STA 160 - Data Science Practices
Final Report
Superconductivity Data set

Author: Jordan Brennan

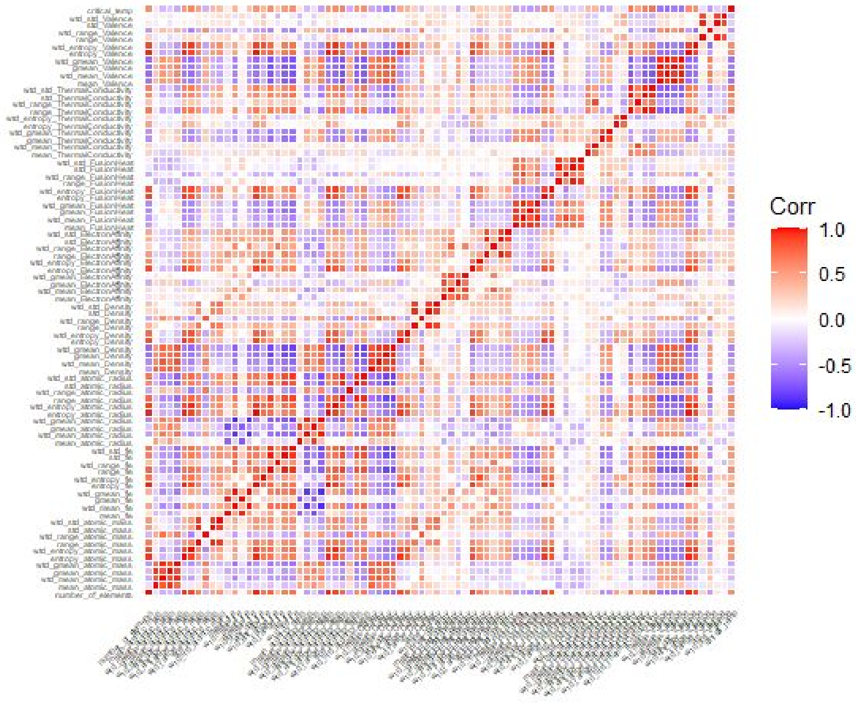
I. INTRODUCTION

The goal of this report is to explore a multivariate data set from the UCI Machine Learning Repository, Superconductivity. Points of interest include but are not limited to, comparing multiple labels with respect to one single feature, measuring associative relations between a numerical response variable and multiple covariate features, and working with linear regression to best fit the data. We will begin with analysis of the Superconductivity data, containing all continuous variables.

II. SUPERCONDUCTIVITY

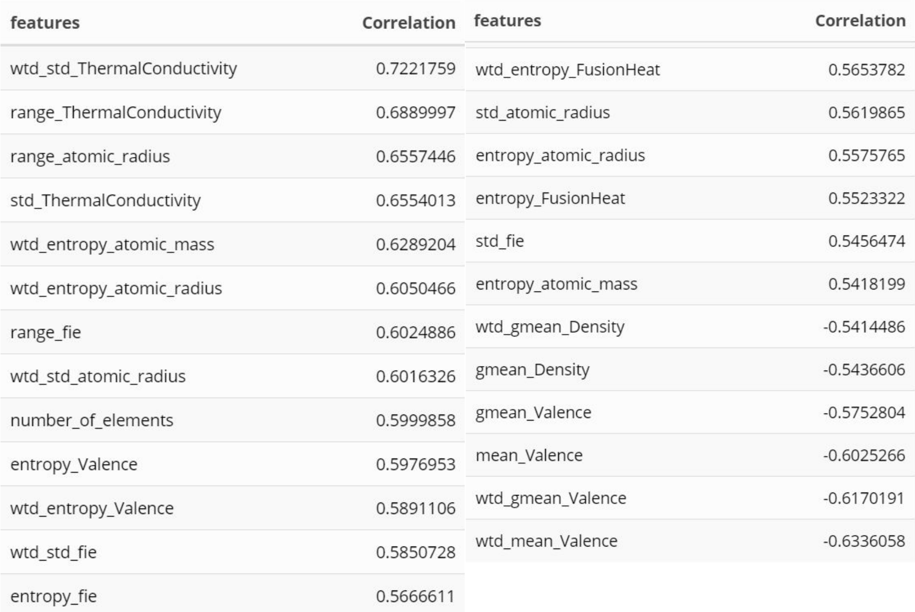
We will be taking a closer look at the Superconductivity data, where the target variable is the critical temperature along with 81 unique predictor variables. The data set can be used for the tasks of finding associations of predictors with critical temperature as well as fitting the data with a regression model. Given that there are 81 predictor variables, we must start with preprocessing the data and filtering out predictors that are more or less insignificant to the target variable critical temperature. In order to get an idea of what variables are related to each other negatively or positively, let's first take a glance at the correlation matrix, (**Fig. 1**) on the next page. Note the legend on the side to tell the difference of a positive(red)/negative(blue) relationship and the intensity of that relationship depending on the shade of the color, with darker colors representing stronger relationships.

