Superconductivity Exploratory Data Analysis and Prediction

Christina Chang

***Abstract* — The purpose of this report is to predict the critical temperature of superconductors. The performance of five different regression models was compared by using cross validated R2 scores and cross validated root mean squared error. Overall, the random forest regression model is better suited for this problem when compared to linear regression, support vector regression, multi-layer perceptron regression, decision tree regression, and k-nearest neighbors regression. The hyperparameters of the random forest regression model were tuned and the model had a 10-fold cross validated R2 score of 0.75 and a 10-fold cross validated root mean squared error of 11.61. Some important features for determining the critical temperature of a superconductor were atomic radius and thermal conductivity.**

I. Introduction

Superconductivity is when a charge moves through a material without resistance [1]. Superconductive materials are important in electronics because they help to make circuits compact, therefore, they make them more energy efficient [1]. However, superconductors are only useful below a certain critical temperature [2]. At very low temperatures, the superconductor may even have infinite conductivity, or zero electrical resistance [2]. This means that the electrons or ions can move through the material with no disruption [2]. The motivation of this project is to create a regression model that predicts the critical temperature of superconductors. If the regression model performs well, it can be used to determine which superconductors will work well in an electrical appliance or other environment.

The superconductivity data set is from the UCI Machine Learning Repository, and it includes 21,263 observations of superconductors. Each observation has 168 attributes; 81 attributes describe the features of the superconductor and 86 attributes describe the chemical formula of the superconductor. The last feature is the critical temperature of the superconductor. All the features in this data set are numerical.

II. Exploratory Data Analysis

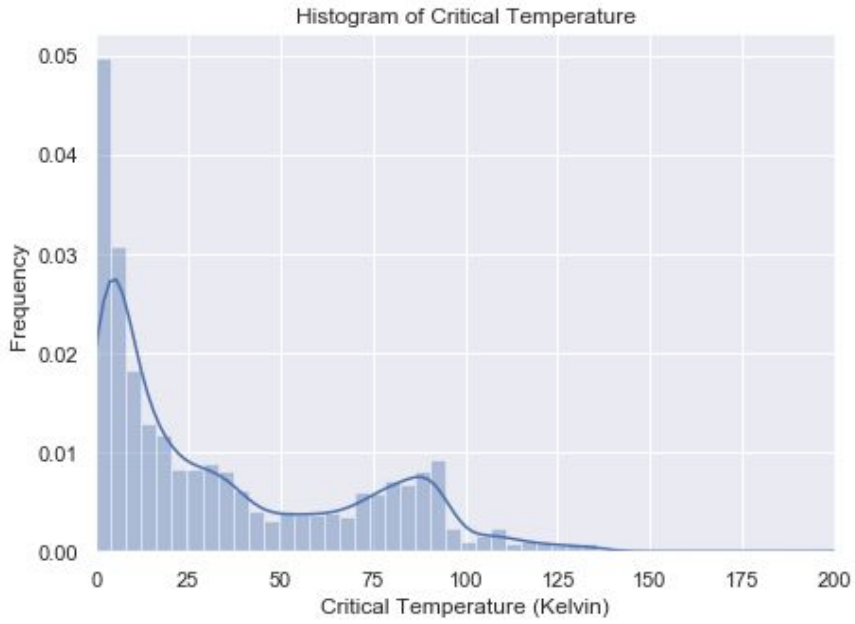

Figure 1: Histogram of critical temperature.

Figure 1 is a histogram of the critical temperatures. The critical temperatures are heavily right skewed and there is a small peak at about 90 K. Most of the materials have a critical temperature between 0 K and 10 K. Also, there is an outlier with a critical temperature of about 200 K.

