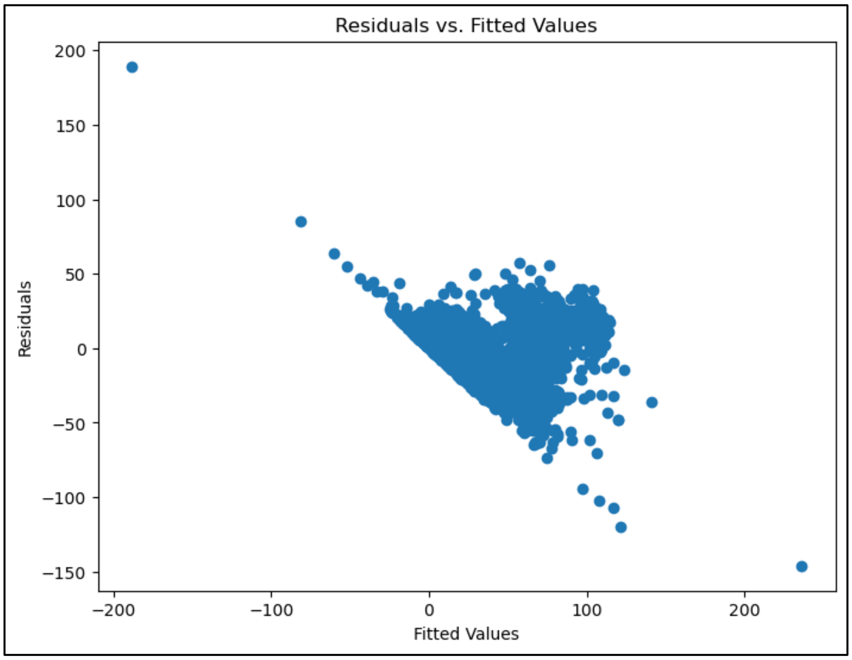


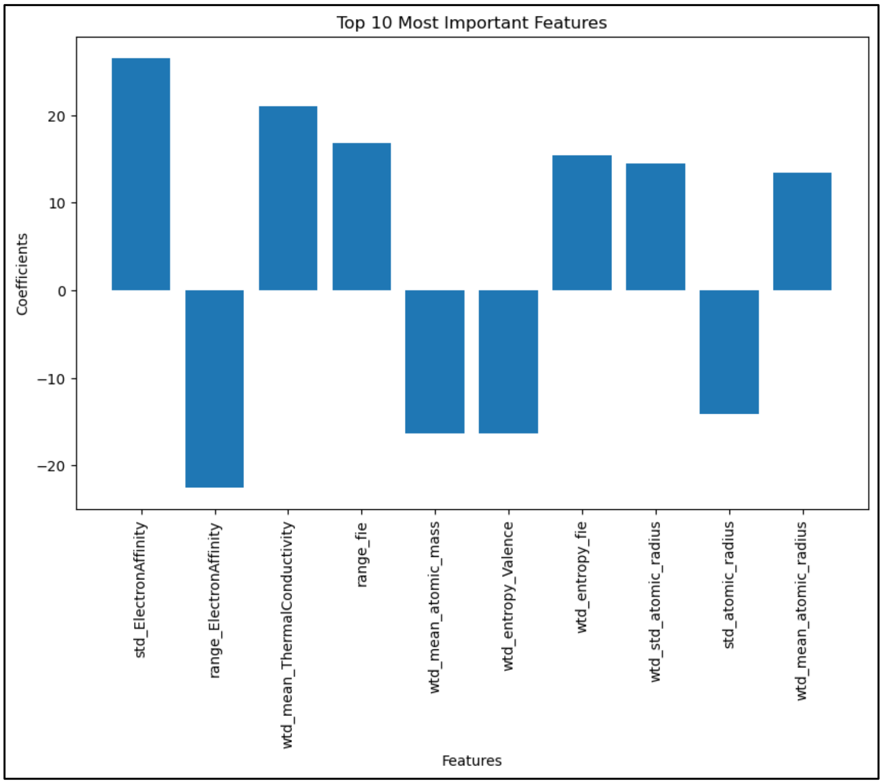
Figure 5: *Residual vs. Fitted Plot(LASSO)*



Feature Importance:

Here are the top 10 features and their coefficients for the Lasso Model.

Figure 6: *Feature Importance (LASSO)*



Ridge | L2 Regularization

Ridge Regression, also known as L2 regularization, is an extension of linear regression that aims to find the best-fitting line through the data and control the coefficients' magnitude. Unlike the Lasso, it doesn't drive the coefficients toward exactly 0.

