A FINE-GRAINED ANALYSIS OF XGBOOST PREDICTING T_c

DANIEL BRISENO

1. Introduction

This project is a review of the ability of the XGBoost regression tree algorithm's ability to predict the critical temperature (T_c) of superconductors, as proposed by Dr. Hamdieh in [1].

In [1], Dr. Hamidieh reports that a properly optimized XGBoost model can give an out-of-sample residual mean squared error (RMSE) of approximately 9.5° K. Here, I will present a finer grained analysis of the performance of XGBoost. More specifically, I will attempt to answer the following questions:

(1) How well does XGBoost preform at predicting T_c of Iron, Cuprate, Mercury and MgB_2 based super conductors? Does this performance improve or worsen when trained only on Iron, Cuprate, or MgB_2 based superconductors?

(2) If we divide the testing data by quartiles of T_c, how well does XGBoost preform at predicting T_c of compounds in each quartile? Is XGBoost reliable when predicting the T_c of a compound with a high T_c?

(3) If we divide the predictions of XGBoost by quartiles of predicted T_c, how accurate and precise is XGBoost's prediction when the prediction falls in a given quartile? Is the predicted value of T_c reliable when this predicted value is high?

2. The Dataset

The data used to train and test the XGBoost model both in [1] and in this report is taken from the Superconductivity Dataset found in the UCI Machine Learning Repository[2]. It is a collection of 21263 superconductors with 82 features defined for each superconductor. The 82nd feature is T_c and will serves as the target label. The features of the superconductors are generated from atomic properties of the elements in the superconductor. These properties are summarized in Table 1 (this table can also be found in [1]).

For a given superconductor in the data, the atomic properties of the elements which make up that superconductor are combined according to the summary statistics listed below:

(1) Mean

(2) Weighted mean

(3) Geometric mean

(4) Weighted geometric mean

(5) Entropy

(6) Weighted entropy

(7) Range

(8) Weighted range

(9) Standard deviation

(10) Weighted standard deviation

A more in-depth explanation of these summary statistics and their associated formulas can be found in Table 2 of [1].

Variable	Units	Description
Atomic Mass	Atomic mass units (AMU)	Total proton and neutron rest masses
First Ionization Energy	Kilo-Joules per mole (kJ/mol)	Energy required to remove a valence electron
Atomic Radius	Picometer (pm)	Calculated atomic radius
Density	Kilograms per meters cubed (kg/m3)	Density at standard temperature and pressure
Electron Affinity	Kilo-Joules per mole (kJ/mol)	Energy required to add an electron to a neutral atom
Fusion Heat	Kilo-Joules per mole (kJ/mol)	Energy to change from solid to liquid without temperature change
Thermal Conductivity	Watts per meter-Kelvin (W/(m K))	Thermal conductivity coefficient κ
Valence	No units	Typical number of chemical bonds formed by the element

TABLE 1. Elemental properties used to define features used by XGboost to predict T_c.

3. Methods

In this section I will describe the data collection process by first specifying the error statistics collected for predicted T_c values, then describing how these error statistics were collected for any relevant subset of the training and testing data, then finally describing the subsets themselves.

3.1. **Error Statistics.** Let P be the predicted value of T_c and A be the actual value. Then I define the **raw error** err as $err := P - A$. Note that a negative raw error indicates an under-prediction, and a positive raw error indicates an over-prediction.

This investigation consisted of collecting summary raw error statistics for T_c predictions by XGBoost on relevant subsets of the testing and training data. The error statistics collected are summarized in Table 2.

Statistic	Description
RMSE	The residual mean squared error of predictions
ave_err	Raw average of error
std_err	Standard deviation of raw error values
under_cnt	Number of under-predictions
over_cnt	Number of over-predictions
correct_cnt	Number of exactly correct predictions. Expected to be 0
ave_under	Average value of under-prediction raw error
ave_over	Average value of over-prediction raw error
std_under	Standard deviation of under-prediction raw error
std_over	Standard deviation of over-prediction raw error

TABLE 2. Summary error statistics collected for predicted T_c values.

3.2. **Subsets Studied.** Previously, I introduced three questions concerning the performance of XGBoost across three categories: superconductors containing certain elements, superconductors in different true T_c quartiles, and superconductors in different predicted T_c quartiles.

To this end, there are three classes of subsets under consideration: the element subsets, the true T_c quartiles, and the predicted T_c quartiles.

