Interpreting and Predicting the Critical Temperatures of Superconductors.

Ryan Smith
University of California, Davis

***Abstract*- An exploration into the features of 21263 superconductors, and how specific attributes influence the critical temperature of the superconductor. Prediction will be performed to attempt to find the variables which affect the critical temperature the most.**

I. Introduction

In this report I will analyze two data sets: one titled 'train' which contains 81 pieces of information, with ten pertaining to each of 8 main characteristics for each of 21263 superconductors, along with the critical temperature as the 82nd piece of data. The other dataset, titled 'unique_m' contains the chemical formula for each of the superconductors broken up by the quantity of each element found in each superconductor. These data sets come from Kam Hamidieh at the University of Pennsylvania department of Statistics. Both of the data sets which will be analyzed in this report are publicly available from the University of California, Irvine Machine Learning Repository.

In this paper the focus will be applying numerous models to the data in order to predict the critical temperature of a superconductor as accurately as possible. To do this I will first perform basic data exploratory data analysis to see the relationships between the features. I will use principal component analysis to see how many dimensions the data can be reduced to without losing too much information, which will give me a rough idea of how strongly related the various information in the data set is. I will then use several statistical models and techniques for predicting the critical temperature of each superconductor. I will compare the performance of each of these predictive models in order to find the method which is able to predict the critical temperature of a superconductor based on the 81 attributes the best. Also, I will use the importance assigned to each variable by the best performing models to conclude which variables have the largest effect on the critical temperature of a superconductor.

A. Train dataset

The train data set that I will explore contains 82 pieces of information on 21263 superconductors. There are 8 main chemical properties of each compound: atomic mass, first ionization energy, atomic radius, density, electron affinity, fusion heat, thermal conductivity, and valence. For each of these 8 main characteristics there at 10 measurements: mean, weighted mean, gmean, weighted gmean, entropy, weighted entropy, range, weighted range, standard deviation, and weighted standard deviation. These make up 80 of the pieces of data for each superconductor. The remaining two pieces of data is the number of elements, and the critical temperature of each superconductor.

B. *Unique_m dataset*

The Unique_m data set that I will explore contains the unique chemical formula for each of the 21263 superconductors. For each superconductor, it specifies the quantity of each element, such as Helium or Hydrogen. In this paper I will focus on predicting the critical temperature based on the quantity of each of these elements found in the superconductor. To accomplish this, I will first perform basic exploratory data analysis to get an understanding of the correlations between the elements found in each superconductor. I will then use various prediction methods to find which minimizes the prediction error on unseen data.

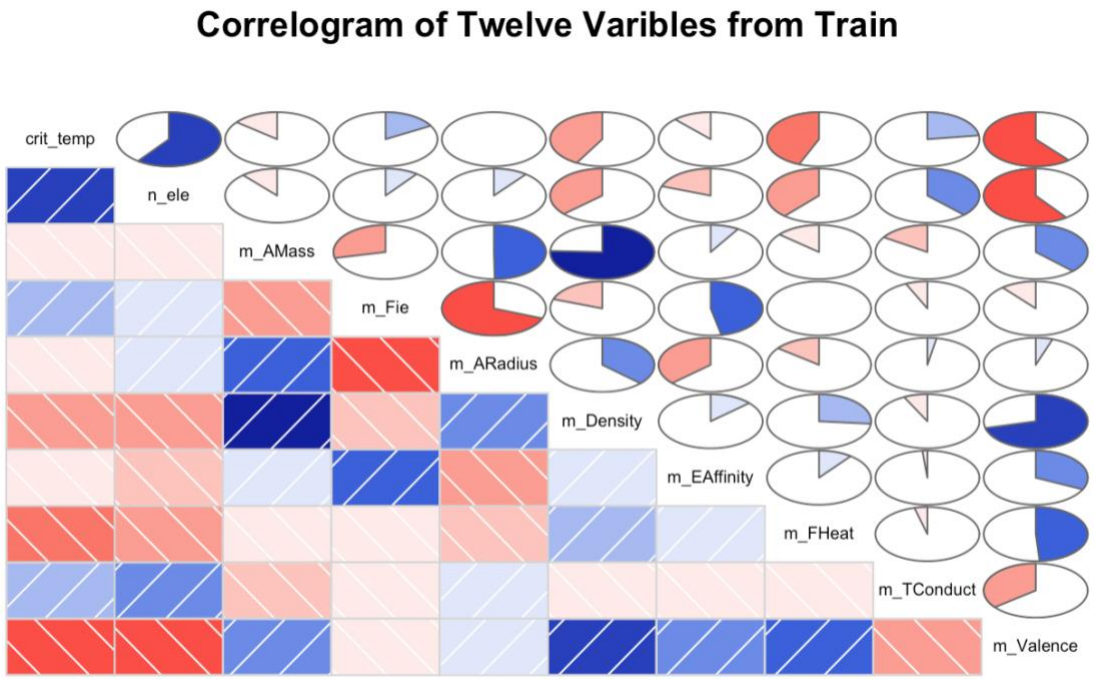


Figure 1. Correlogram of twelve variables from the Train dataset. Blue represents a positive correlation, while red represents a negative correlation. Darker is stronger correlation, whereas lighter is a weaker correlation.

