Superconductivity: Exploratory Data Analysis and Feature Extraction with Common Regression Algorithms

Xiaozhu Zhang

1.Introduction

There is no free lunch. This saying is true in the field of statistical learning as well. There is not a single perfect algorithm that always outperforms other algorithms; every technique has its strength and weakness. Therefore, it is important for us to inspect the datasets first before feeding it to the algorithm, as well as to consider the tradeoff between interpretation and prediction. In this paper, we will explore how some of the common regression algorithms work in a high dimensional space setting by analyzing the dataset *superconductivity.* First, we will conduct some exploratory data analysis to better understand the relationship between the features and outcome(critical temperature). After a preliminary inspection of the data, we will proceed to compare the performance of each regression algorithm on predicting the outcome. And finally, we will take a look at which features are deemed important by these algorithms.

2. Analysis

A. Dataset

The *superconductivity* dataset(training) contains 82 features(81 predictors, 1 outcome) and 21263 instances. All of the features are numerical(contains both discrete and continuous), and there is no missing value.

B. The outcome variable: critical temperature

Before diving into investigating predictor variables, it can be helpful to take a look at the distribution of the outcome variable in case of abnormality. As shown in Figure 1, we can see from the density plot that there are two peaks(one around zero, the other is around 100). The boxplot suggests that most of the instances have critical temperatures between 0 and 70(from upper tail to 3rd Quartile) The outcome variable looks fine and we can proceed to examine the predictors.

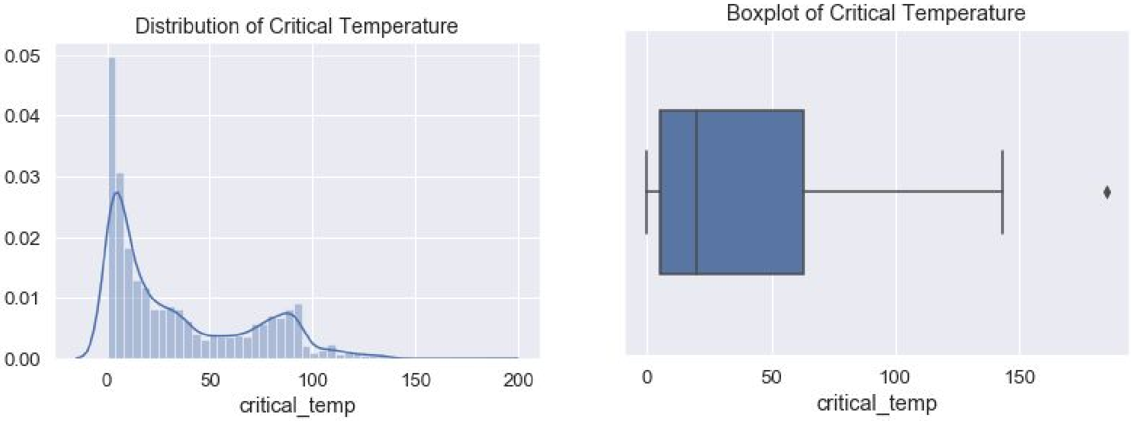
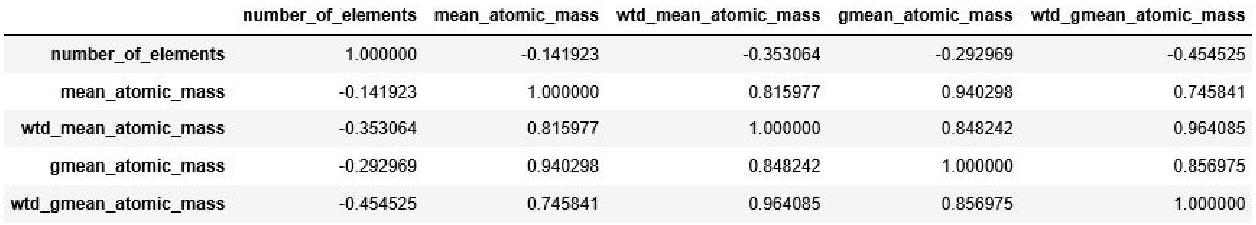


Figure 1: Density plot and Boxplot of Critical Temperature(outcome)