3.2.1. *The Element Subsets.* There were four subsets of the data which I call the element subsets. These subsets are: superconductors containing Fe (Iron), containing Cu (Copper), containing MgB_2, and containing Hg (Mercury).

The motivation for the Fe, Cu, and MgB_2 subsets is the body of previous work which applied machine learning to predicting T_c for superconductors containing these compounds. Predicting T_c for superconductors containing these compounds has received priority in the literature due to the practical applications of such superconductors. Thus, it would be useful to characterize how well XGBoost performs on these superconductors.

The motivation for the Hg subset comes from the distribution of T_c values for all superconductors in the dataset, summarized in Figure 1. This distribution is extremely skewed towards low values, and is not particularly clustered around a mean. Thus, it

seems possible that the model would be biased towards low T_c values. Superconductors containing Hg offer a "worst case" analysis. Not only do these superconductors have the highest mean T_c of any other elemental class in the dataset, they also have the fourth highest variance [1].

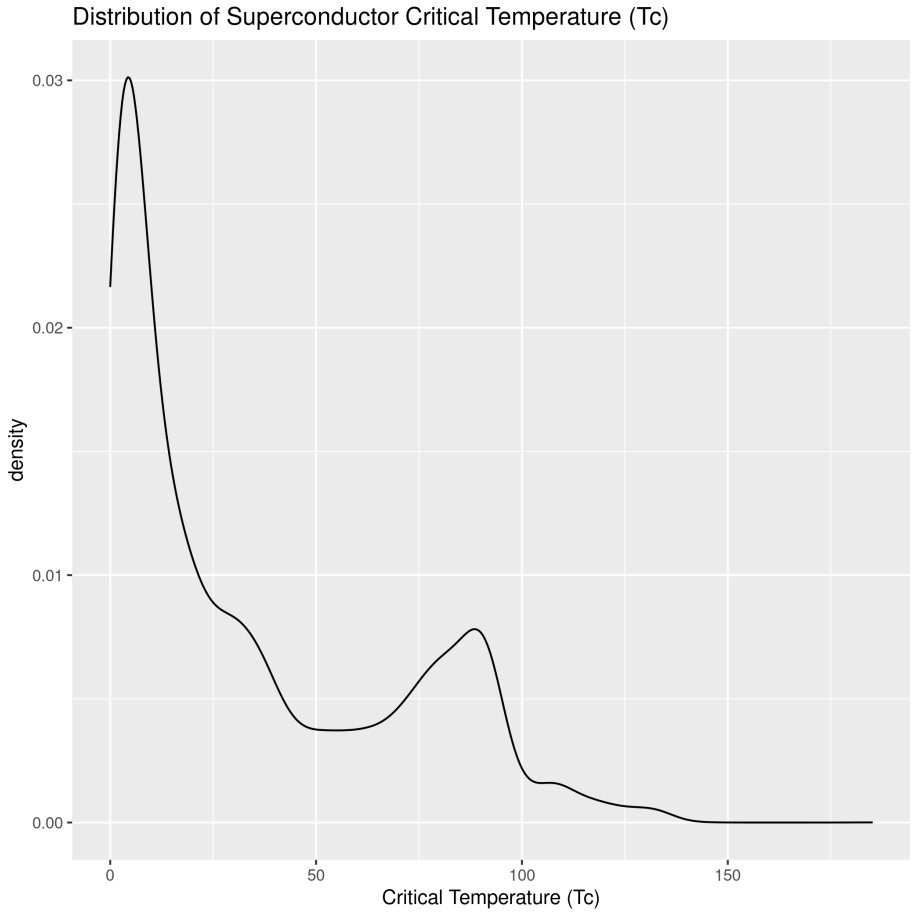

FIGURE 1. Density plot of T_c values for all superconductors in dataset.

Besides determining the performance of XGBoost on these four subsets, we will analyze any potential benefits of training XGBoost only on compounds of a given element subset when trying to predict T_c values for that subset. The motivation for this is to determine if it would be useful to have separate models for these subsets (as has been done in literature previous to [1]), or if the more generalizable XGBoost model trained on the entire training data gives accurate enough predictions.

Thus, there are two different test conditions for the element subsets.

(1) No Retrain: The XGBoost model will be trained on an unrestricted random partition of the data.

(2) Retrain: The XGBoost model will be trained only on superconductors belonging to a single element subset.

3.2.2. *The True T_c Quartiles.* As one would expect, these subsets are divided by T_c quartile. The motivation for these subsets is again the distribution of T_c values. Since the distribution is skewed towards low values, it is likely that the algorithm under-predicts values in the upper quartiles.