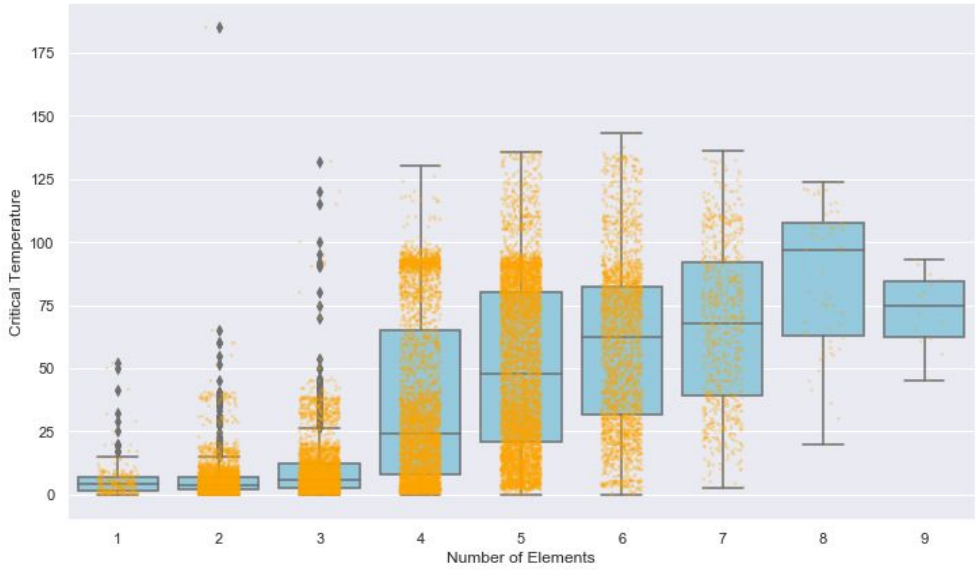

Figure 2: Boxplot of critical temperatures grouped by number of elements.

Figure 2 is a boxplot of critical temperature grouped by the number of elements. Most of the observations are concentrated between two elements and six elements. There are many outliers in the groups where the material has one, two, or three elements. As the number of elements increases, the material seems to have a higher critical temperature. Furthermore, there is a large increase in critical temperature between materials with three elements and materials with four elements.

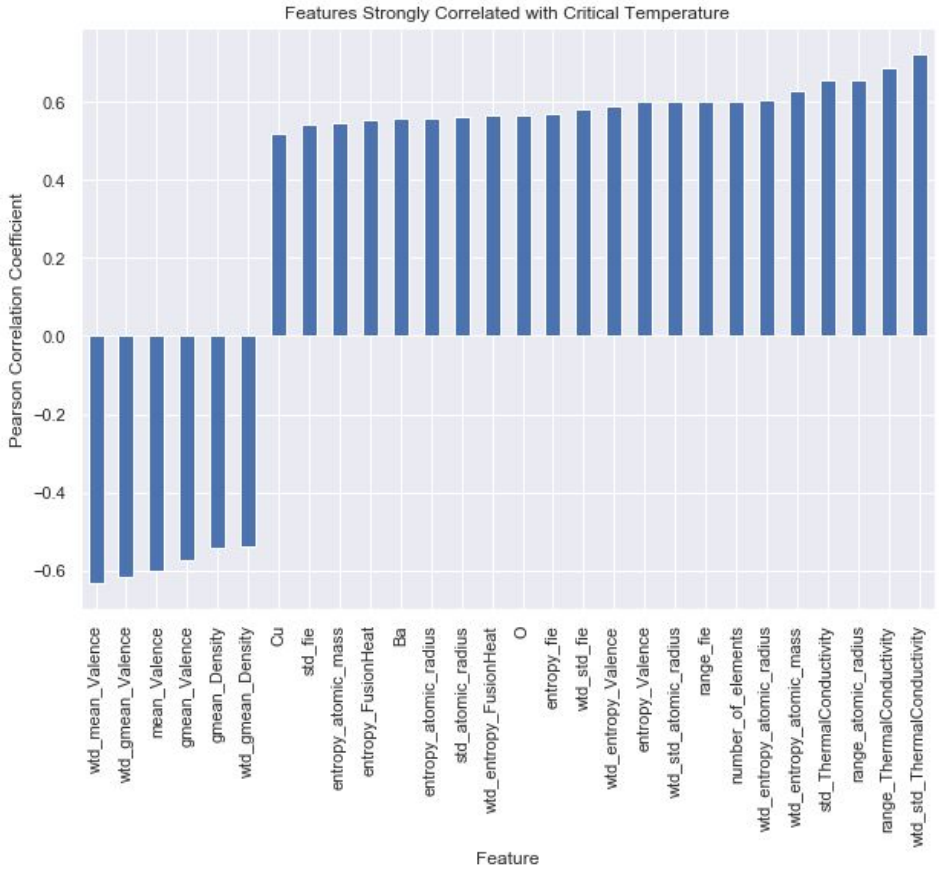

Figure 3: Features strongly correlated with critical temperature.