C.Correlation between predictor variables

In order to obtain an overview of the correlation between the attributes, we can calculate the correlation matrix and plot it as a heatmap. Due to the large number of predictors, Table 1 shows part of the correlation matrix, we can see many numbers are close to 1, which suggest these predictors are highly correlated with one another. This leads us wondering whether this happens to the other predictive variables as well. If so, we may face the problem of multicollinearity. Figure 2 shows the correlation heatmap, and it appears that the attributes that are highly correlated to the outcome variable are also highly correlated with each other.



	number_of_elements	mean_atomic_mass	wtd_mean_atomic_mass	gmean_atomic_mass	wtd_gmean_atomic_mass
number_of_elements	1.000000	-0.141923	-0.353064	-0.292969	-0.454525
mean_atomic_mass	-0.141923	1.000000	0.815977	0.940298	0.745841
wtd_mean_atomic_mass	-0.353064	0.815977	1.000000	0.848242	0.964085
gmean_atomic_mass	-0.292969	0.940298	0.848242	1.000000	0.856975
wtd_gmean_atomic_mass	-0.454525	0.745841	0.964085	0.856975	1.000000

Table 1: A peek of the correlation matrix

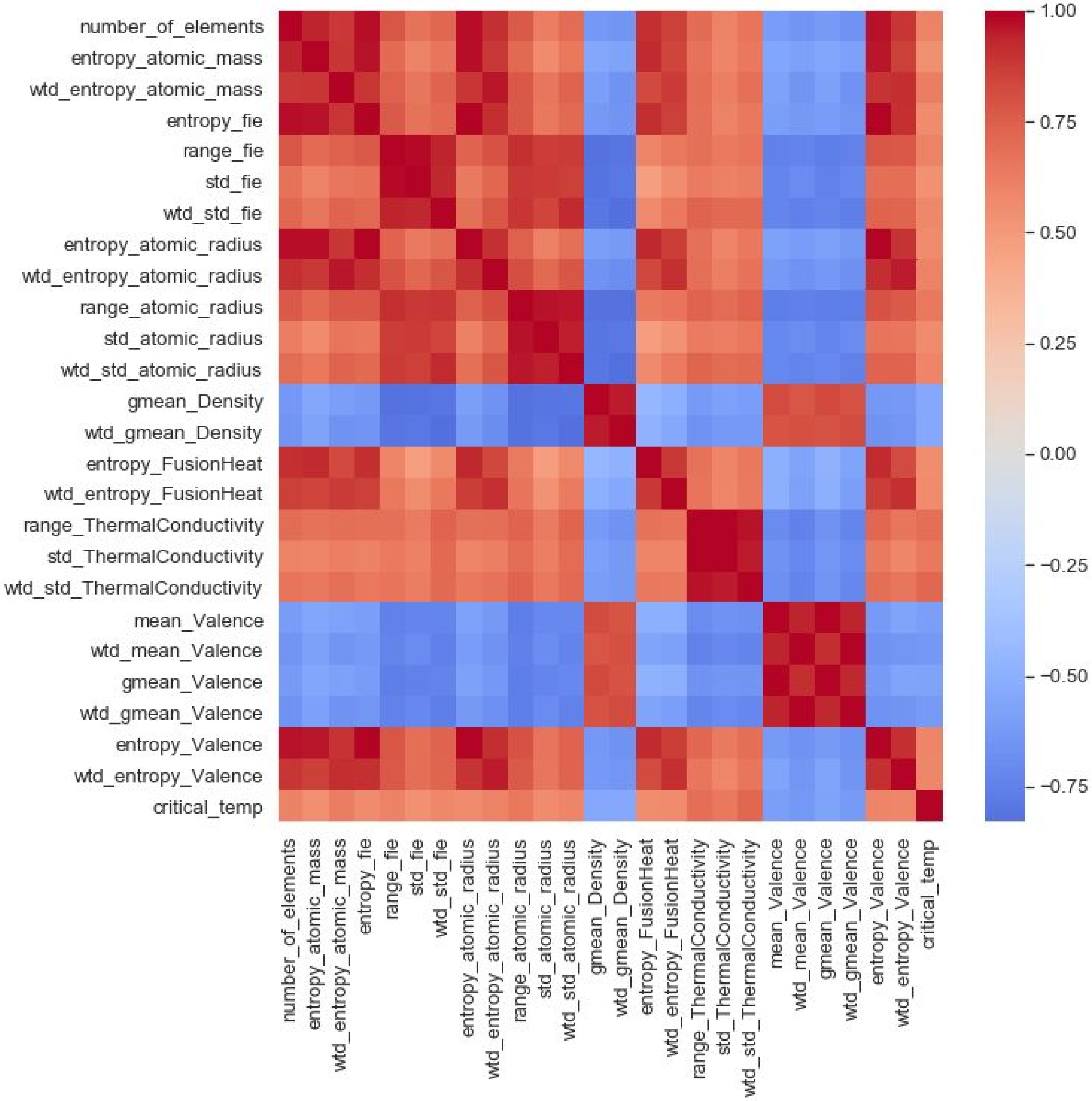


Figure 2: Correlation heatmap of some attributes (Note that because it can be hard to read the heatmap when we plot all of the predictor variables, the selected attributes in this heatmap are highly correlated to the outcome variable, specifically, the correlation of these attributes with the outcome variables are above 0.5.)

D.Principal component analysis

In the previous section, we suspect the existence of multicollinearity based on the correlation matrix. To confirm our suspicion, we can apply a common dimension reduction technique PCA to this dataset after the procedure of standardization(we standardize the predictive attribute to eliminate the effect of different units.) Figure 3 shows the explained variance ratio as the number of components increases. Based on the plot, we can see that with approximately twenty components we can explain over 90% of the variance. This confirms our suspicion that a lot of the features provide the same information. However, for the purpose of this paper, because we would like to predict as well as to retain the interpretability of each feature, we will use the original data instead of reducing its dimension.

