Predicting the critical temperature of Cu, O and Fe based superconductors using Machine learning techniques

Mohamed ABDUL GAFOOR
Department of Computer Science
Cork Institute of Technology

Abstract:
Superconductors have enormous applications in modern world. Even though phenomenon of superconductivity has known to the scientific community a century ago, some of the crucial features predicting the superconductivity remains unclear. In this paper several regression models have been studied to predict the critical temperature (Tc) of Cu/O/Fe-based superconductors. The Root-Mean-Square-Error of around 12.64K obtained for Oxygen-based, 12.91K obtained for Copper-based and 9.23K obtained for Iron-based superconductors. We observed Cuprate and Oxygen-based superconductors form two clusters approximately close to 25K and 85K, where as in Iron-based superconductors no discernible two clusters formed.

1. Introduction

The electrical resistance of certain materials entirely disappear at very low temperatures, is one of the most exciting phenomenon in theoretical condensed matter physics. In 1911, a Dutch physicist, Heike Kamerlingh Onnes and his assistant observed the phenomenon of superconductivity while investigating the resistance of materials at low temperature. When the electrons move through the materials at very low temperature (below the critical temperature, Tc), those electrons experience zero resistance. This means, there is no heat, sound or any other form of energy loss during this state. In Figure 1, this can be seen clearly in the case of mercury which exhibits a sudden drop in the electrical resistance at 4.2 K.

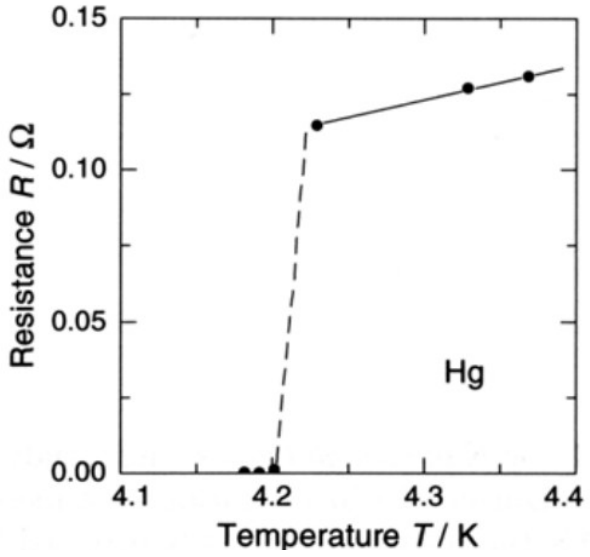

Figure 1: The resistance of Hg measured by Heike Kamerlingh Onnes

After this discovery many physicists observed similar behavior in other metals when they cooled down to a certain temperature.

Compound	T_c (K)	Compound	T_c (K)	Compound	T_c (K)
Nb_3Sn	18.1	MgB_2	39	UPt_3	0.5
Nb_3Ge	23.2	$PbMo_6S_8$	15	UPd_2Al_3	2
Cs_3C_{60}	19	YPd_2B_2C	23	$(TMTSF)_2ClO_4$	1.2
Cs_3C_{60}	40	$HoNi_2B_2C$	7.5	$(ET)_2Cu[Ni(CN)_2]Br$	11.5
High-T_c superconductor		T_c (K)		High-T_c superconductor	T_c (K)
$La_{1.83}Sr_{0.17}CuO_4$		38		$Tl_2Ba_2Ca_2Cu_3O_{10+x}$	125
$YBa_2Cu_3O_{6+x}$		93		$HgBa2Ca_2Cu_30_{8+x}$	135
$Bi_2Sr_2Ca_2Cu_3O_{10+x}$		107		$Hg_{0.8}Tl_{0.2}Ba_2Ca_2Cu_30_{8.33}$	134

Figure 2: Superconducting critical temperature for different metals.