II. Analyzing the Datasets

A. *Exploratory Data Analysis*

To begin the exploration of the datasets, I will first plot a correlogram of the train datasets that will show the correlation between the twelve main characteristics in the dataset. This is shown in Figure 1.

From fig 1, you can see that of all the means of each chemical characteristics of each superconductor, the variables that have the highest positive correlation with critical temperature of a superconductor are the number of elements, the mean fie, and the mean thermal conductivity. The variables with the highest negative correlation with critical temperature are the mean density, the mean fusion heat, and the mean valence.

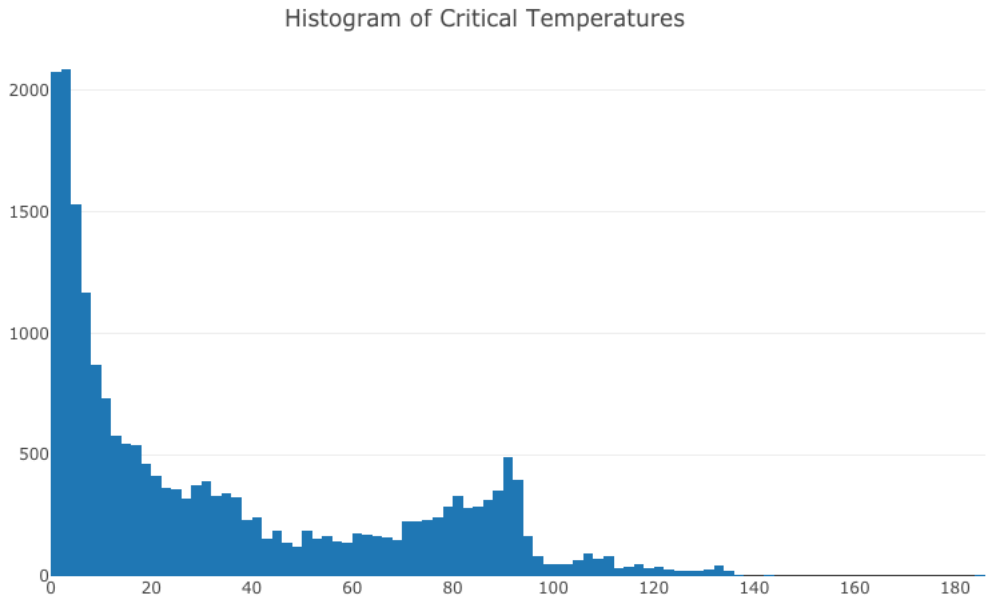


Figure 2. A histogram of the critical temperatures of the superconductors.

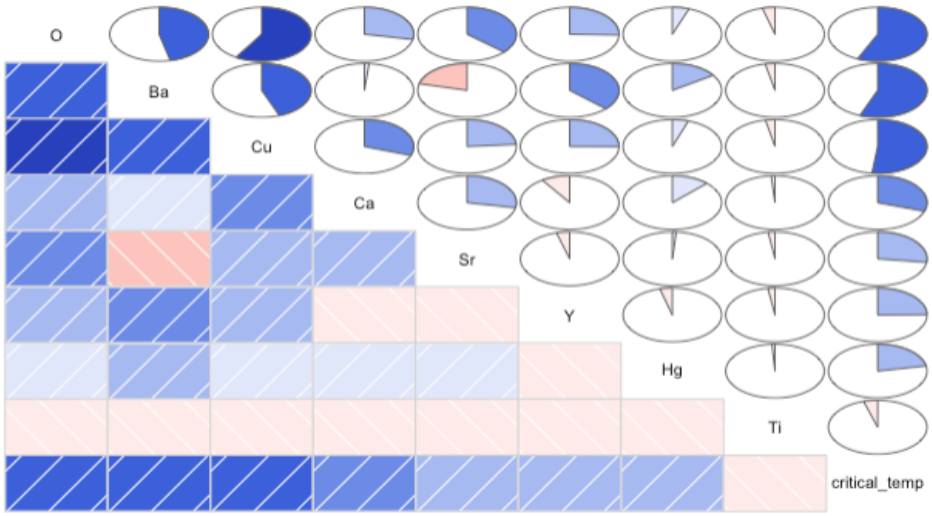
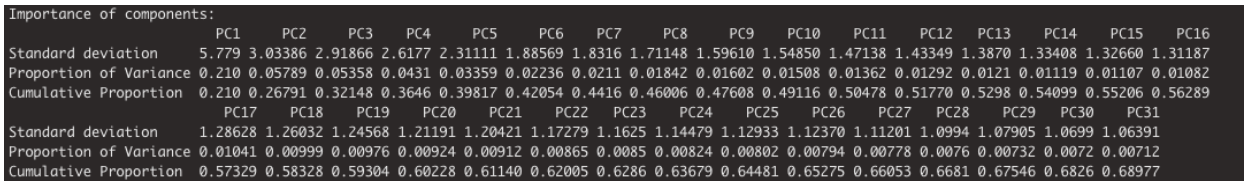


