Modelling Critical Temperature of Super Conducting Materials.

Names:

Dylan Scott

Jobin Joseph

Taylor Bonar

1 Introduction:

Our team was tasked with investigating critical temperature of a list of chemical compounds. The goal would be to build a model that could predict the critical temperature of any new material presented to the model given its features provided. This would be very helpful as chemical structures that are identified as having the right properties could be used in determining what could be used in producing products. The data was provided in the form of two sheets. ‘unique\_m’ was a one hot encoded sheet noting all the elements that could be found within the compound. ‘train’ listed all the other features for the compound. After some initial investigation we found it best to not use ‘unique\_m’ and solely focus on the features provided in ‘train’.

2 Method:

Upon investigating the data, we found that there were no null vales. In ‘train’ there was 21,263 number of values with 81 number of features. No categorical data found outside the ‘unique\_m’ document. As expected, we went ahead and isolated my target variable to ‘crit temp’ and removed it from data set. We checked the variables for skewness and found that standard scaling our features would be beneficial for our predictions as well as lowering multicollinearity which we will check for later. We then standard scaled our data by using sklearn StandardScaler which normalized our features. We then checked for multicollinearity since they could lead to bias in our models by having features which are too closely related. A graph of our feature correlations can be found in Fig 1. By iterating though multiple correlation scores we finally landed on .9 as the best option. Starting with 81 columns we found 38 features to be colinear and removed them from the dataset. Fields removed can be found in Fig 2.

Chart, treemap chart

Description automatically generated

**Figure 1.** Corrleation Heat Map

Text

Description automatically generated

**Figure 2. Features removed due multicollinearity**

## Model Building and Evaluation:

Before running any L1 or L2 models we looked to get a linear regression base line using mean squared error (MSE) for our result. This model did not include any cross validation and was simply used to compare our other models to. The simple linear regression model resulted in a MSE of 357.67.

For our next models of L1 and L2 we used cross validation (CV) of 10 splits, and we set the random state to ensure consistency in our comparison of multiple runs.

Alpha is a very important metric in L1 and L2 as picking the right alpha will help improve the model while also preventing many outcomes that would lead us to the wrong conclusion. For L1 a smaller alpha is expected as it suppresses the features that have a low coefficient while boosting the features that have a high coefficient. This is their absolute values that we are talking about because highly negative coefficients are also considered “high impact”.

For L1 we ran multiple iterations changing the alpha as part of our hyper-tuning our parameters. Like mentioned above alpha is an important parameter to set properly. Like we expected as alpha got closer to 0 our MSE also lowered. We found that the appropriate alpha was .01 by using the elbow method with our cross-validation score (CV Score) to see where the best improvement drops off our graph. Overall, we achieved a MSE of 362.95 which does not mean it preformed worse than our base linear regression model. Due to issues with overfitting a lower MSE does not equal a better model.

For L2 we expect alpha to act the opposite and for our model to improve as alpha increases. Once again, we did 10 iterations through different alphas increasing alpha by a factor of 10 each time. Using the elbow method and looking at the CV Score we found that an alpha of 100 was best. Our L2 model yielded an MSE of 357.557.

# 3 Results:

For our results both L1 and L2 had a lower MSE when compared to simple liner regression. This does not mean the model did worse and could indicate some overfitting with a linear model. The MSE for L1 was 362.95 where as L2 had a sightly lower MSE of 357.557. The alpha levels for both models were selected by looking at the best CV Score for each level of alpha. For L1 we can see the prediction line in Fig 3 where it shows a slight positive linear regression. There appears to be a few outliers in the model that were not removed due to a worse performing model. In Fig 4 we can see how our L2 model performed slightly better but still with a similar positive linear slope.   
When it comes to feature importance, we found there were many similarities between the two models. The top 20 features for both models can be seen in Fig 5. Thermal Conductivity was seen as the most important feature for both models and Electron Affinity was also in the top 3 for both models. Thermal Conductivity was seen as a strong positive correlation whereas Electron Affinity was seen as a strong negative correlation. The top 5 differs where in L1 atomic mass is seen as having a strong positive correlation and is seen as the 4th most important feature. Where in L2 atomic mass is seen as far less important than other features presented in the top 20.

Chart, scatter chart

Description automatically generated

**Figure 3. L1 Predictions**

Chart, scatter chart

Description automatically generated

**Figure 4. L2 Predictions**

Graphical user interface

Description automatically generated with medium confidence

**Figure 5. L1 and L2 Feature Importance**

# 4 Conclusions:

In conclusion, when comparing the MSE of our L1 and L2 models we saw that L2 performed slightly better. This is to be expected as L1 is most notably used for feature importance since as alpha decreases the model lowers the importance of low coefficient features. For modeling a linear regression, we would suggest using L2. We believe the L2 will produce the best results for predicting the critical temperatures of superconducting materials.

Looking at collinearity we found many of our features were closely related and would suggest looking at those features before using all of them to model. Standard scaling the data is important to lower collinearity as well as standardize our data and ensure normal distributions. Overall L2 runs quick and efficiently on this dataset and it our choice for linear regression.

# A Code

Submitted in Jupyter Notebook