Superconductors have many practical applications because of its ability to conduct electricity with zero resistance. Thus it could provide energy savings in a wide range of applications from Magnetic Resonance Imaging (MRI) to magnetic levitation (Maglev) trains. Other prominent applications include the superconducting coils used to maintain high magnetic fields in the Large Hadron Collider at CERN, where the existence of Higgs Boson was recently confirmed, and the extremely sensitive magnetic field measuring devices called SQUIDs (Superconducting Quantum Interference Devices)[1]. Nevertheless, superconductors are restricted by their operating temperature which usually at around 40k. There has been more anticipation on Copper (Cu) based materials called cuprates, but recent iron-based (Fe) materials have attained fascinating interest as a second class of high-temperature superconductors. Owolabi et al have developed predicting Support Vector Machine model for Fe-based superconductors and found strong relation between lattice parameters and the superconducting temperature[2]. Extensive databases covering several measured and calculated materials properties have been created over the years[3]. Figure 2 shows some of the high-temperature cuprates superconductors with their critical temperature. In this study we focus on three superconductor types which have the elements of either Copper, Oxygen or Iron. Our purpose is predicting the critical temperature (Tc) of these superconductors based on its chemical formula. For our study the original superconductor data have been taken from the Superconducting Material Database maintained by Japan's National Institute for Materials Science (NIMS). Then the data has been preprocessed and 21, 263 superconductors have obtained by Hamidieh et al and it is available at UCI Machine Learning Repository.

2. Research

In this study the XGBoost and Random Forest Machine learning Models have been deployed to predict the critical temperature of the superconductors. Multilinear Regression model is used as benchmark to evaluate the model performance. At the beginning several machine learning models have been tested, including KNN, SVM, Multilayer Perceptron (MLP). None of them performed well. Since MLP took more than 6hrs to run a single iteration without dimensionality reduction (without PCA), with numerous parameters to tune the model, I decided to terminate the process after the first iteration, yet the performance was quite low for this dataset.

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/m^3)	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 ($W/(m \times K)$)	thermal conductivity coefficient κ
Valence	no units	typical number of chemical bonds formed by the element

Table 1: This table shows the properties of an element which are used for creating features to predict T_c.[1]

After extracting the original datasets from NIMS, the boiling point variable has been dropped as it contains some missing values and the fusion heat variable is added as it contains no missing values[1]. But it is worth noting here that these two variables are highly correlated, hence no negative influence can occur in the predicted temperatue. At the end about 67% of the original data has been extracted from NIMS and it is available at UCI Machine Learning Repository[1].

From which, another subset of data is extracted to extensively study about the cuprates, iron and oxygen based superconductors. Out of 21, 263 superconductors, we searched for any missing values, fortunately there is no columns or rows contain NaN values. Moreover we checked the percentage of the components that we are interested in, as a results we found, Oxygen based superconductors is roughly 56%, Copper-based superconductors is around 51% and Iron-based superconductors is around 11%. Here we have decided to choose the iron because iron is commonly used element in our daily applications and recently, optimism of reaching room temperature in superconductivity has been highlighted when iron based superconductor was discovered. Finally, we combined both train.csv and unique_m.csv (only selected columns contain Cu, O, Fe) from UCI repository to form a *big-data*. Then we have created another three sets of data which contains only Cu, only O, only Fe. The length of the original dataset was 21, 263 as we mentioned earlier. After we manipulate, the data contains Cu has a length of 10838, O has the length of 11964 and Fe has the length of 2339.

2.1 Feature extraction

The Table 1 shows the variables and the Table 2 shows the features that we are interested in this study. We can extract 10 features from 1 variable. Since we have 8 variables (see Table 1), 80 features can be obtained. The 81st feature is the number of elements in the superconductor. Hence in total we have 82 columns (82nd is the critical temperature) and 21, 263 rows.

In Table 2, the last column is a sample calculation; features based on thermal conductivities for Re_6Zr_1 are derived and reported to two decimal places. Rhenium and Zirconium's thermal conductivity coefficients are $t_1 = 48$ and $t_2 = 23$ W/(m×K) respectively. Hence the mean thermal conductivity is 35.5 W/(mK).

The ratio of the elements in the material of Re_6Zr_1 is; $p_1 = 6/7$, $p_2 = 1/7$
The fraction of thermal total energy; $w_1 = t_1/(t_1+t_2) = 48/(48+23) = 48/71$, $w_2 = 23/71$
The weighted value of themal conductivities; $A = p_1w_1/(p_1w_1+p_2w_2) = 0.926$, $B = 0.074$