Figure 3. A correlogram of the unique_m dataset. Darker is a stronger absolute correlation and blue is a positive correlation while red is a negative correlation.

Fig 2 gives a good general understanding of the distribution of the critical temperatures. There are a significant amount of superconductors that have a critical temperature below 20, and then the frequency drops off until it hits a low point of 50. It then rises until it peaks at 90, and drops off at 100 and continues to drop off.

Fig 3 shows the highest correlation between the 12 variables in the unique_m with the largest absolute correlation. These variables are all positively correlated with the critical_temperatures, with none of the largest absolute correlations being negative.



```
Importance of components:
                          PC1     PC2     PC3     PC4     PC5     PC6     PC7     PC8     PC9    PC10    PC11    PC12    PC13    PC14    PC15    PC16
Standard deviation     5.779 3.03386 2.91866 2.6177 2.31111 1.88569 1.8316 1.71148 1.59610 1.54850 1.47138 1.43349 1.3870 1.33408 1.32660 1.31187
Proportion of Variance 0.210 0.05789 0.05358 0.0431 0.03359 0.02236 0.0211 0.01842 0.01602 0.01508 0.01362 0.01292 0.0121 0.01119 0.01107 0.01082
Cumulative Proportion  0.210 0.26791 0.32148 0.3646 0.39817 0.42054 0.4416 0.46006 0.47608 0.49116 0.50478 0.51770 0.5298 0.54099 0.55206 0.56289
                          PC17    PC18    PC19    PC20    PC21    PC22    PC23    PC24    PC25    PC26    PC27    PC28    PC29    PC30    PC31
Standard deviation     1.28628 1.26032 1.24568 1.21191 1.20421 1.17279 1.1625 1.14479 1.12933 1.12370 1.11201 1.0994 1.07905 1.0699 1.06391
Proportion of Variance 0.01041 0.00999 0.00976 0.00924 0.00912 0.00865 0.0085 0.00824 0.00802 0.00794 0.00778 0.0076 0.00732 0.0072 0.00712
Cumulative Proportion  0.57329 0.58328 0.59304 0.60228 0.61140 0.62005 0.6286 0.63679 0.64481 0.65275 0.66053 0.6681 0.67546 0.6826 0.68977
```

Figure 4. Principal component analysis of the combined train and unique_m dataset.

III. Principal Component Analysis

The next objective was to perform principal component analysis on the train dataset to get a rough idea of how correlated the data is, and to see how many dimensions are necessary to contain most of the information of the dataset. I combined both of the datasets and removed the variables from unique_m that have no occurrences to get a data frame with 157 variables on the 21263 superconductors. I then performed principal component analysis and was able to reduce the number of dimensions down to 33 while still maintaining 0.70363 of the original variance. You can see the results of the first 31 principal components in fig 4. This lets me know that many of the variables can be removed without losing too much of the original information found in the dataset.

IV. Classification

I will now perform classification on the combined data set in order to predict which class of critical temperature a superconductor will fall into based on its values for its variables. First, I will divide the critical temperature of each superconductor to one of four classes, which will be decided as follows: class 0 if the critical temperature is between 0 and 30, class 1 if the critical temperature is between 30 and 60, class 2 if the critical temperature is between 60 and 90, and finally class 3 if the critical temperature is above 90. To do this I will use a neural network, using the Keras neural network API, to classify the critical temperatures of unseen superconductors. Figure 4 shows a plot of the loss value, and the accuracy of the model improving as each of the 30 epochs is performed by the model to better fit the data. I used three dense layers in my neural network: the first had a dimensionality of 2056, the second had a dimensionality of 1024, and the third and final had a dimensionality of 4, to match the number of classes that are being predicted. For the first two layers I used the relu activation method, and for the final one I used the softmax activation. After running all 30 epochs, the accuracy of the classification was 65%, and you can see the improvement in accuracy across the epochs in fig 5.