The Pearson correlation coefficient between critical temperature and each feature was calculated. There are 28 features where the absolute value of the Pearson correlation coefficient is greater than 0.5. These features are shown in Figure 3.

Features relating to thermal conductivity and atomic radius have a strong positive correlation with critical temperature. On the other hand, features relating to valence and density have a strong negative correlation with critical temperature. The largest positively correlated feature is the weighted standard deviation of thermal conductivity, with a correlation value of 0.72. The largest negatively correlated feature is the weighted mean of valence, with a correlation value of -0.63.

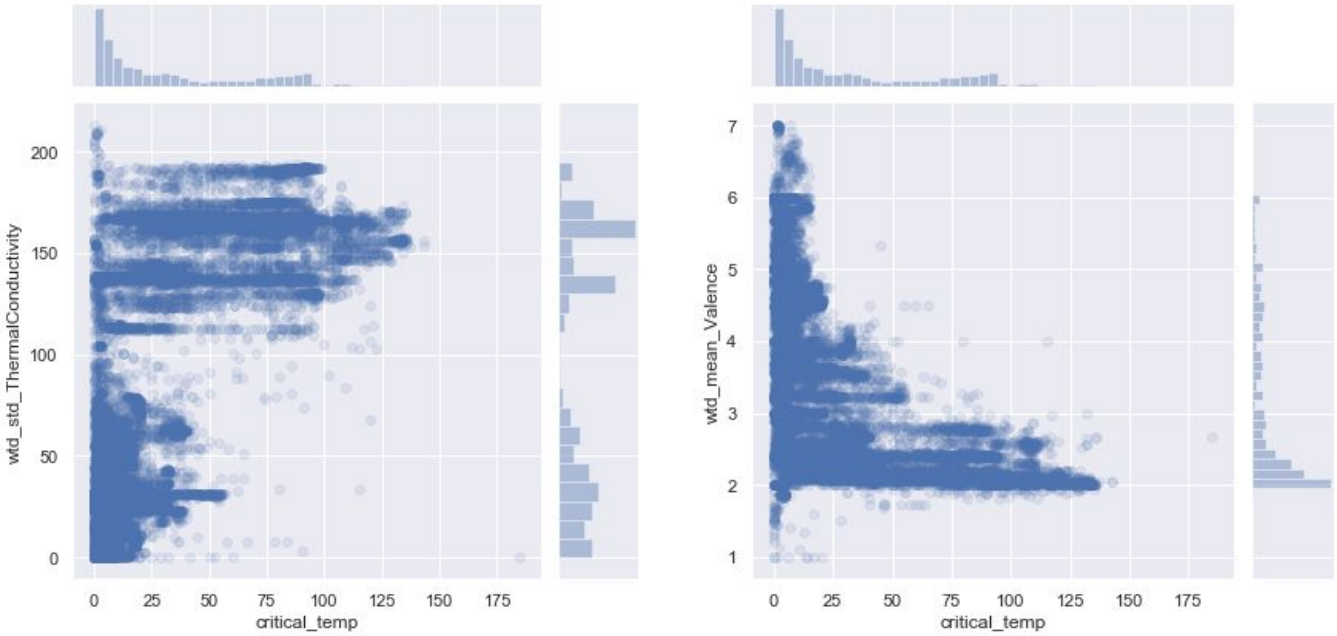

Figure 4: Scatter plots of features highly correlated with critical temperature.

Figure 4 includes scatter plots of the weighted standard deviation of thermal conductivity (left) and weighted mean of valence (right) with respect to critical temperature. The left plot shows that materials with higher critical temperature have higher values for weighted standard deviation of thermal conductivity. The right plot shows that materials with higher critical temperature have lower values for weighted mean valence.