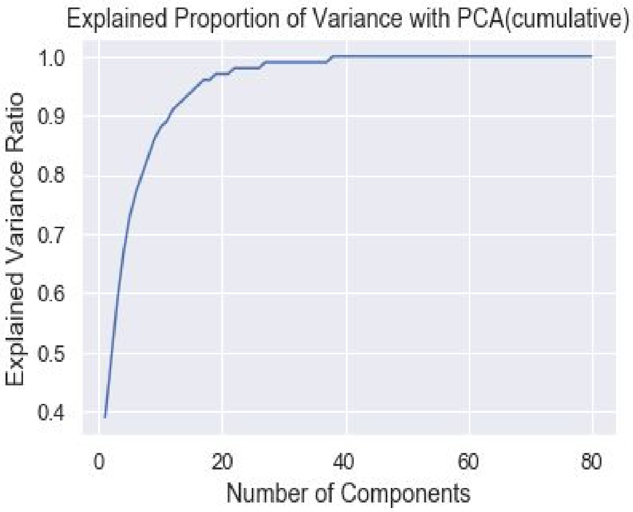


Figure 3:Percent of Variance Explained By Number of Components(cumulative)

3. Prediction and Feature Extraction

A.Prediction

With the previous findings in mind, this section will be focusing on prediction and comparison of the performance of some common regression algorithms. The basic linear regression will be used as baseline.

The regression algorithms that will be discussed here include techniques that utilize regularization such as Ridge and LASSO, as well as tree-based methods such as decision trees and random forest.

	Linear regression	Ridge	LASSO	Decision Tree	Random Forest
Score	0.73(+/- 0.03)	0.73(+/- 0.03)	0.72 (+/- 0.02)	0.83 (+/- 0.04)	0.91 (+/- 0.01)

Table 2: Performance of Linear Regression, LASSO, Ridge, Decision Tree and Random Forest.

Table 2 shows the performance of the different regression algorithms. The criteria used for evaluation of performance is R square(ie.score in sklearn package.) The results shown in Table 2 is the average score over 5-fold cross validation and their 95% confidence interval. We can see that compared to linear regression, Ridge and LASSO which utilize regularization have similar performance, whereas tree-based algorithms have significantly better performance. This suggests that there might be some non-linear nature in the superconductivity data.

B. Extract Features in Depth

In addition to comparing the prediction performance for each variable, we are also interested in what features are deemed important(or influential) by these regression techniques. We know that LASSO and Ridge are linear models that utilize shrinkage, and specifically LASSO can shrink the coefficients of unimportant features to 0, so they can perform feature selection as well. In this section, we will take a look at the similarities and differences of the features that are deemed as important by linear methods(Ridge and LASSO) and non-linear methods (Decision tree and random forest.) The features with larger coefficients should be more influential at predicting.