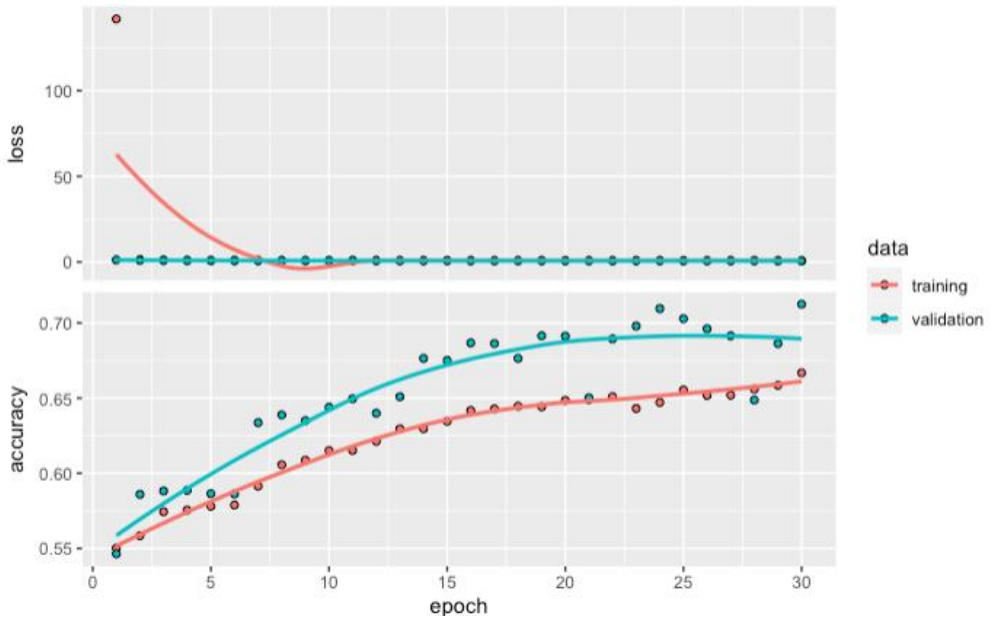


Figure 5. Graph of neural network accuracy and loss results across the 30 epochs.

V. Prediction of Critical Temperature

Now I will perform several different techniques and models of predicting the critical temperature of a superconductor. To do this I will start with just the train data set because it has much higher correlation with critical temperature than unique_m and the size is much more manageable with just one of the datasets. I will use several methods with just this dataset, and then I will incorporate the unique_m dataset which yields a dataframe with 168 variables about 21263 different superconductors. I will then clean the data using multiple different techniques in order to maximize the accuracy of my predictions. To determine the performance of each of my predictive methods, I will use a similar process for each technique. I will sample 70% of the data to be the training data, and the remaining 30% to be the testing data. I will then fit the training data to the model with critical temperature as the response variable and every other variable in the dataframe as an explanatory variable, and then predict the critical temperature of the testing data. I will then compute the RMSE (Root Mean Squared Error) of each of the methods by taking the predicted critical temperature and subtracting the true critical temperature, squaring it, then add up these values for all of the predictions, and then dividing by the number of predictions to find the average root squared error. For all of the models, I will repeat that process 100 times and take the mean RMSE across the 100 trials, but for my best model I will also verify the RMSE by using a k-fold cross validation technique, and conclude with the importance of each variable.

A. Prediction Using Train Dataset

To begin my prediction of critical temperature using only the train dataset, I will use a basic linear regression model on the data. The response variable will be critical temperature, and every other variable will simply be added

as explanatory variables. For this model, I use every available variable in the train dataset, and split the 21263 into a 70/30 training versus test data split. I train the model on the training data, and then used the R predict function to predict the critical temperature of the testing data's critical temperature. I repeated this process 100 times, and took the mean root mean squared error (RMSE) of the 100 trials. The result was an average RMSE of 17.62599 across the 100 runs.

The next method I used to predict the critical temperature of the test data was linear discriminant analysis. The data set was the entire train data set. The formula was again critical temperature as the response variable, and every other variable added as explanatory variables. The data was split into a 70/30 training versus test split and the model was trained on the training data and the test data critical temperature was predicted 100 times. The mean RMSE across the 100 runs was 17.39181.