Feature & Description	Formula	Sample Value
Mean	$= \mu = (t_1 + t_2)/2$	35.5
Weighted mean	$= \nu = (p_1 t_1) + (p_2 t_2)$	44.43
Geometric mean	$= (t_1 t_2)^{1/2}$	33.23
Weighted geometric mean	$= (t_1)^{p_1} (t_2)^{p_2}$	43.21
Entropy	$= -w_1 \ln(w_1) - w_2 \ln(w_2)$	0.63
Weighted entropy	$= -A \ln(A) - B \ln(B)$	0.26
Range	$= t_1 - t_2 \ (t_1 > t_2)$	25
Weighted range	$= p_1 t_1 - p_2 t_2$	37.86
Standard deviation	$= [(1/2)((t_1 - \mu)^2 + (t_2 - \mu)^2)]^{1/2}$	12.5
Weighted standard deviation	$= [p_1 (t_1 - \nu)^2 + p_2 (t_2 - \nu)^2)]^{1/2}$	8.75

Table 2: This table summarizes the procedure for feature extraction from material's chemical formula[1].

To speed-up the algorithm and to select the features that contributes the most for the prediction of the critical temperature, *principal components analysis* (PCA) was used. It helps to reduce dimension of a dataset from a large set to small set that still contains most of the information. It transforms number of correlated variables into a smaller number of uncorrelated variables name as principal components. The Figure 2.2 shows the geometry of principal components.

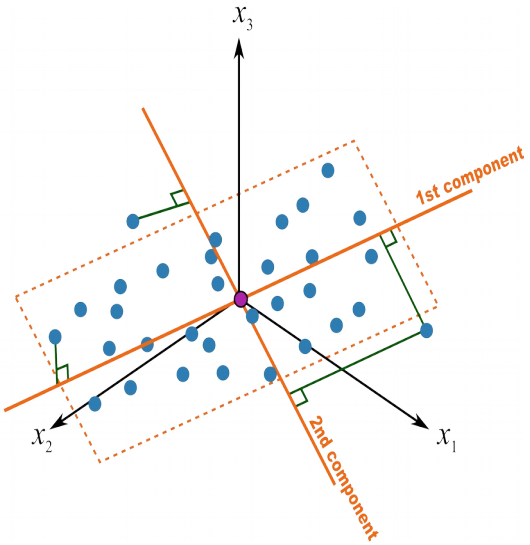

Figure 2.2 : The geometry of principal components

3. Methodology

The performance of the model is evaluated using the RMSE, best_score and R2 values. The out-of-sample values are estimated as follow;

Out-of-Sample estimation.

1. The data is randomly divided into 2/3 to train and 1/3 to test.
2. The model is fitted using the train data.
3. The critical temperature is predicted using test data.
4. The root mean square error (RMSE), best_score and R2 value are calculated for 5 iteration.
5. Found the average values of RMSE, best_score and R2 values.

One of the problem in any machine learning algorithm is limited access to data. Hence over-fitting is one of the main issue that we must face. To address this issue, a well-known approach is to use cross-validation. It is a process of splitting the dataset into X parts and take each one of them as a test set and use the rest to train the model. At the end the performance indicators are averaged. In fact there is no universally accepted evaluation methodology for evaluating the machine learning algorithms, every method has its own pros and cons. In our study we have chosen 10 Fold cross-validation, although we have checked the 20, 30 and 40 fold validation and found 10 is doing reasonably good. For example the 40 fold cross validation only increase the performance by 1.5%.

3.1 Extreme Gradient Boosting (XGBoost) framework

XGBoost is a machine learning algorithm that has newly been influencing in every Kaggle competitions these days. It is an implementation of gradient boosted decision trees, which has been designed for better performance and speed. This was first proposed by Tianqi Chen et al in 2016[4]. Whether regression or classification task, XGBoost performing well in both types. Since it is written in C++, it is relatively faster compare to other ensemble techniques and very useful because XGBoost is parallelizable, power of multi-core computers can be utilized.

The functional form of XGBoost can be written as follow[1],

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i),$$

Where x_i is the i^{th} input feature vector, $\hat{y}_i$ is the predicted response, and f_1, ...f_K is a sequence of trees. The i^{th} tree f_t is added by minimizing the following objective function:

Objective with respect to f_t is given by,

$$\sum_{i=1}^{n} L(\underbrace{y_i}_{observed}, \underbrace{\hat{y}_i^{(t-1)} + f_t(x_i)}_{predicted}) + \Omega(f_t),$$

Here L is the desired loss function, n is the total sample size, y_i's are the response values, $\hat{y}_i^{(t-1)}$ is the i^{th} predicted response at the t-1 step, and Ω is the penalty function. The form of the Ω is:

$$\Omega(f) = \gamma T + (1/2)\lambda \sum_{j=1}^{T} w_j^2,$$

Where T is the number of leaves in each tree, w_j is the leaf weight and λ and γ are regularization parameters.