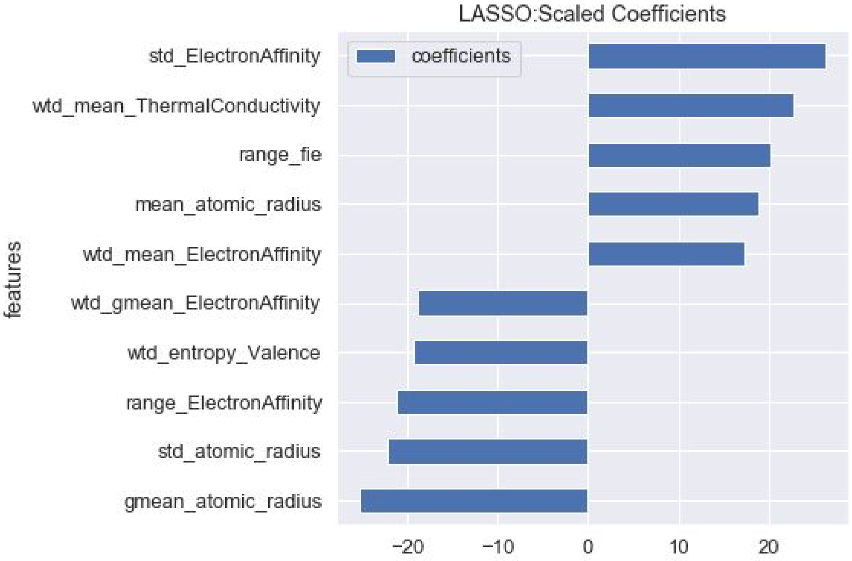


Figure 4: Top 10 features selected by Lasso

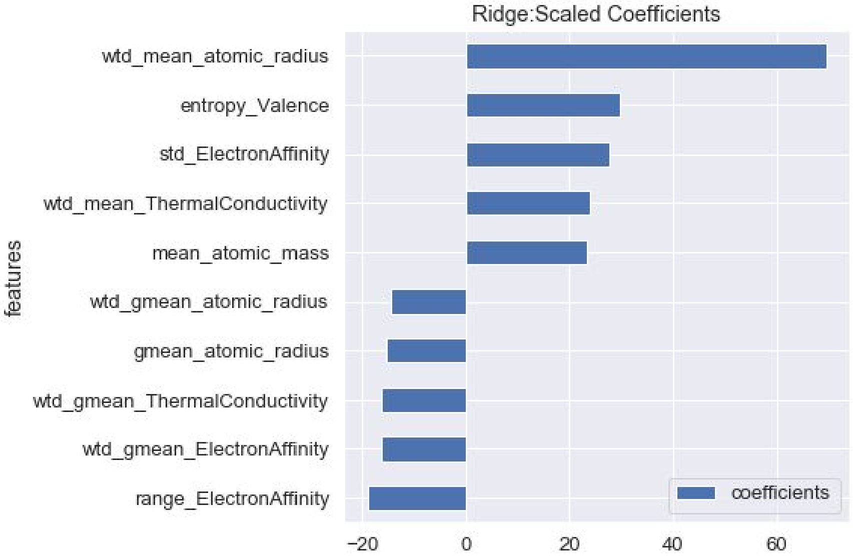


Figure 5: Top 10 features selected by Ridge

From Figure 4 and Figure 5, we can see the top 10 influential features selected by LASSO and Ridge. Of these features, some of them are selected by both techniques such as range_ElectronAffinity, wtd_gmean_ElectronAffinity, gmean_atomic_radius, wtd_mean_ThermalConductivity, and std_ElectronAffinity. Specifically, increases in wtd_gmean_ElectronAffinity, range_ElectronAffinity, wtd_gmean_ElectronAffinity, or gmean_atomic_radius result in *decrease* in the outcome(critical temperature), whereas increases in std_ElectronAffinity results in *increase* in critical temperature.