In our pursuit of the most accurate model to predict the Critical Temperature, we will use cross-validation, a technique that provides insight into how well the model will perform on unseen data, with the cross_val_score and cross_val_predict functions. This entails repeatedly splitting the dataset into training and testing sets and averaging the performance across all splits. In addition, we will use RidgeCV.

Results

The results of our Ridge Regression are in, with the best alpha identified as 0.01. The corresponding cross-validation Root Mean Squared Error (RMSE) scores provide a comprehensive picture of the model's performance across different splits of the data: [16.93111055, 17.75577535, 18.70122445, 16.1260031, 17.88780085]. The average RMSE across all folds is calculated to be 17.48, with a comparable value of 17.50 achieved using cross_val_predict. These findings reinforce the efficacy of Ridge Regression in achieving accurate predictions while managing coefficient magnitudes and the cross-validation techniques utilized to emphasize the model's robustness and suitability for real-world applications.

However, when we predict the holdout set, we get a higher RMSE of 24.77233804560577.

Assumption Checks:

As seen below, we can see that the qq-plot and the residual plot look identical to that of our LASSO Regression model. Therefore, we can conclude that the assumption check conclusions are the same.

Figure 7: *QQ-Plot of Residuals(Ridge)*

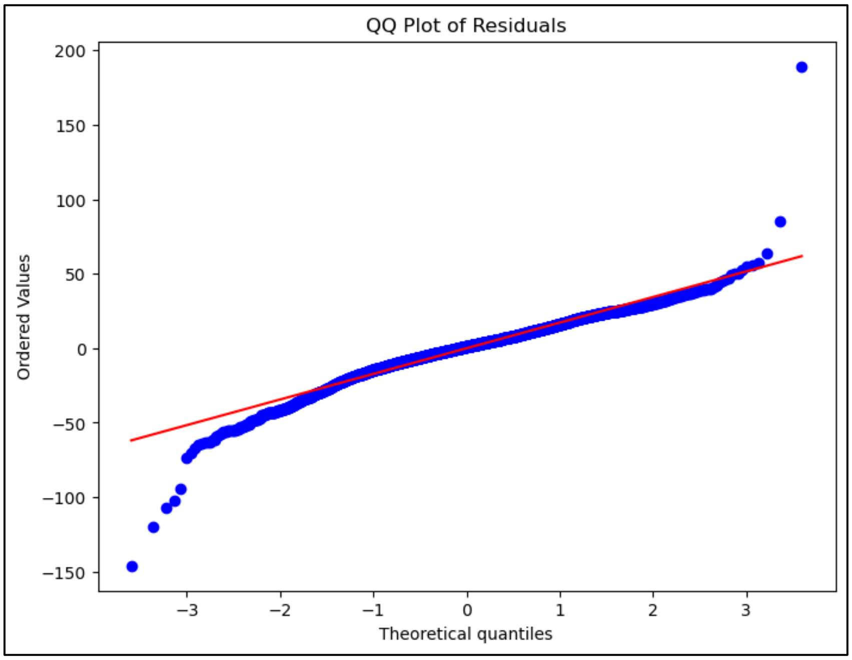
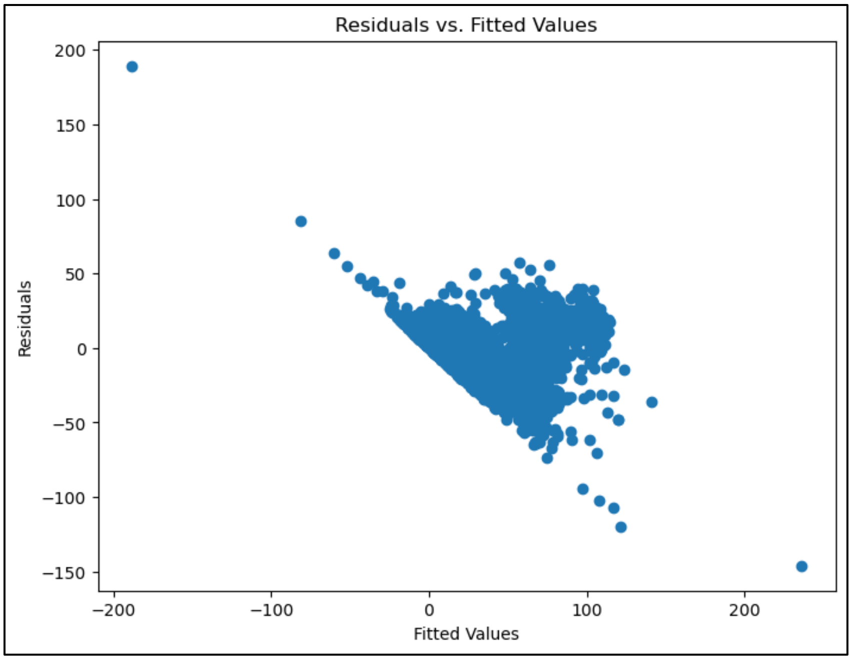