Figure 1: Correlation Matrix Superconductivity Data - All Features



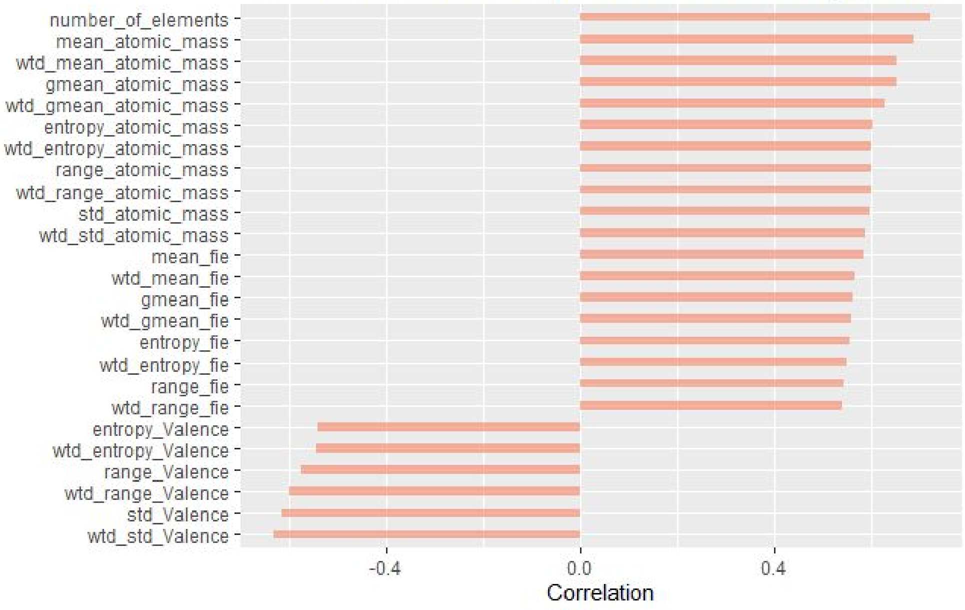
The correlation matrix above gives us a rough understanding of the relationships between all of the variables, and avert your attention to the very top row, critical temperature. We can see a mix of positively and negatively correlated variables along with some lightly shaded boxes that represent little correlation. This matrix uses the pearson method for finding the correlation between each variable, and next will construct a table that consists of the strongest, positive correlation values in regards to critical temperature. On the following page in (**Table 1.**) and (**Fig.2**), we can see the top 25 features that have the greatest Pearson correlation coefficients along with their respective R values. These 25 features were selected upon the criteria of either having a correlation greater than 0.5 or less than -.05. We will then take the data that consists of just these 25 of 81 original predictors and split them into training and testing sets for proceeding with OLS and Lasso regression.

Table 1: Correlation Coefficients with Respect to Critical Temp



features	Correlation
wtd_std_ThermalConductivity	0.7221759
range_ThermalConductivity	0.6889997
range_atomic_radius	0.6557446
std_ThermalConductivity	0.6554013
wtd_entropy_atomic_mass	0.6289204
wtd_entropy_atomic_radius	0.6050466
range_fie	0.6024886
wtd_std_atomic_radius	0.6016326
number_of_elements	0.5999858
entropy_Valence	0.5976953
wtd_entropy_Valence	0.5891106
wtd_std_fie	0.5850728
entropy_fie	0.5666611
wtd_entropy_FusionHeat	0.5653782
std_atomic_radius	0.5619865
entropy_atomic_radius	0.5575765
entropy_FusionHeat	0.5523322
std_fie	0.5456474
entropy_atomic_mass	0.5418199
wtd_gmean_Density	-0.5414486
gmean_Density	-0.5436606
gmean_Valence	-0.5752804
mean_Valence	-0.6025266
wtd_gmean_Valence	-0.6170191
wtd_mean_Valence	-0.6336058

Figure 1: Pearson Correlation Coefficient with Critical Temperature



For linear regression, we assume that the relationship between the predictor and response variable is linear. In order to prepare the data for regression, the numeric predictor variables must be scaled to ensure they won't adversely influence the regression model. After some data manipulation, I noticed that transforming the target variable, critical temperature, by `ln(1 +x)` yielded a better R-squared when performing regression. According to the QQ-plot in (**Fig. 3**), the data set seems to be normal with a slight negative skew with extreme values suggested by a slightly heavy tail, and (**Fig.2**) ensures the homogeneity of variances is checked.