Identifying this under-prediction (or any other systematic error) may lead to a better optimized algorithm which takes a likely under-prediction of certain superconductors into account.

3.2.3. *The Predicted T_c Quartiles.* These subsets are the superconductor quartiles, where each superconductor is placed in a quartile according to its *predicted* T_c value. The motivation for this subset is identical to the motivation for the true T_c quartiles, with one distinction. With the predicted T_c quartiles, we will take into account the fact that we do not know the true T_c of an arbitrary conductor. Because of this, implementing a correction term based off of true T_c values would prove difficult, if not impossible.

If a systematic errors can be identified within the predicted T_c quartiles, then implementing a correction term would be easy. We would only need to pick the suitable correction term based off of the observed predicted value of T_c.

3.2.4. *Control Data.* We will also analyze the performance of XGBoost on a simple 2/3 training and 1/3 testing partition of the superconductor data.

3.3. **Collection of Error Statistics.** The collection of error statistics was as follows:

(1) The dataset described in section 2 is split into 2/3 training data 1/3 testing data.

(2) The model is trained and tested on the corresponding partitions, and residuals are taken for each prediction on the testing data.

(3) Summary statistics of the residuals are taken over the classes of superconductors described in 3.2.

In the case of the retraining condition on the element subsets, an additional step is added before step (1), where the entire dataset is partitioned into element subsets, and steps (1)-(3) proceed using a single element subset.

4. Results

4.1. **Results on the Control Data.** Figure 2 presents average error statistics from the 50 error vectors collected for the control data. Note that the average RMSE is of about 9.5° K, thus we can confirm the out-of-sample RMSE presented in [1].

	RMSE	ave_err	std_err	under_cnt	over_cnt	ave_under	std_under	ave_over	std_over
1	9.43	-0.14	9.427	3615.94	3472.06	-5.222	7.588	5.153	8.148

Figure 2. Average error statistics from 50 random test/train splits.

Moreover, we see that we have a mean error near 0, and a standard deviation of error nearly equal to the RMSE. This is characteristic for normally distributed residuals,

and indicates that the RMSE can be attributed to random error, rather than a bias in the model.

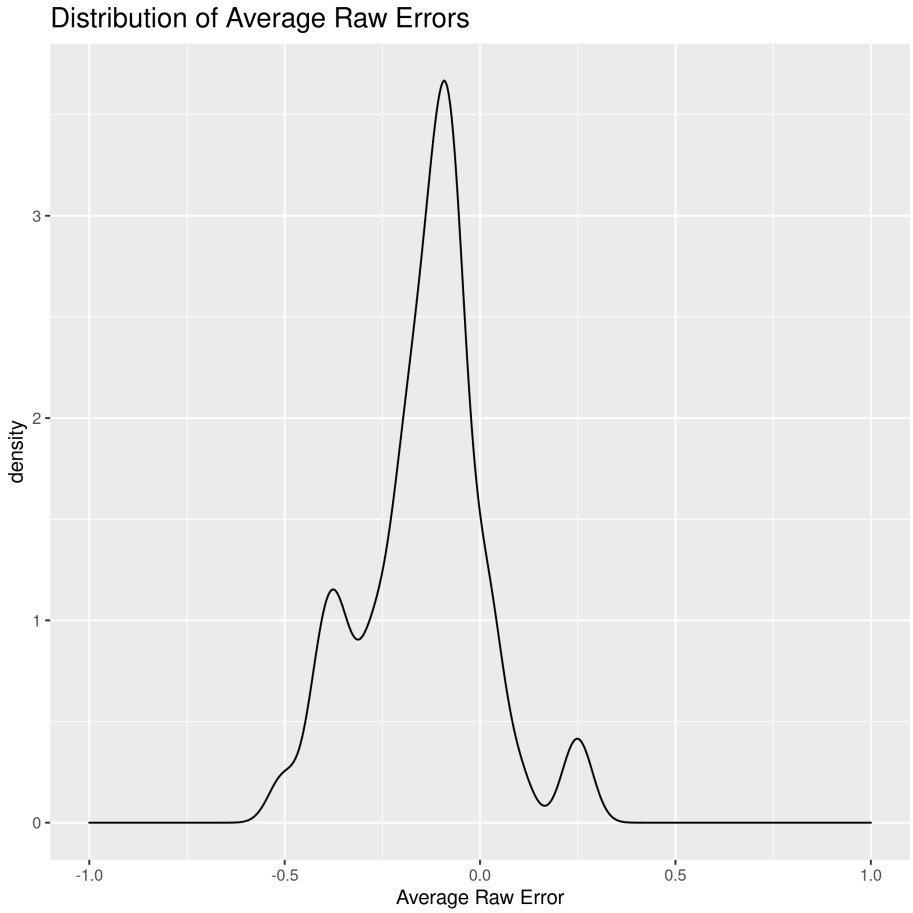

FIGURE 3. Density plot of average raw errors. Raw error averages were taken from 50 random train/test splits.