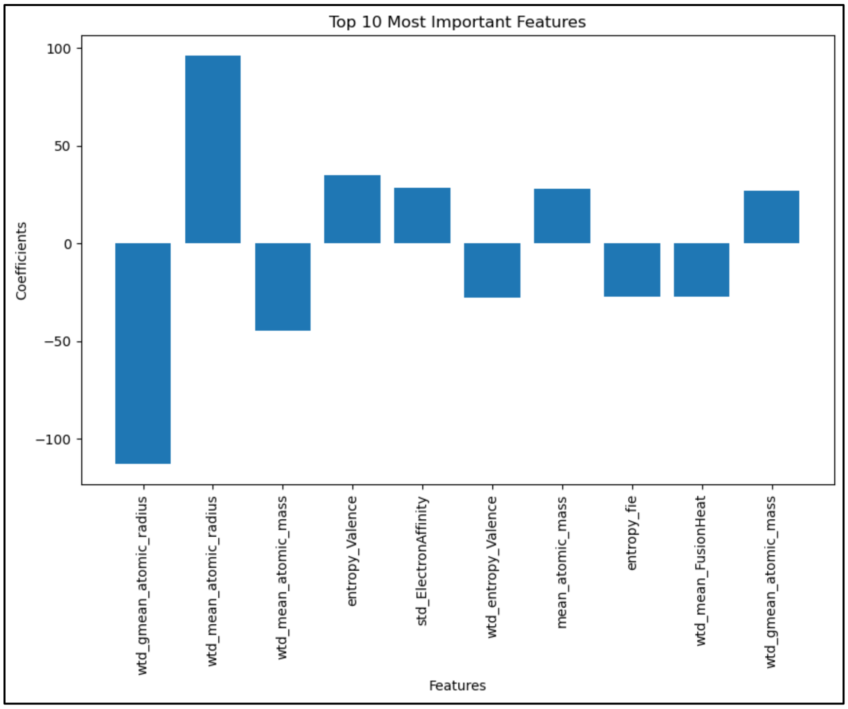
Figure 8: *Residual Plot(Ridge)*



Feature Importance:

Here are the top 10 features and their coefficients for the Ridge Model.

Figure 9: *Feature Importance (Ridge)*



Linear Regression with Elastic Net Feature Selection (Transformed Target)

Linear Regression with Elastic Net is a powerful regression technique that combines the regularization methods of Lasso (L1) and Ridge (L2). Elastic Net effectively balances the benefits of feature selection (Lasso) and coefficient magnitude control (Ridge) by incorporating both penalties, making it suitable for high-dimensional datasets. This method aids in the identification of important features while avoiding overfitting, resulting in a more interpretable and robust regression model.

Like what we did with the LASSO and Ridge models, we will utilize the cross_val_score and cross_val_predict functions for cross-validation. In addition to the optimal alpha, we will also be finding the optimal l1_ratio, which determines the balance between the Lasso (L1) and Ridge (L2) penalties applied to the model's coefficients. The l1_ratio value ranges between 0 and 1, where 0 corresponds to Ridge regression (only L2 penalty), and 1 corresponds to Lasso regression (only L1 penalty). Intermediate values of l1_ratio allow a combination of both penalties, offering a flexible approach that captures the strengths of both regularization methods.

It's noteworthy that the target variable has undergone a Box-Cox transformation. The Box-Cox transformation is a statistical technique used to stabilize variance and make the data more closely resemble a normal distribution. This transformation is particularly useful when dealing with data that violates assumptions of normality or homoscedasticity, common in linear regression.

Results:

The results from the cross-validated Linear Regression and Elastic Net models reveal consistent performance across multiple folds. Each fold's root mean squared errors (RMSE) indicate relatively low variations, with values ranging from 1.3381 to 1.4104 (Note that these values are not on the Original scale). This suggests that both models provide stable predictions across different subsets of the data.