The final method I used on the training data set was random forest regression. The train data set was too large to run on my computer, so I instead opted to just use the mean of each of the ten main characteristics, along with critical temperature and number of elements. I trained the model with the training data and predicted the testing data critical temperature again, and got an RMSE of 13.46 across 100 run times. The random forest regression was unstable in RStudio, and often caused the program to freeze before completing due to the large complexity of the dataset.

B. Prediction Using Both Training and Unique_m Cleaned Data Sets

After completing the prediction models with just the training data set, I combined the training and unique_m dataset to include all 183 observations available for each superconductor. I initially fitted a linear regression model to the training data and then predicted the test data, but got an RMSE of 49.823, which is significantly worse than just the training data.

I then decided to clean up the data to make it smaller, and only include variables that are necessary to prediction. The first thing I did to the dataset was remove He, Ne, Kr, Xe, Po, Rn, At, and Pm since they do not have any occurrences for any superconductor. I then found the correlation matrix for all of the remaining 156 variables, and chose to remove all of the variable which have less than a 0.2 absolute correlation with critical temperature as these do not have much influence on the critical temperature. After this step the data frame that remained had 74 variables. I again calculated the correlation of these variables, and found pairs that had a greater than 0.95 correlation with each other. For each of these pairs, I removed whichever variable had a smaller absolute correlation with critical temperature to try and combat the issue of multicollinearity which could disrupt my predictions due to the data having a large number of highly correlated variables. The result was a data frame which I will refer to as reduced dataset, which had 67 observations for each of the 21263 superconductors.

With this reduced dataset, I split the data in to a 70/30 training and test data split and again fit the training data to a linear model with critical temperature as the response variable and every other variable as explanatory variables. I then predicted the critical temperatures of the test data set, and then calculated the RMSE. I repeated that process 100 times and got a RMSE of 17.769, which is slightly worse than when I simply fit the original train data set to the same model.

Next I fit the reduced training dataset to a linear discriminant analysis model with critical temperature as the response variable and every other variable as explanatory variables. I predicted the critical temperature of the reduced test dataset 100 times and calculated the RMSE for each of those 100 runs. The mean RMSE was 17.9283, which is roughly 0.3 lower than the linear discriminant analysis on the original train dataset.

The final model I used was a random forest regression model, which I trained on the reduced training dataset with critical temperature as the response variable and every other variable as explanatory variables. I repeated the splitting of the data into training and test data with a 70/30 split 10 times, and found the mean RMSE to be 9.22, which is significantly lower than any of the other methods I previously used. This is also significantly lower than the RMSE of the random forest prediction on just the training data set. To verify this result, I used a k-split cross validation with 10 folds, which resulted in a RMSE of 9.95872. This was by far the most accurate prediction method out of the methods I used.

III. Conclusions

Analyzing both the train and unique_m data sets were not simple due to a few reasons. First, the size of each of the original data sets were very large, with train having 82 columns, and unique_m having 88 columns, both with 21263 rows(the number of superconductors that information was collected on). This made many of the standard prediction models not work with the default data sets because of the computation requirements for a dataset of this size. The data was also hard to work with because many of the variables were closely connected. This made having the issue of multicollinearity a large concern, as variables such as mean atomic mass and weighted mean atomic mass had high correlation which implies that as one goes up, the other will also, which can affect your predictions.

I initially attempted to ignore these concerns, and fit the train data set to a linear regression, linear discriminant analysis, and a random forest model, and used these to predict the critical temperature of superconductors. The resulting root mean square errors however were much larger than I hoped to obtain, so I focused on finding the best methods to combat these problems. To solve these problems, I first removed variables with an absolute correlation less than 0.2 to reduce the size of the data set by removing variables that do not affect critical temperature significantly. Next, I found pairs of data with correlation greater than 0.95, and removed the one with the lowest absolute correlation with critical temperature to remove the issue of multicollinearity without sacrificing too much accuracy on the predictions. This reduced the number of variables from 170 down to 67, which I then fit to a linear regression model, a linear discriminant analysis model, and a random forest model to predict the critical temperature of superconductors based on the explanatory variables. Before cleaning the data the best RMSE I got for my predictions was with the random forest model on the train dataset, which resulted in a RMSE of 13.46. After cleaning the data, the random forest regressor still gave me the best RMSE of 9.95872.

References

[1] Hamidieh, Kam, “A data-driven statistical model for predicting the critical temperature of a superconductor,” Computational Materials Science, 2018.
[2] X. Wang, F. Hsieh, ‘Coarse- and fine-scale geometric information content of Multiclass Classification and implied Data-driven Intelligence,’ University of California, Davis.
[3] Gasner, Joe, “Predicting the Temperature of a Superconductor”, Github, January 26, 2019.