By plotting the average raw error from 50 random train/test splits in Figure 3, we do see that the model has a slight under-prediction bias, since the distribution of average errors is skewed towards the left. This is not surprising, since the distribution of superconductor critical temperature is heavily skewed towards low critical temperatures (as seen in Figure 1).

Figure 4 more clearly shows the nature of the slight under-prediction bias. In particular we see that:

(1) XGBoost correctly finds a sharp peak in the distribution of T_c values near a temperature of 0° K. However, the model slightly over-estimates the number of superconductors present in this low T_c cluster.

(2) XGBoost correctly identifies a second smaller peak in the distribution with high T_c values. However, XGBoost over-estimates the number of superconductors present in this high T_c cluster, and slightly under-estimates the temperature at which this cluster occurs.

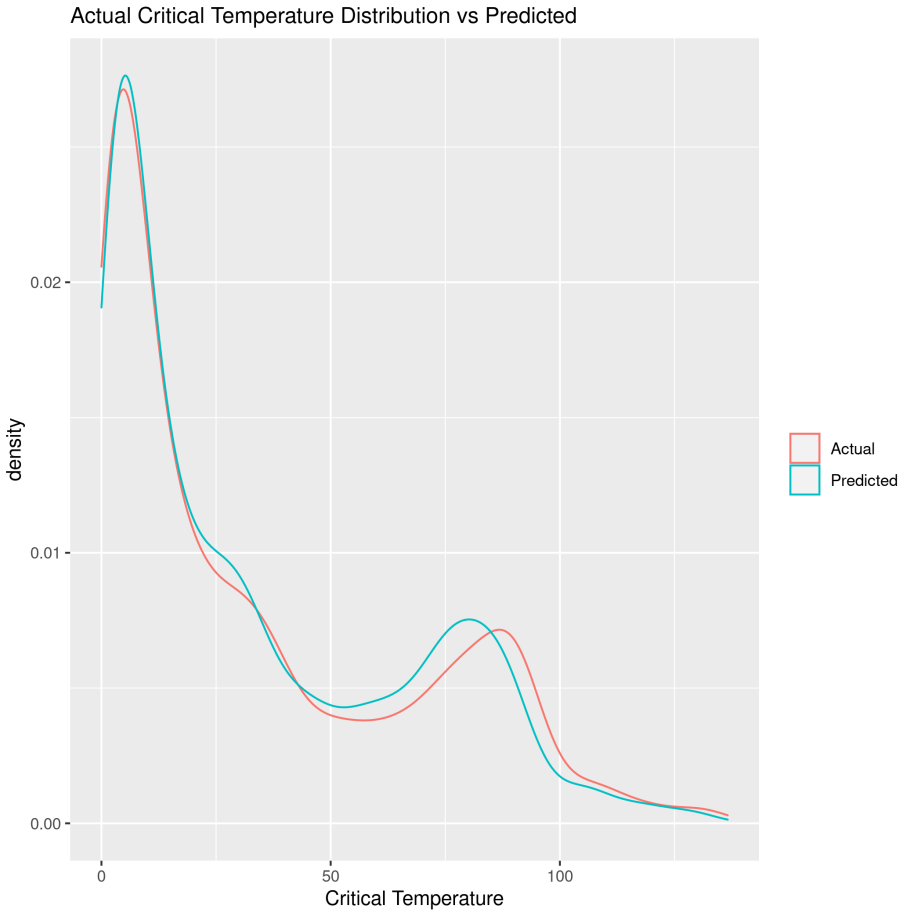

FIGURE 4. Actual test partition T_c distribution plotted alongside the predicted T_c distribution.

4.2. **Results on the Element Subsets.** Figure 5 presents average error statistics for the 4 element subsets when XGBoost is trained on a random train partition in the data. Thus, the XGBoost model used to collect these error statistics can be taken to be identical to the model in the control data section.