The data for the XGBoost split randomly into 2/3 for the training and 1/3 for the testing the model. The GridSearchCV has been used to fine tune the parameters for the XGBoost. Initially the following parameters have passed onto the GridSearchCV and the rest of the parameters have been set to default.
'n_estimators': [100, 150, 200], 'learning_rate': [0.1, 0.15, 0.2], 'gamma': [0], 'max_depth': [3,5,7], 'subsample': [0.5, 0.75, 1], 'colsample_bytree': [1], 'booster': ['gbtree', 'dart']. Hence the total grid size is 162.

The best model has been obtained as follow;

Oxygen-based

Best params: {'booster': 'dart', 'colsample_bytree': 1, 'gamma': 0, 'learning_rate': 0.1, 'max_depth': 7, 'n_estimators': 200, 'subsample': 0.75}

Copper-based

Best params: {'booster': 'gbtree', 'colsample_bytree': 1, 'gamma': 0, 'learning_rate': 0.1, 'max_depth': 7, 'n_estimators': 200, 'subsample': 0.75}

Iron-based

Best params: {'booster': 'dart', 'colsample_bytree': 1, 'gamma': 0, 'learning_rate': 0.1, 'max_depth': 7, 'n_estimators': 100, 'subsample': 0.75}

Superconductor type	**RMSE**	**best_score**	**R2**
Oxygen-based	12.721854177677583	0.8523858622668411	0.8500645563766662
Copper-based	12.762817548280848	0.8366586463911764	0.839731079432047
Iron-based	9.03169978125998	0.8154306177887648	0.8237778374225339

Table 3: This shows the RMSE, best_score and R2 average values of Oxygen-based, Copper-based and Iron-based after 5 iterations.

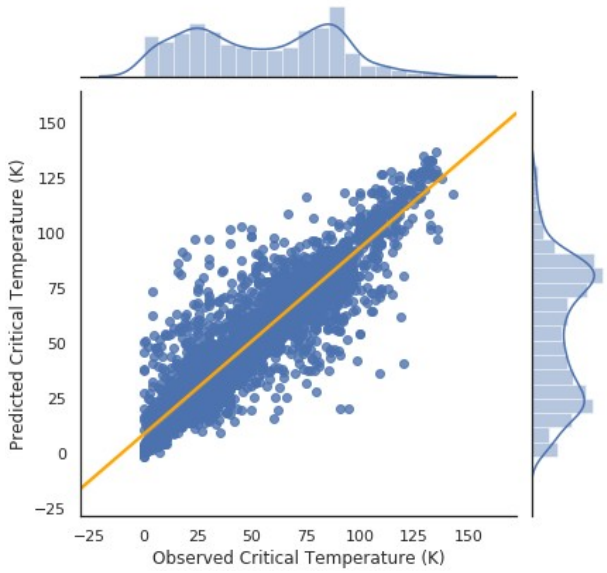

Figure (3.a)

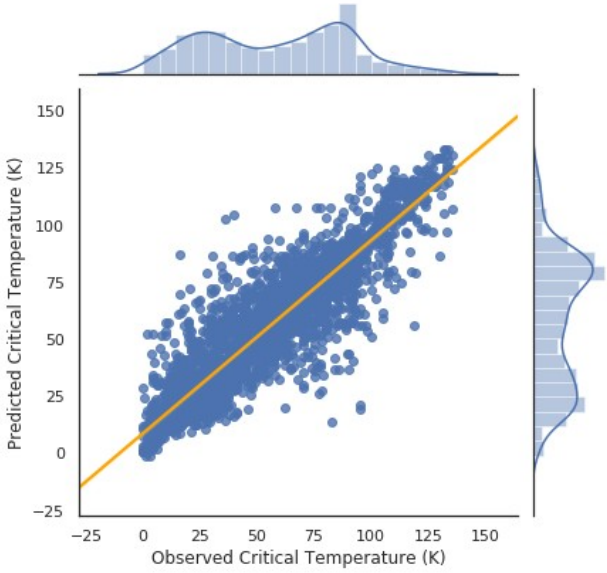

Figure (3.b)

Figure 3: This shows the XGBoost Predicted critical temperature Vs Observed critical temperature. Figure (3.a) for Oxygen-based and Figure (3.b) for Copper-based.