III. Methods

A. Feature Selection

The original data set has 167 features, which is fairly large. Therefore, only a subset of the features will be used in order to reduce overfitting, reduce training time, and improve the accuracy of the model [3]. Lasso regression, an embedded method, was used to reduce the feature set. Lasso regression performs L1 regularization, meaning that it adds a penalty term equal to the summation of the absolute values of the coefficients [3]. The lasso regression model picked 14 features, and eliminated the other 153 features. Figure 5 shows the features lasso chose and their feature importances.

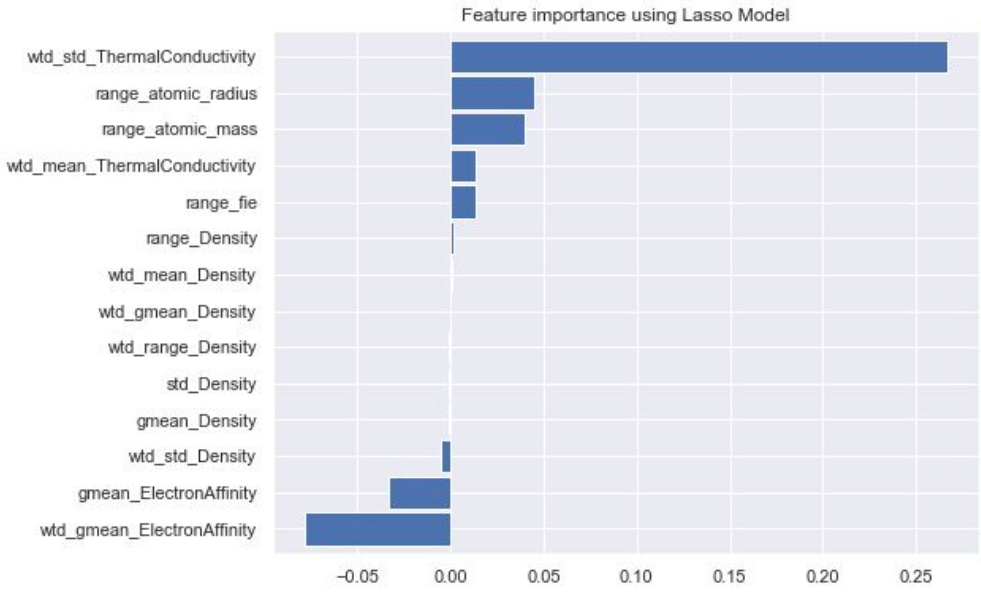

Figure 5: Feature importance from the lasso model.

B. Training and Testing Set

The reduced data set was standardized and split into 80% training and 20% testing.

C. Model Selection

To select a model, six different regression models were compared: linear regression, support vector regression, k-nearest neighbors regression, decision tree regression, random forest regression, and multi-layer perceptron regression. The purpose of this is to estimate the performance of each model (without tuning the hyperparameters). The best performing model will be selected and hyperparameters will be tuned on this model.

	train_r2	test_r2	test_rmse
Random Forest Regression	0.977155	0.918110	9.754758
K-nearest Neighbors Regression	0.960687	0.898280	10.871873
Decision Tree Regression	0.985734	0.856664	12.905593
Multi-layer Perceptron Regression	0.803285	0.796068	15.393723
Support Vector Regression	0.757310	0.749889	17.047750
Linear Regression	0.612229	0.612878	21.209240

Table 1: Comparison of model performance for six different regression models.

The metrics used to compare the models were the training R2, testing R2, and test root mean squared error (RMSE). These are shown in Table 1. The worst performing model was linear regression, with a test R2 of 0.61 and test RMSE of 21.21. Both random forest regression and k-nearest neighbors regression performed very well. However, the random forest regression model slightly outperformed the k-nearest neighbors regression. The random forest regression model had a test R2 of 0.92 and test RMSE of 9.75.

Random forest models are an ensemble learning method that uses multiple decision trees to make predictions [4]. Since the default model of random forest performed best on this data set, the hyperparameters will be tuned on this model.