	RMSE	ave_err	std_err	under_cnt	over_cnt	ave_under	std_under	ave_over	std_over	Element
1	8.303	-0.129	8.296	410.54	367.52	-4.989	6.255	5.296	6.750	Fe
2	13.867	-0.299	13.825	159.72	124.56	-8.654	8.649	10.424	11.596	Hg
3	12.257	-0.272	12.251	2005.34	1611.68	-7.538	8.346	8.769	10.126	Cu
4	9.008	-1.541	8.247	14.62	2.50	-4.469	3.276	16.320	7.027	B2Mg

FIGURE 5. Average error statistics from 50 error vectors per element subset.

As expected, we see that the model performed worst on superconductors containing mercury, with a RMSE of 13.867. None of the element subsets show a clear prediction

bias towards high or low T_c predictions, with the possible exception of B_2Mg based superconductors. The model under-predicted T_c for these superconductors approximately six times more times than it over-predicted. However, we can also see that this element subset had a very small sample size, which is the likely source of the poor predictions.

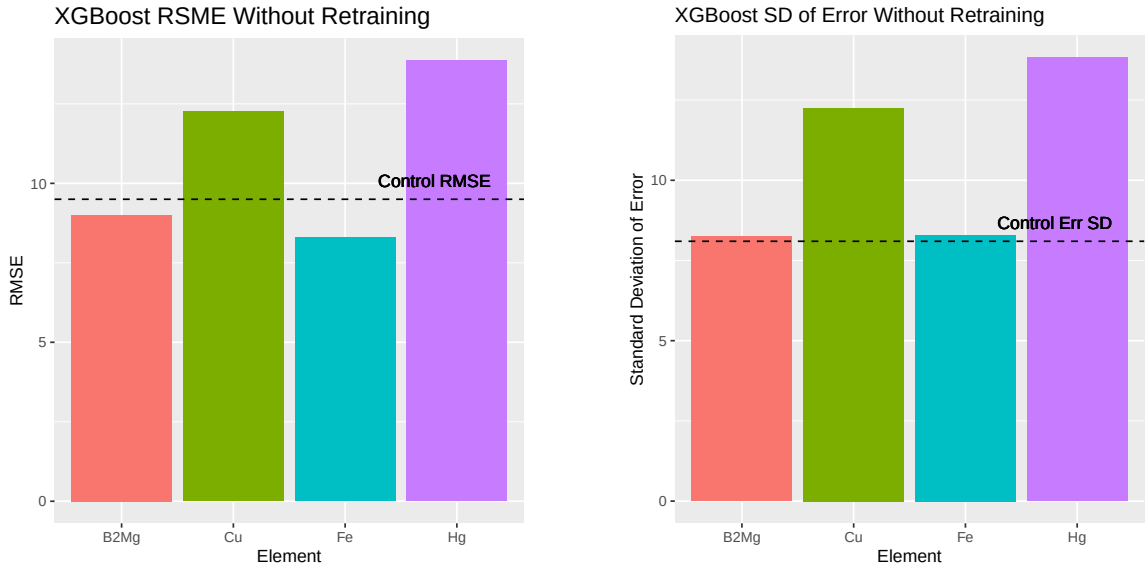

FIGURE 6. RMSE and standard deviation of error on the element subsets without retraining XGBoost on the element subsets. The RMSE and standard deviation of error for the control data are indicated with dashed lines.

Initially, I had suspected that superconductors containing Hg would have a severe under-prediction bias, since Hg containing superconductors have the highest average T_c. However, we do not see this bias, as the average error is near zero and we see that the number of under predictions is nearly the same as the number of over predictions.

The overall performance of the model on the element subsets is better seen in Figure 6. Here, we can clearly see that the model performed worse on copper and mercury containing superconductors than the reported RMSE of 9.5 in [1]. Moreover, we can see that this poorer performance is likely due to random error in the model's predictions rather than some correctable bias, since the standard deviation of the prediction residuals closely mirrors the RSME.

It is important to note that while the RMSE is up to 4°K higher in the element subsets than in the control data, the distribution of superconductor T_c ranges from nearly 0°K to 150°K. Thus, the model predicts T_c reasonably well across all element subsets.