Two main characteristics can be observed from the Figure 3. The first one is that from the observed data in both copper and oxygen based, there are two peaks appear (see the top part of the Figures). This shows clearly that the observed critical temperature can be classified into two major classes, one is around a centroid of 25K and the other centroid is around 85K. Figure 5 shows the kernel density estimation (kde) of observed critical temperature of the Copper-based and Oxygen-based superconductor. kde is a non-parametric way to calculate the probability density function of a random variable. This is very useful to visualize the *"shape"* of the data that we are interested in.

The second one is the best-fit line is more or less centered around the scatter plot, implies no severe biases. However, in Figure 4, no two peaks are observed in Iron-based based superconductors. Figure (4.a) clearly shows a single skewed type profile in the kde plot.

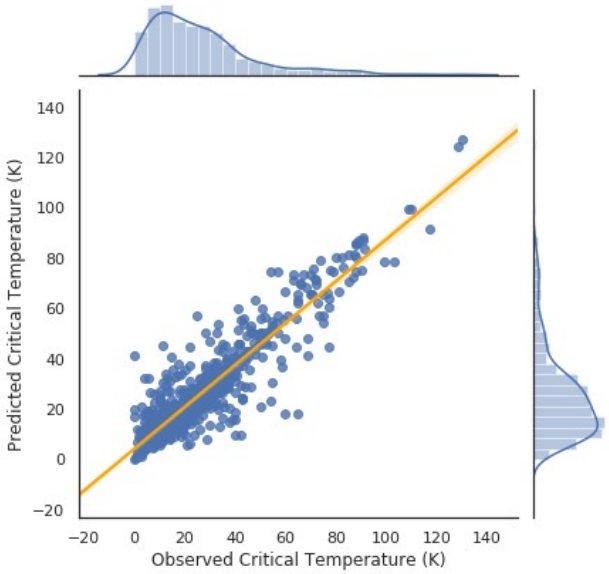

Figure (4.a)

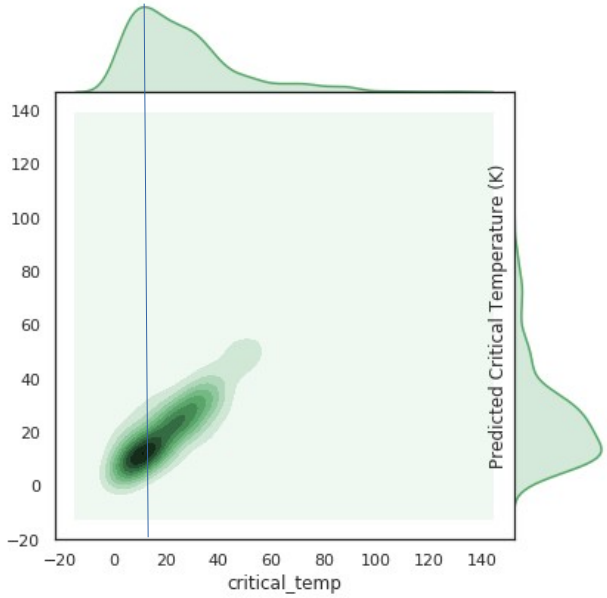

Figure (4.a)

Figure 4: This shows the XGBoost Predicted critical temperature Vs Observed critical temperature for the Iron-based superconductors and the corresponding kde plot.