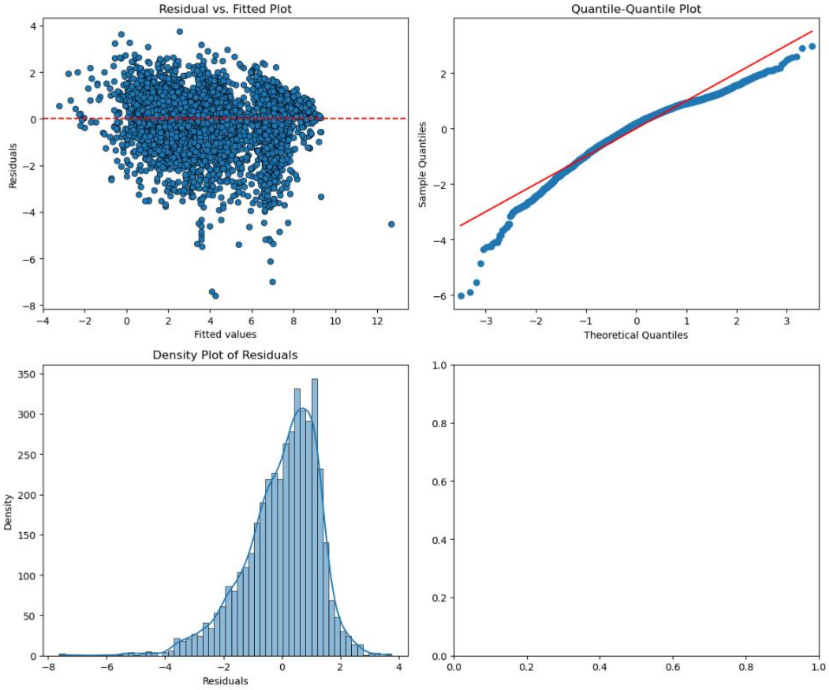
The Linear Regression with Elastic Net technique (after inverse transforming the target variable), combining Lasso (L1) and Ridge (L2) regularization, achieves a training RMSE of 16.8308 and a test RMSE of 17.6238.

Assumption Checks:

- Normal Distribution: In Figure 10, in the histogram below, the data is skewed to the left. In the QQ Plot, we can see that the points on the end of each side of the line are pretty far from the line, indicating that there is evidence of a few outliers. However, we can largely say that the normal distribution assumption has been met.
- Independence: For this project, we will assume that the data is independent.

- Constant Variance: In the Residual vs. Fitted plot, our data points look much better than what we had gotten earlier with the Ridge and LASSO, but we would still hesitate to say
- Linearity: In order for this assumption to be met, the residuals should be randomly scattered around the horizontal line at zero. This means that there should be no clear pattern or trend in the residuals. However, we do not see this in the plot as it produces one big blob in the center, so we can conclude that the assumption has been violated.

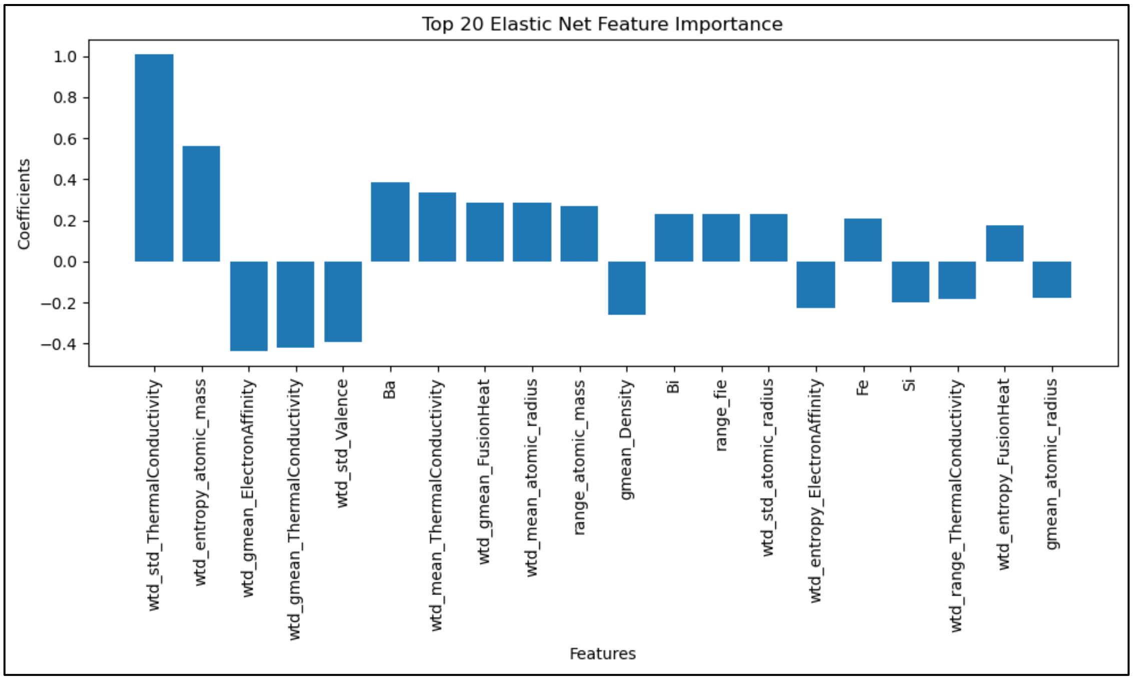
Figure 10: *Assumption Checks (Elastic Net)*