D. *Hyperparameter Tuning*

Hyperparameter tuning is a process that searches the hyperparameter space for a set of hyperparameters that will optimize the model performance [5]. The parameter grid included different values for the following parameters: bootstrap (the method for sampling data points), max_depth (maximum depth of tree), max_features (maximum number of features considered for splitting a node), min_samples_leaf (minimum number of data points allowed in a leaf node), min_samples_split (minimum number of data points placed in a node before it is split), and n_estimators (number of trees in the forest) [6].

First, a random search of hyperparameters was performed to get an estimate of good parameters. This method randomly chooses combinations of parameters from a wide range of hyperparameter values [5]. The random search narrows down the range of optimal hyperparameters [5]. The function used to do this in Python was RandomizedSearchCV. Ten different combinations were tested using five fold cross validation, for a total of 50 fits. The default model of the random forest regressor had a test RMSE of 9.75. After random search training, the test RMSE decreased to 9.70, which is a 0.59% improvement.

Then, a grid search of parameters was performed to concentrate the search and obtain the optimal hyperparameters. The function used to do this in Python was GridSearchCV. Grid search tries every combination possible in the hyperparameter space [6]. The hyperparameter space in grid search was concentrated around the best parameters obtained from the random search. After tuning the hyperparameters with grid search, the test RMSE decreased to 9.55, or a 2.08% improvement when compared to the default model of the random forest regressor.

IV. Results

A. *Optimal Random Forest Regression Model*

The best parameters determined from the grid search were bootstrap = False, max_depth = 20, max_features = 'log2', min_samples_leaf = 2, min_samples_split = 5, and n_estimators = 200.

B. *Feature Importances*

Figure 6 shows the feature importances from the optimal random forest regression model. The most important features in the Random Forest model were the range of the atomic radius, weighted standard deviation of thermal conductivity, and the range of first ionization energy.

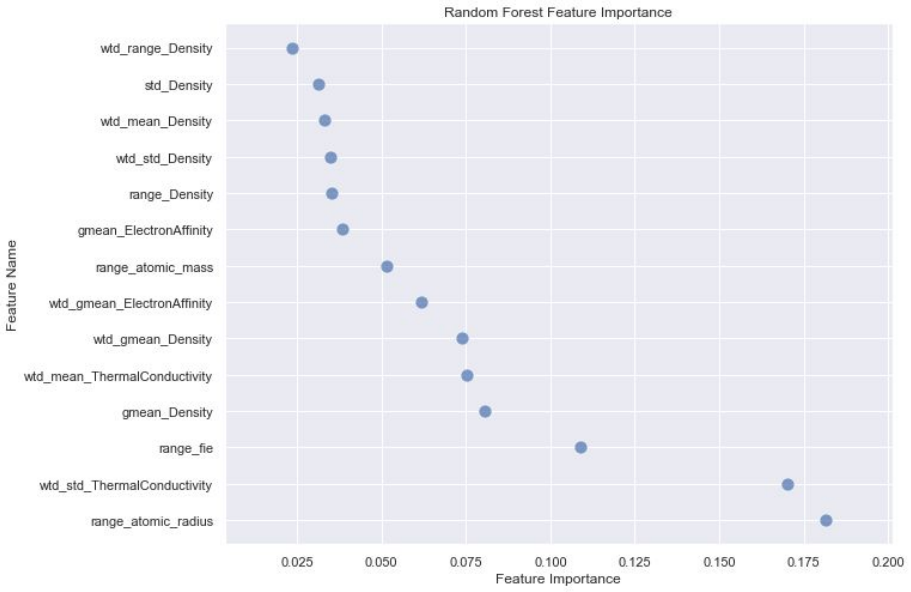

Figure 6: Feature importances obtained from random forest model after hyperparameter tuning.

V. Model Evaluation

A. *Cross Validation Scores*

The 10-fold cross validated R2 score had a mean of 0.75 and variance of 0.10. The 10-fold cross validated RMSE had a mean of 11.61 and variance of 5.09.