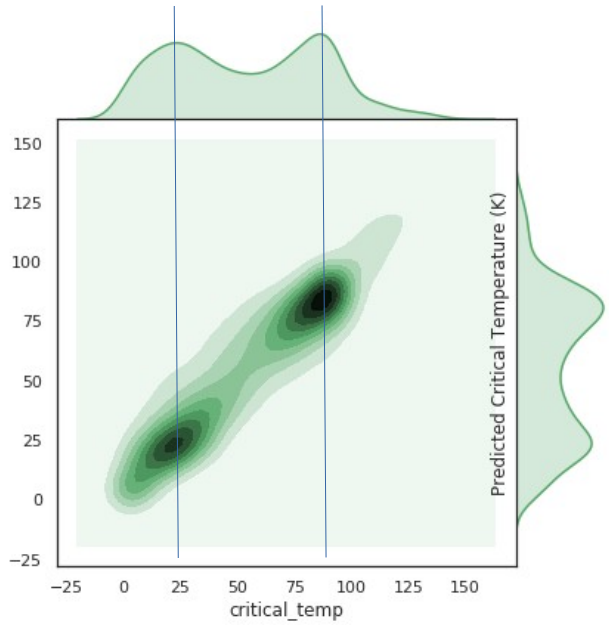

Figure (5.a) Oxygen-based

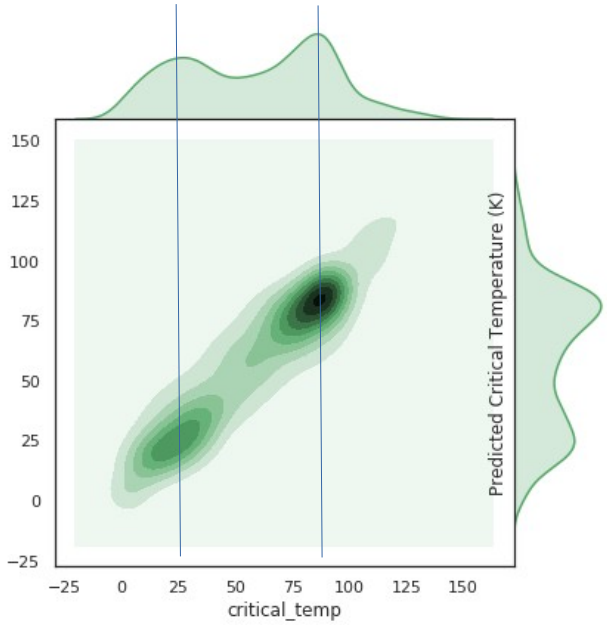

Figure (5.a) Copper-based

Figure 5: This shows the kernel density estimation (kde) of observed critical temperature (XGBoost).

3.2 Random Forest Regressor framework

Random forests or random decision forests are an ensemble learning method for classification, regression problems on various sub-samples of the dataset. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement if *bootstrap=True* (default). Random forests build many decision trees, though we can control the number of trees built, and yeilds the importance features by averaging uncertainity reduction obtained by all the features across all the trees. The default measure of uncertainty used in a random forest is the Gini index.

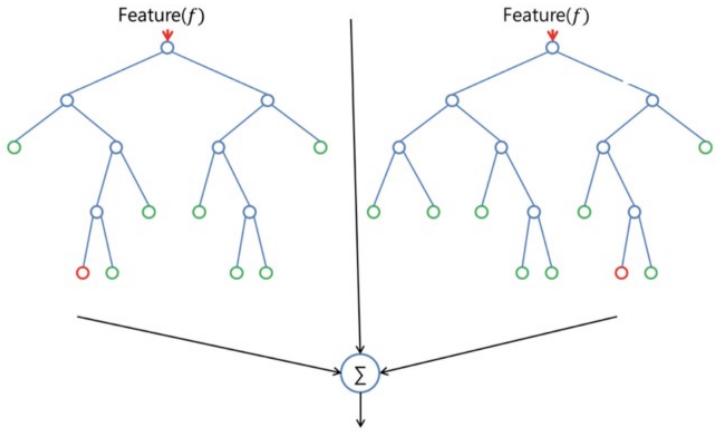

Figure 6: A typical random forest diagram

Figure 6 show that there are several decision trees as base learners. Every tree is given a subset of random data. This algorithm uses Bagging (Bootstrap Aggregating) techniques from ensemble methods. The basic framework is that each base leaner is trained on a different subset of random data. Hence finally it predicts based on the majority of votes if it is a classification problem or aggregation if it is a regression problem from each of the decision trees made. The data for the Random Forest (RF) split randomly into 2/3 for the training and 1/3 for the testing the model. The GridSearchCV has been used to fine tune the parameters for the RF. Initially the following parameters have passed onto the GridSearchCV and the rest of the parameters have been set to default.
"n_estimators": [10,20,30], "criterion": ['mse'], "max_features" : ["sqrt", "log2"], "min_samples_split": [2,4,8], "bootstrap": [True, False]

