Case Study 3: Email Spam

Matt Farrow

May 11, 2022

1 Introduction

The data for this analysis comes from UCI’s Machine Learning Repository and examines the relationship between superconductive materials and their respective critical temperatures. Superconductive materials have the unique ability of being able to have their electrical resistance disappear as their temperatures drops below a specific critical temperature.

The ovjective of this case study is to use linear regression with both L1 and L2 regularization to exame the ability to predict a superconductor’s critical temperature. In addition, this analysis will attempt to determine which feature(s) lend the most importance to the prediction.

Unrelated to this analysis, my grandfather was an engineer on the Superconducting Super Collider here in Texas before it was shut down in 1993.

2 Methods

## 2.1 Data Examination

The initial data set is comprised of two separate files: train.csv and unique\_m.csv. As a result of the files being able to matched row-by-row, they were combined into a single data set for the purposes of this analysis. The response variable critical\_temp existed in both data sets; the variable was dropped from one of the data sets prior to joining. The new data set contains 21,263 observations and 168 variables including our response.

In understanding the data, there were no missing values and the data appeared to be in a proper format for proceeding with analysis. The exception to that was the material variable which contained 15,542 unique strings of data. This amount of unique data would not prove useful to a linear regression analysis, so material was dropped from the data set.

An examination of the response variable critical\_temp as a histogram shows a right-skewed distribution (Figure 1). Applying a log-transformation (Figure 2) did not serve to create a more normal distribution, instead the data is now left-skewed. Therefore, the analysis proceeded with the original data.

Chart, histogram

Description automatically generated

Figure : Histogram of Critical Temperature

Chart, histogram

Description automatically generated

Figure : Histogram of Critical Temperature (Log-Transformed)

## 2.2 Model Preparation & Execution

The response variable, critical\_temp, was separated from the rest of the data and then both the X and y data sets were split into test and train data sets using a 75%/25% split. Sklearn’s pipeline feature was used to collect the scaler (RobustScaler) and model (Lasso & Ridge) to be used in each model.

A sequence of potential alpha values was defined and then those pieces were wrapped into a GridSearchCV process where 10-fold cross validation was performed to determine the most appropriate parameters for the model. After fitting both models, the resulting neg\_mean\_absolute\_error and selected alpha value were examined.

|  |  |
| --- | --- |
| Best Score | -12.676 |
| Best Alpha | 0.001 |

Table : L1 (LASSO) Model Training Results

|  |  |
| --- | --- |
| Best Score | -12.636 |
| Best Alpha | 0.3 |

Table 2: L2 (Ridge) Model Training Results

3 Results

## 3.1 Model Results

Once trained, the fitted models were used to make predictions on the testing data that had been held out. For each model, an value along with the mean absolute error (MAE) were reported for performance comparision.

|  |  |
| --- | --- |
|  | 0.475 |
| Mean Absolute Error | 12.927 |

Table 3: L1 (LASSO) Model Testing Results

|  |  |
| --- | --- |
|  | 0.474 |
| Mean Absolute Error | 12.912 |

Table 4: L2 (Ridge) Model Testing Results

Interestingly, both models returned almost identical and MAE values. It is unclear whether an error was made during the modeling, or if the two models do indeed perform to such a close similarity.

## 3.2 Coefficient Weights

After completing the linear regression with both L1 (LASSO) and L2 (Ridge) regularlizations, the coefficients were analyzed to determine which coefficients most contributed to each model. The top five for each model are shown below.

|  |  |
| --- | --- |
| Variable | Coefficient |
| range\_ThermalConductivity | 30.494 |
| std\_ThermalConductivity | 29.282 |
| wtd\_entropy\_Valence | 29.185 |
| entropy\_fie | 27.979 |
| range\_fie | 25.123 |

Table 5: L1 (LASSO) Model Top 5 Coefficients

|  |  |
| --- | --- |
| Variable | Coefficient |
| wtd\_gmean\_atomic\_radius | 82.504 |
| wtd\_mean\_atomic\_radius | 76.231 |
| wtd\_mean\_atomic\_mass | 40.307 |
| wtd\_entropy\_Valence | 36.01 |
| std\_ThermalConductivity | 33.523 |

Table 6: L2 (Ridge) Model Top 5 Coefficients

4 Conclusion

In examining the top five coefficients of each model, additional work may have been needed to avoid potential multicollinearity as both models contained high performing coefficients that appear as though they may be correlated. It is also interesting to note that the coefficients of the L1 (LASSO) model have a much narrower distribution; within the L2 (Ridge) model, the top two performing coefficients were significantly separated from the next values.

# Appendix

## Sources

* [Lasso Regression with Python](https://www.kirenz.com/post/2019-08-12-python-lasso-regression-auto/#model-evaluation)
* [How to Use Sklearn Pipelines For Ridiculously Neat Code](https://towardsdatascience.com/how-to-use-sklearn-pipelines-for-ridiculously-neat-code-a61ab66ca90d)
* [Pre-Process Data with Pipeline to Prevent Data Leakage during Cross-Validation](https://towardsdatascience.com/pre-process-data-with-pipeline-to-prevent-data-leakage-during-cross-validation-e3442cca7fdc)
* [sklearn.linear\_model.LinearRegression](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html)
* [How to Develop LASSO Regression Models in Python](https://machinelearningmastery.com/lasso-regression-with-python/)
* [How to create a linear regression model using Scikit-Learn](https://practicaldatascience.co.uk/machine-learning/how-to-create-a-linear-regression-model-using-scikit-learn)
* [Linear Regression in Python](https://realpython.com/linear-regression-in-python/)

## Code

Code begins on the following page.