Figure 7 shows that retraining the model on only superconductors belonging to the elemental subset on which it is tested on did not improve the performance of XGBoost. It is interesting to note that the model's performance in both the Retrain and No Retrain conditions was very similar, despite drastic changes to the training data. I

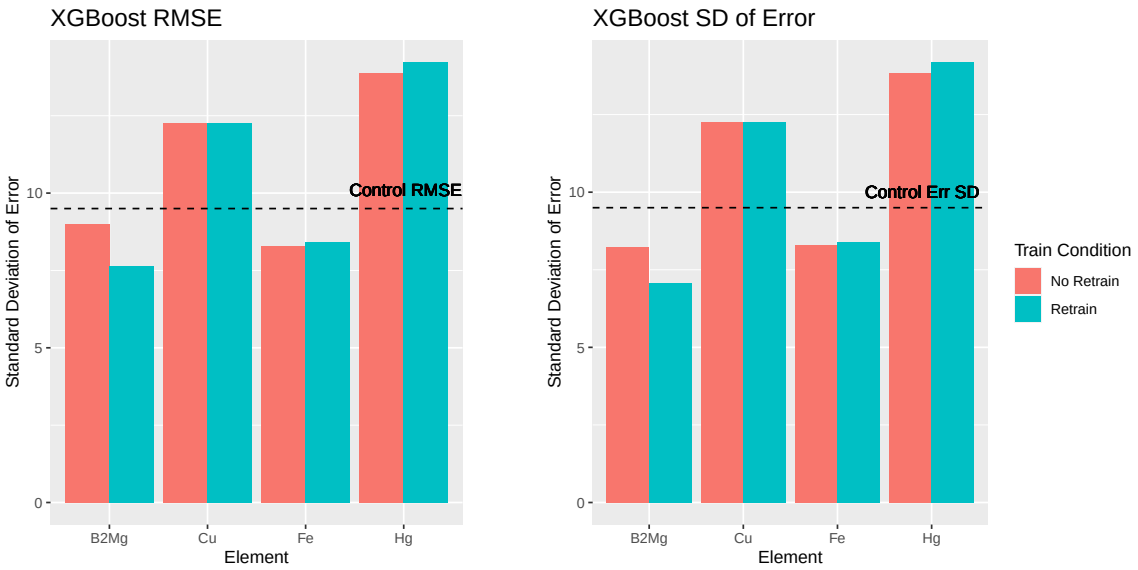

FIGURE 7. Comparison of RMSE and Standard deviation between re-training and no-retraining conditions

	RMSE	ave_err	std_err	under_cnt	over_cnt	ave_under	std_under	ave_over	std_over	Element
1	8.416	-0.035	8.409	407.86	370.20	-4.982	6.424	5.412	6.763	Fe
2	14.247	-0.162	14.205	156.52	127.76	-8.883	9.239	10.522	11.657	Hg
3	12.254	-0.262	12.248	2006.78	1610.24	-7.529	8.318	8.793	10.136	Cu
4	7.649	-0.216	7.065	14.06	3.06	-3.048	1.281	13.689	7.638	B2Mg

FIGURE 8. Average error statistics from 50 error vectors per element subset. In this case, XGBoost was retrained on the elemental subsets.

	RMSE	ave_err	std_err	under_cnt	over_cnt	ave_under	std_under	ave_over	std_over	Quartile
1	4.263	1.446	4.008	413.08	1355.2	-0.515	0.551	2.043	4.394	1
2	7.418	1.582	7.247	981.7	790.08	-1.742	1.878	5.714	9.086	2
3	11.4	1.331	11.32	962.14	815.98	-5.705	5.884	9.628	10.542	3
4	12.311	-4.926	11.28	1259.02	510.8	-9.111	10.195	5.391	5.899	4

FIGURE 9. Average error statistics from 50 error vectors collected for each true T_c quartile.

believe that this might be due to the the fact that XGBoost is a gradient boosted decision tree algorithm. Thus, by reducing the training sample to only the element subsets, we may have simply re-created the portions of the decision trees in XGBoost which lead to a prediction for superconductors containing Fe, Cu, Mg or B_2Mg.