<u>Feature Importance:</u>

Here are the top 20 features and their coefficients for the linear regression model using Elastic Net for Feature selection and extraction.

Figure 11: *Feature Importance (Elastic-Net)*



Linear Regression with LASSO Feature Selection (Transformed Target)

This section will repeat the above steps without the Ridge regularization. Linear Regression with LASSO Feature Selection is a potent regression technique that harnesses the Lasso (L1) regularization method. By applying the L1 penalty, LASSO effectively achieves both feature selection and coefficient magnitude control. This renders it especially adept at handling high-dimensional datasets. This approach identifies key features while mitigating overfitting, culminating in an interpretable and resilient regression model.

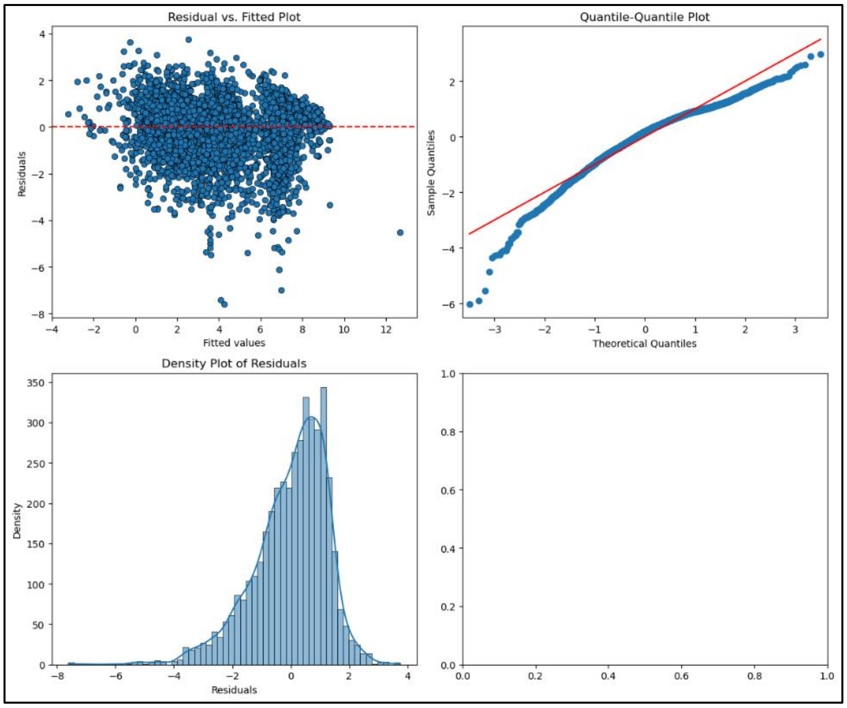
Results

Following the procedure, we observed that the training root mean squared error (RMSE) on the original scale was 16.3393, while the test RMSE was 18.6762. This implies that the model performed well on the training data, but slightly worse on the unseen test data. This outcome indicates that the model could be slightly overfitting to the training data.

Assumption Checks:

As seen below, we can see that the QQ-plot and the residual plot look identical to our Linear Regression model with Elastic Net Feature Selection. Therefore, we can conclude that the assumption check conclusions are the same.