B. *Residual Plot*

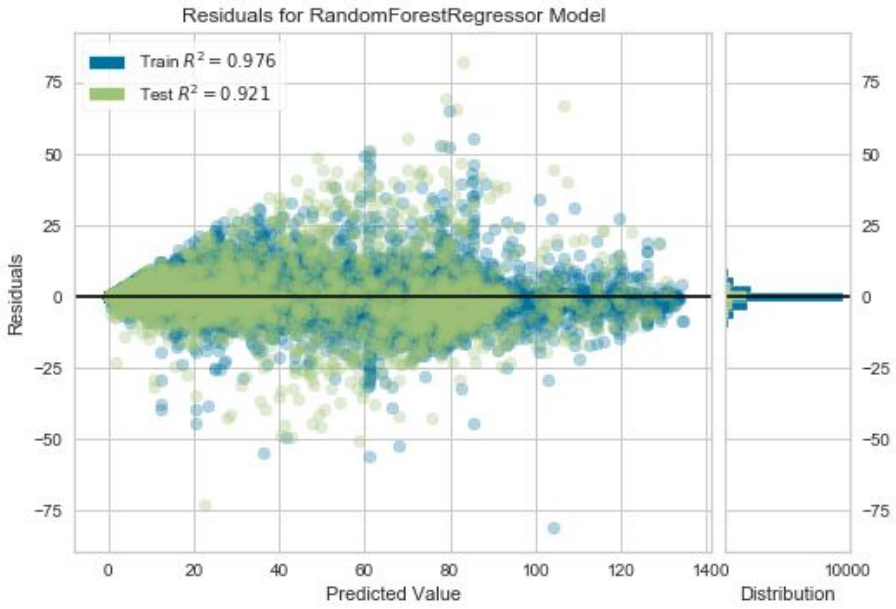

Figure 7: Residual plot for the random forest regression model.

Figure 7 shows the residual plot for the optimal random forest regression model. The residual is calculated by subtracting the predicted critical temperature from the actual critical temperature. The plot shows that the residuals are clustered around the middle and mostly centered around the zero line. There is no general clear pattern in the residual plot, but there are a couple outliers in the data.

C. *Plot of Actual vs. Predicted Values*

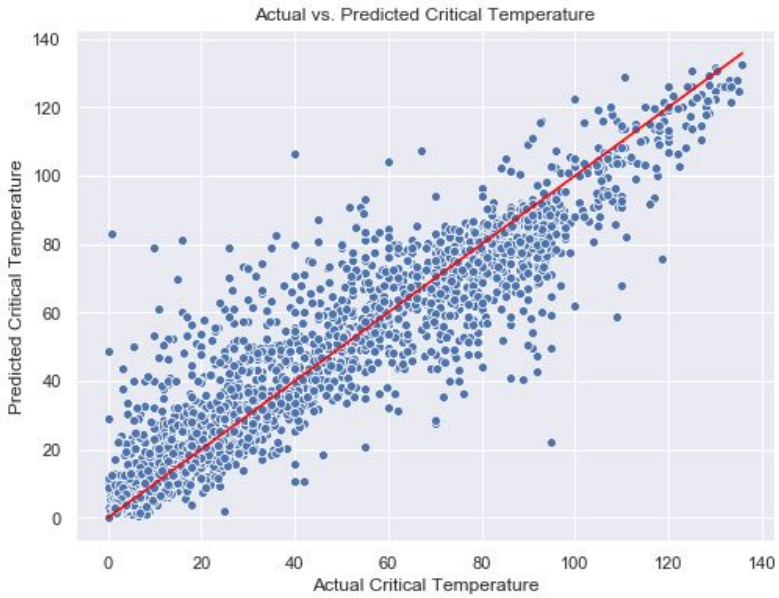

Figure 8: Plot of actual vs. predicted critical temperature.

Figure 8 shows a scatterplot of actual critical temperature against the predicted critical temperature. There is a strong correlation between the actual critical temperature and the predicted critical temperature. However, a couple points are quite far from the red identity line.