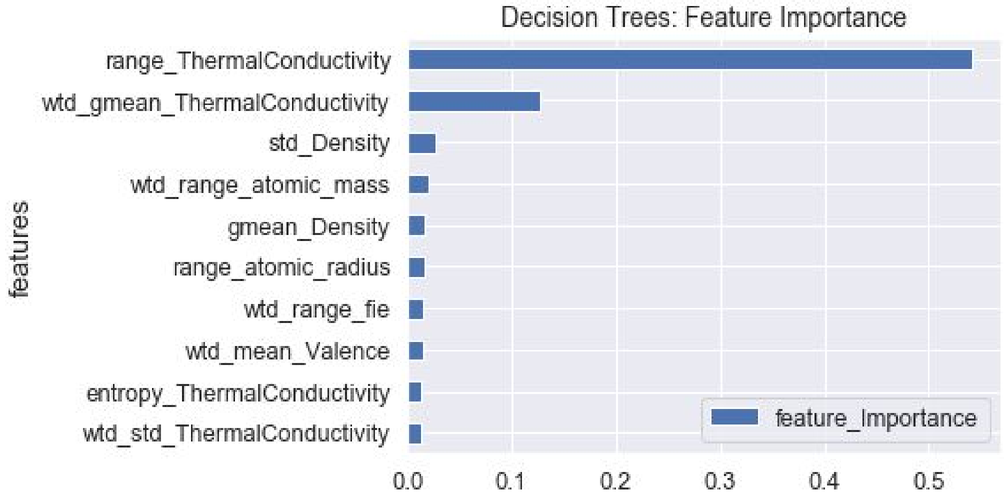


Figure 6: Top 10 features selected by Decision Tree

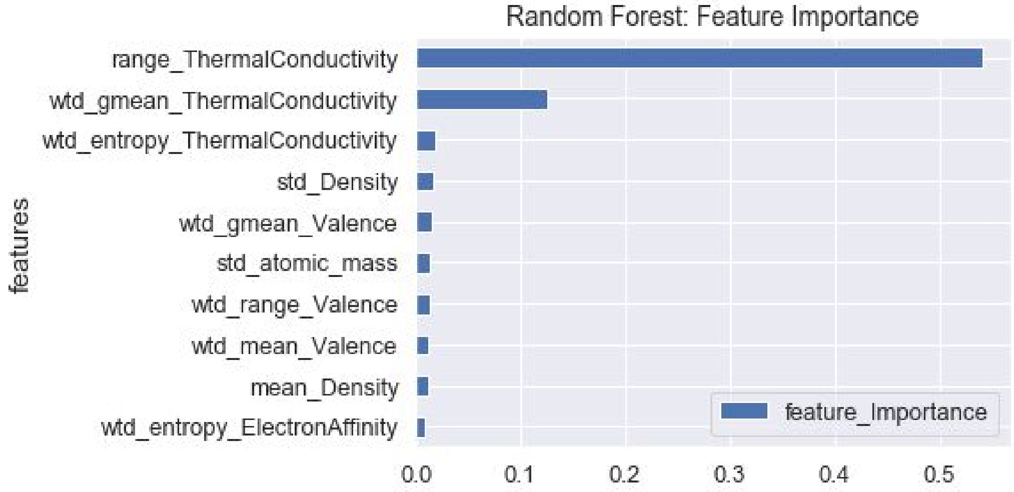


Figure 7: Top 10 features selected by Random Forest

From Figure 6 and Figure 7, we can see the top 10 influential features selected by Decision Tree and Random Forest. These features are selected through the *feature_importances_ in* command in scikit-learn. Since tree-based algorithms are non-linear, we are not able to interpret how each feature directly affects the outcome. Nevertheless, we can still obtain some insights about the features. For example, range_ThermalConductivity and wtd_gmean_ThermalConductivity are deemed as important features by both decision trees and random forest.

After knowing what features are considered as important, let us see how they are related to the outcome variable. After a quick observation of Figure 8 and Figure 9, it seems these predictor variables are not strongly correlated with the critical temperature. This might explain why the non linear model(tree-based) performs much better than the linear models(Ridge and Lasso.)