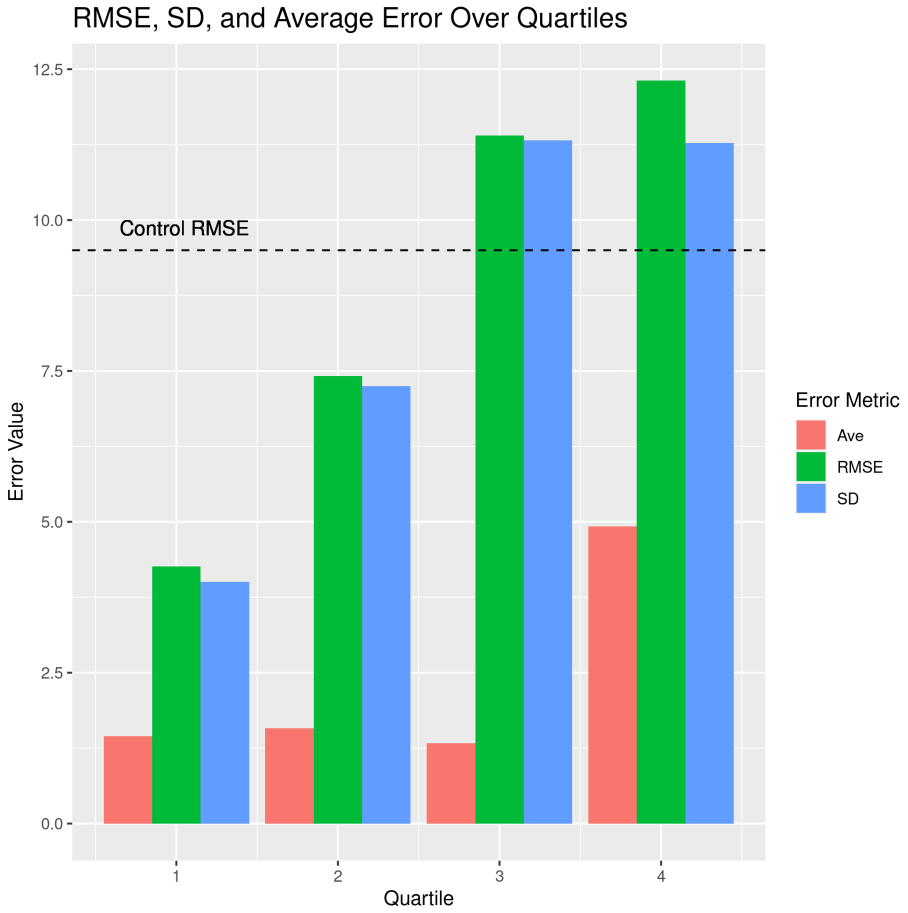

FIGURE 10. RSME, standard deviation of error, and the average of raw errors plotted alongside each other for each true T_c quartile.

4.3. **Results on the True T_c Quartiles.** Figure 9 presents average error statistics for the 4 true T_c quartiles. While the model performs reasonably well across all quartiles, we can see that the model performs best for superconductors with low T_c, and worse on superconductors with a high T_c.

This is not ideal, since the primary use-case of this model would be to find a superconductor with a high T_c. However, Figure 9 suggests that the XGBoost model is likely to under-predict the critical temperature of such a material.

However, by looking at the average error, we see that for the 4th quartile, the model has a -5° K bias. Thus, it might be possible to correct for under predictions in the 4th quartile, and so slightly improve the performance of the model in this quartile.

4.4. **Results on the Predicted T_c Quartiles.** Of course, we cannot know in which T_c quartile a superconductor with an unknown T_c will belong. Thus, I investigated if this bias is still present if each superconductor is placed in a quartile based off of its *predicted* T_c.

	RMSE	ave_err	std_err	under_cnt	over_cnt	ave_under	std_under	ave_over	std_over	Quartile
1	2.401	0.15	2.394	622.86	1149.14	-1.32	3.296	0.948	0.93	1
2	5.452	-0.143	5.449	964.54	807.46	-2.9	5.33	3.153	3.294	2
3	13.025	-0.169	13.018	943.78	828.22	-8.274	10.27	9.067	9.048	3
4	12.233	-0.398	12.221	1084.76	687.24	-6.869	6.504	9.817	12.149	4

FIGURE 11. Average error statistics from 50 error vectors collected for each predicted T_c quartile.