D. *Checking for Normality of Residuals*

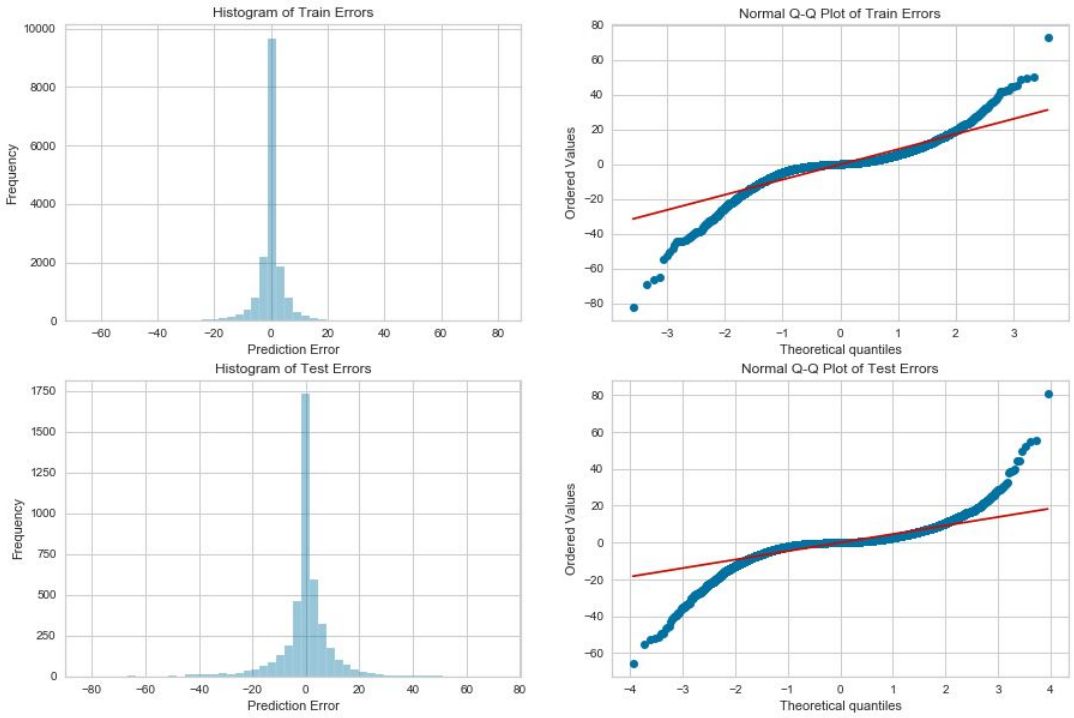

Figure 9: Histogram (left) and Q-Q plot (right) of train and test residuals.

Figure 9 shows histograms and Q-Q plots of the train and test residuals. Both histograms for the train and test errors are heavy tailed, meaning that they have many extreme positive and extreme negative values. The Q-Q plot shows that the relationship between the sample percentiles and theoretical percentiles is not linear, thus, the error terms are not normally distributed.

VI. Conclusion

Exploratory data analysis of the superconductivity data set revealed that critical temperature of a superconductor increases as the number of elements increases. Some important elements in determining the critical temperature of a material are copper and oxygen. In addition, fitting the data set with different regression models shows that random forest regression and k-nearest neighbors regression achieved the best results, while support vector regression and linear regression did not perform very well.

Based on the random forest regression model, the most important variables were related to atomic radius and thermal conductivity. Each of these variables had a feature with an importance score greater than 0.15. Many features extracted from the density attribute were also important in determining the critical temperature. Thus, these attributes may be significant when determining the critical temperature of a superconductor.

References

[1] S. Westerdale, "Superconducting Metals: Finding Critical Temperatures and Observing Phenomena," *MIT Department of Physics*, April 2010.

[2] "Superconductivity," *CERN European Organization for Nuclear Research*.

[3] V. Fonti, “Feature Selection using LASSO,” *VU Amsterdam Research Paper in Business Analytics*, March 2017.

[4] G. Groner, “A Random Forest Based Classifier for Error Prediction of Highly Individualized Products,” *Machine Learning for Cyber Physical Systems*, p 26-35, December 2018.

[5] J. Bergstra and Y. Bengio, “Random Search for Hyper-Parameter Optimization,” *Journal of Machine Learning Research*, vol. 13, December 2012.

[6] W. Koehrsen, “Hyperparameter Tuning the Random Forest in Python,” *Towards Data Science*, January 2018.