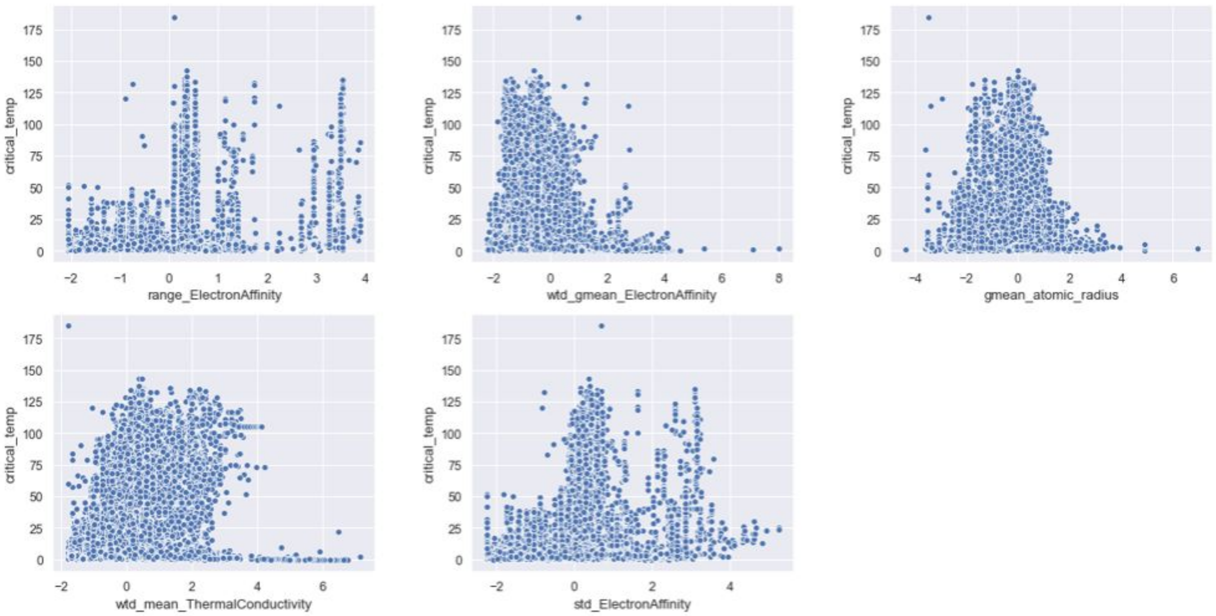


Figure 8: Scatterplot of Critical Temperature vs Features Selected by Ridge and LASSO

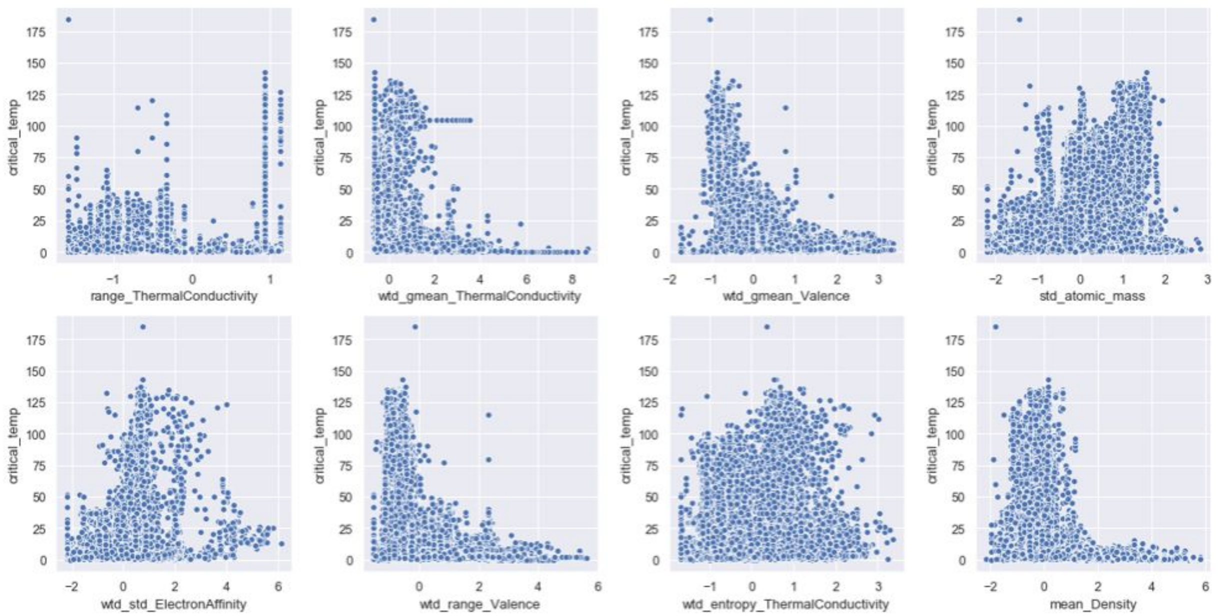


Figure 9: Scatterplot of Critical Temperature vs Feature Selected by Decision Tree and Random Forest

4. Discussion

In summary, our linear models including linear regression, LASSO and Ridge exhibit good performance at predicting for the superconductivity dataset. However, non-linear methods(tree-based) give a much more accurate prediction compared to linear methods. This might suggest there is some non-linearity lying in the structure of this dataset and non-linear methods may be more powerful in prediction at the cost of interpretability. We should also note that despite the fact that random forest gives the best result, it is also the most computationally-expensive algorithm(takes the longest time to train the model). Therefore, when working with the superconductivity dataset, if one was to choose a moderately accurate and less computationally less expensive algorithm, the decision tree for regression may be a suitable choice.