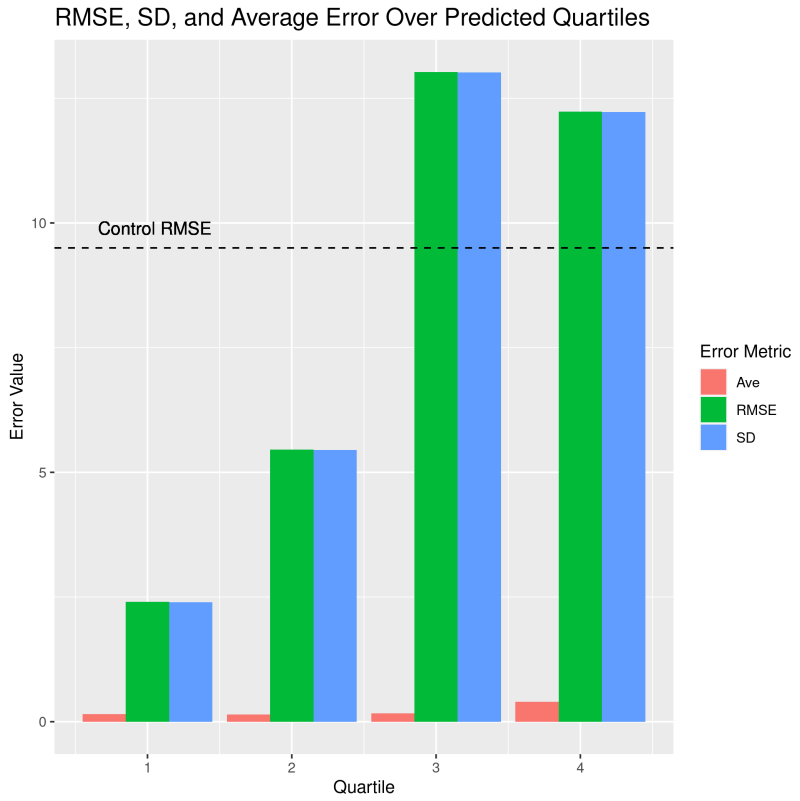

FIGURE 12. RMSE, standard deviation of error, and the average of raw error plotted alongside each other for each predicted T_c quartile.

Unfortunately, Figures 11 and 12 show that this is not the case. While we still see that the performance of the model worsens in the higher quartiles, there is no bias in the predictions for any quartile.

Interestingly, we see that the RMSE in the 3rd quartile was higher than in the 4th. This may be due to the fact that superconductors belonging to the 4th true T_c quartile are mistakenly predicted to belong in the 3rd quartile, due to the under prediction bias we had seen earlier. Thus, the 3rd quartile is likely to contain many superconductors with under predicted T_c.

5. Conclusion

In this report, we analyzed the performance of XGBoost across three categories of superconductors in order to answer the following three questions:

(1) How well does XGBoost preform at predicting T_c of Iron, Cuprate, Mercury and MgB_2 based super conductors? Does this performance improve or worsen when trained only on Iron, Cuprate, or MgB_2 based superconductors?

(2) If we divide the testing data by quartiles of T_c, how well does XGBoost preform at predicting T_c of compounds in each quartile? Is XGBoost reliable when predicting the T_c of a compound with a high T_c?

(3) If we divide the predictions of XGBoost by quartiles of predicted T_c, how accurate and precise is XGBoost's prediction when the prediction falls in a given quartile? Is the predicted value of T_c reliable when this predicted value is high?

We are now in a position to provide answers to these three questions:

(1) XGBoost predicts T_c for Iron, Cuprate, and Mercury based superconductors reasonably well; in each case XGBoost predicts T_c with an RMSE nearly equal to the RMSE of T_c predictions for the control data (9.5° K), but was higher than the control RMSE for cuprate and mercury based superconductors. The performance of XGBoost at predicting T_c for these subsets does not significantly change when using training data only from these subsets.

(2) XGBoost preforms reasonably well across all quartiles, but RMSE does increase for increasing T_c quartiles. Thus, the XGBoost model presented in [1] is reliable for predicting the critical temperature of materials in the 4th quartile, but less reliable than it would be for lower quartiles. This increase in RMSE is partially driven by an increase in prediction bias in the 4th quartile, with XGBoost consistently under predicting T_c in this quartile.

(3) As observed in the true T_c quartiles, XGBoost performs with decreasing accuracy for increasing T_c quartiles, but is still reasonably accurate. Thus, the model is reasonably reliable when the predicted T_c value is high. Unlike for the true T_c quartiles, there is no observable bias in the predictions for the 4th predicted T_c quartile.

References

[1] Kam Hamidieh. "A data-driven statistical model for predicting the critical temperature of a superconductor". In: *Computational Materials Science* 154 (Nov. 2018), pp. 346–354. ISSN: 09270256. DOI: 10.1016/j.commatsci.2018.07.052. URL: https://linkinghub.elsevier.com/retrieve/pii/S0927025618304877 (visited on 12/14/2020).

[2] *Superconductivity Data*. UCI Machine Learning Repository. Nov. 2018. URL: https://archive.ics.uci.edu/ml/datasets/Superconductivty+Data (visited on 12/14/2020).