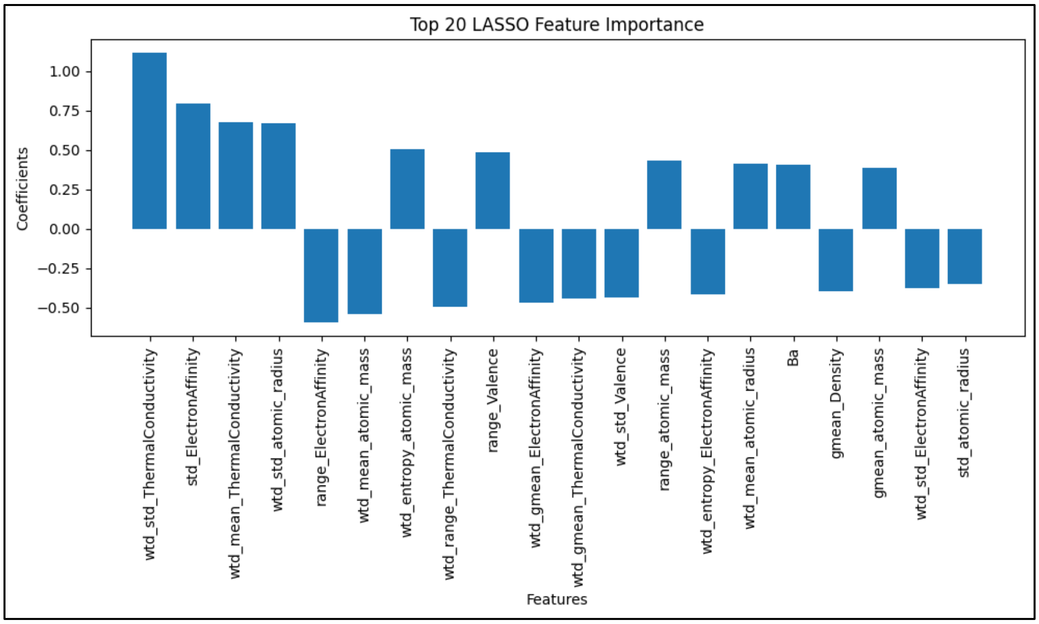
Figure 12: Assumption Checks (LASSO)



Feature Importance:

Here are the top 20 features and their coefficients for the Linear regression model using Lasso for feature selection and extraction.

Figure 13: Feature Importance (LASSO)



Conclusion

The initial model employed Lasso Linear Regression on the raw data, resulting in an RMSE of 17.6157. During validation, the model demonstrated accurate predictions and a keen grasp of data patterns. However, it encountered challenges in generalizing to unseen data, evident by its higher RMSE of 23.5082 on the holdout set.

Similarly, the Ridge Linear Regression model was implemented on the original dataset, yielding an RMSE of 17.5024. The model showcased commendable performance during validation, effectively capturing underlying relationships. However, its ability to predict unseen data was strained, as indicated by the elevated RMSE of 24.7723 on the holdout set.

The Elastic Net Linear Regression model adopted a transformed target variable and achieved an RMSE of 17.6238 during validation. Demonstrating consistency across folds, the model exhibited robustness by striking a balance between L1 and L2 penalties. This equilibrium allowed it to manage feature selection and coefficient control better.

The Lasso Linear Regression model, operating with a transformed target variable, recorded an RMSE of 18.6762 during validation. While proficient in feature selection and coefficient regulation, like its counterparts, it encountered a marginal uptick in RMSE during holdout validation, implying that its predictive power may suffer when tested on unseen data.

Ultimately, the array of linear regression models, each embracing distinct regularization approaches, yielded insights into data intricacies. Their success in capturing relationships and managing model complexity was evident. Yet, as they ventured into uncharted territory, the increase in RMSE underscored the need for further refinement to enhance their capacity to generalize effectively.

Recommendations

1) **Feature Engineering and Selection**: Using domain knowledge to engineer meaningful features can improve model performance significantly. Investigating interactions, polynomial terms, and domain-specific transformations may reveal hidden patterns in the data. Furthermore, feature selection techniques other than regularization, such as mutual information or recursive feature elimination, can help identify the most influential features. Experimenting with different combinations of features and selection methods iteratively can result in a more refined set of inputs for the models.

2) **Experimenting with Different Models:** Experimenting with models other than linear regression, such as decision trees, support vector machines, random forest or gradient boosting, can provide new insights and potentially improved performance. Each model has strengths and weaknesses, and experimenting with various approaches can reveal the best-fit approach for the specific dataset. Furthermore, neural networks and deep learning architectures can capture complex relationships that traditional models may overlook. Iteratively testing and comparing different models' performance can result in a more accurate and versatile predictive system.