Figure 2: Superconductivity Residuals Plot

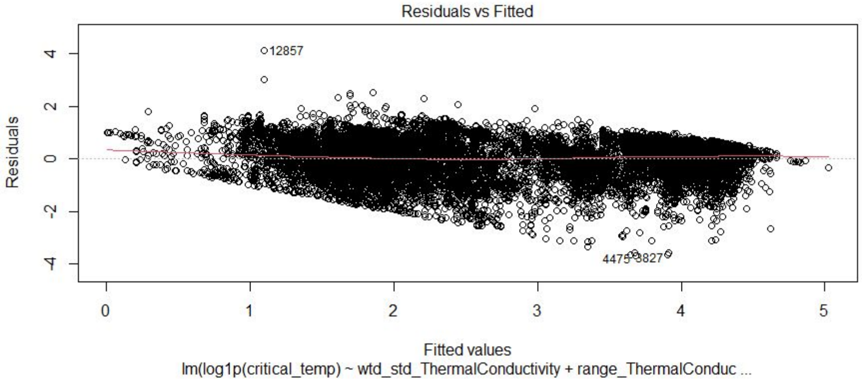
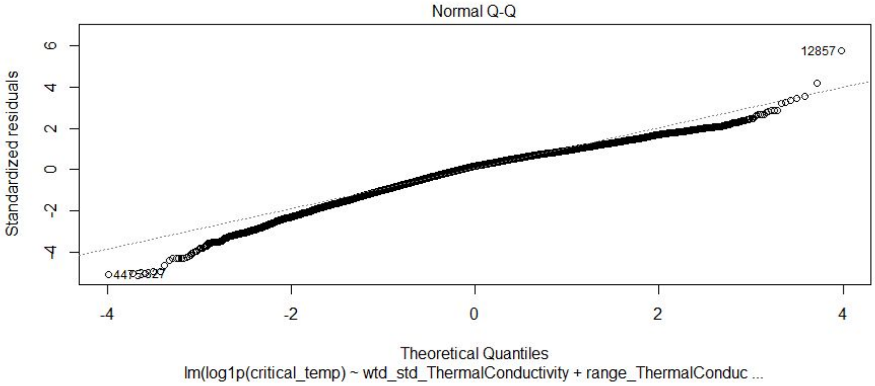


Figure 3: Superconductivity QQ-Plot



The linear regression model with the top 25 predictors yielded an R-squared value of about 69%, and when using this model for predictions on the training data, the RMSE is 46.12. Root Mean Square Error (RMSE) is a way to measure the error of a model in predicting numerical data. Then when using the model on the test data set, it had the same R-squared value of 69% and the RMSE is 45.48, which are close to the training indicating good performance. When comparing the linear model I developed using only ~30% of the predictors with a model using all predictors and no transformation of the target variable, the R-squared is better at 74%, but has a higher RMSE at 64.18 on the training and 63.62 on the testing.

The R-squared value for the linear model is decent at 69%, however let's try using regularization techniques to see if we can develop a better model for prediction. I decided to choose LASSO as it is a type of linear regression that uses shrinkage, where data values are shrunk towards a central point, and in this case, the mean. LASSO uses the L1-norm, and is well-suited for models showing high levels of multicollinearity, which should work well on this dataset. The cost function for LASSO regression uses the sum of absolute values of the coefficients, and can be written as (**1.1**).

$$\sum_{i=1}^{M} (y_i - \hat{y_i})^2 = \sum_{i=1}^{M} \left(y_i - \sum_{j=0}^{p} w_j \times x_{ij} \right)^2 + \lambda \sum_{j=0}^{p} |w_j| \tag{1.1}$$

Cost function for Lasso regression

"The term lambda regularizes the coefficients such that if the coefficients take large values the optimization function is penalized" (Bhattacharyya, 2020). This aids in reducing the multicollinearity between the predictors and complexity of the model. Using LASSO, the R-squared value for the training set is 74% with an RMSE of 17.49, and for the test set, 72%

R-squared value and 17.84 RMSE. These values are better than in the original OLS. The next regularization technique is that of Ridge regression, that is similar to LASSO in the fact that hyperparameter lambda is used to tune the model. The difference between the two, is that Ridge uses the L2 norm instead of the L1, which uses the sum of squares of the coefficients as shown in **(1.2)**

$$\sum_{i=1}^{M} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{M} \left(y_i - \sum_{j=0}^{p} w_j \times x_{ij} \right)^2 + \lambda \sum_{j=0}^{p} w_j^2 \qquad \textbf{(1.2)}$$

Cost function for ridge regression

Using Ridge Regression, we get a similar result to using LASSO because they are very similar procedures. On the training data set, the R-squared value is 74% and a lower RMSE than LASSO's at 14.64. However, both models performed very similarly on the testing set; Ridge yielded an R-squared value of 72% and 17.85 RMSE.

III. CONCLUSION

In analyzing the Superconductivity dataset, we found multicollinearity between the 81 predictors and narrowed it down to the top 25 with the highest correlation in regards to the target variable, critical temperature. Utilizing OLS with the selected predictors yielded a decent model to fit the data, but not fantastic by any means. Both of the regularization techniques, LASSO and Ridge regression, outperform linear regression because we are able to tune the hyperparameter, lambda, to better fit the model. LASSO and Ridge regression both lowered the RMSE tremendously from the linear model, and a higher R-squared. Further analysis using Decision Trees, Random Forest, and other methods for fitting a model for prediction could be used for a better analysis and prediction.

IV. REFERENCES

[1] Bhattacharyya, Saptashwa. “Ridge and Lasso Regression: L1 and L2 Regularization.” *Medium*, Towards Data Science, 4 May 2020, towardsdatascience.com/ridge-and-lasso-regression-a-complete-guide-with-python-scikit-learn-e20e34bcbf0b.