The best model has been obtained as follow;
Oxygen based : Best params: {'bootstrap': False, 'criterion': 'mse', 'max_features': 'sqrt', 'min_samples_split': 8, 'n_estimators': 30}
Copper-based: Best params: {'bootstrap': False, 'criterion': 'mse', 'max_features': 'sqrt', 'min_samples_split': 8, 'n_estimators': 30}

Iron-based: Best params: {'bootstrap': False, 'criterion': 'mse', 'max_features': 'sqrt', 'min_samples_split': 8, 'n_estimators': 20}

Superconductor type	RMSE	best_score	R2
Oxygen-based	12.571480302781522	0.8477443060334092	0.8536382169180966
Copper-based	13.0656990330298	0.8309849064293873	0.8331883046408706
Iron-based	9.432939649342208	0.801291597719278	0.8221730017268666

Table 4: This shows the RMSE, best_score and R2 average values of Oxygen-based, Copper-based and Iron-based after 5 iterations.

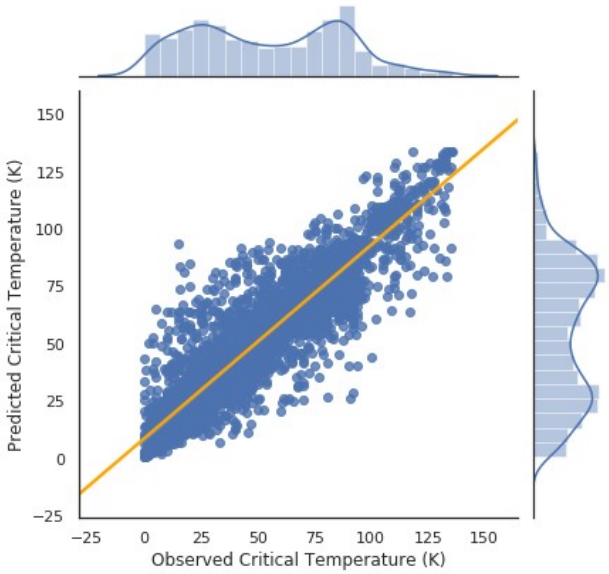

Figure (7.a) Oxygen-based

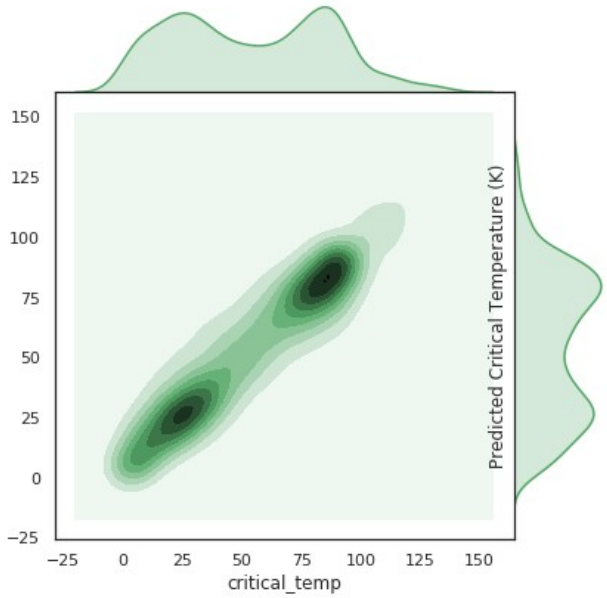

Figure (7.b) Oxygen-based

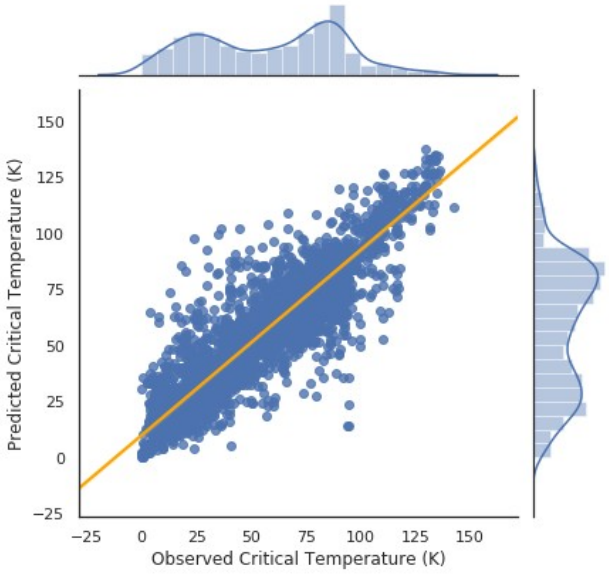

Figure (7.c) Copper-based

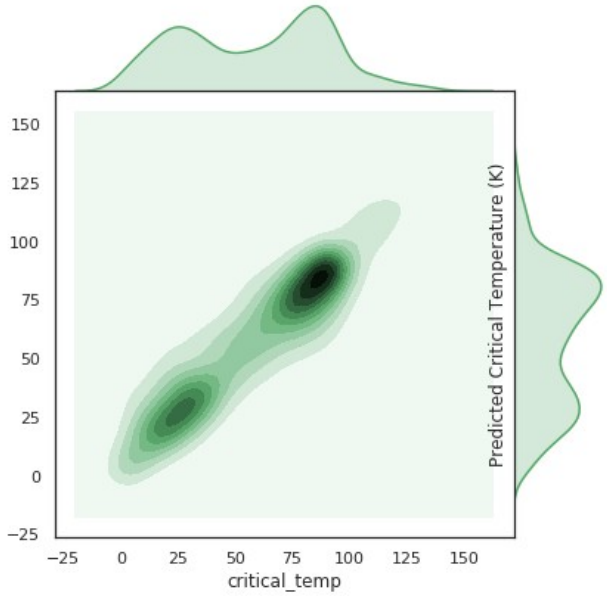

Figure (7.d) Copper-based

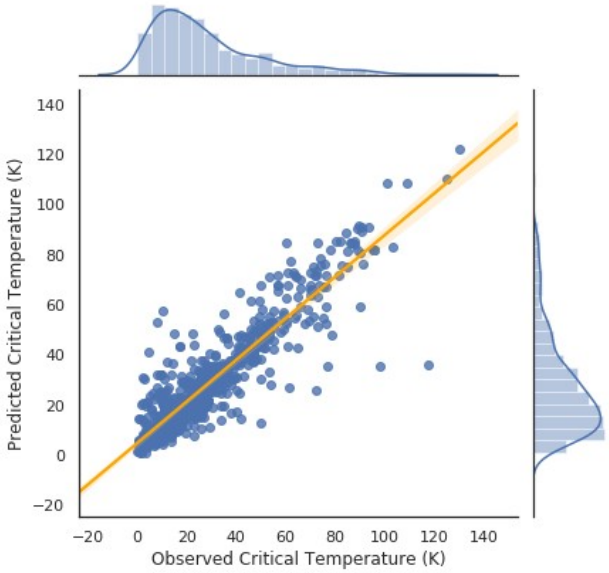

Figure (7.e) Iron-based

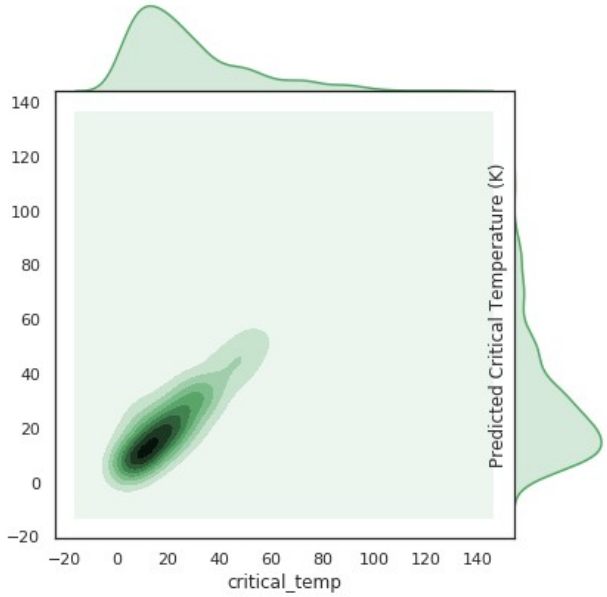

Figure (7.f) Iron-based

Figure 7: This shows the Predicted critical temperature Vs Observed critical temperature and the corresponding kernel density estimation (kde) of observed critical temperature.

	Random Forest	**XGBoost**	**RF/XGB**
Superconductor type	**RMSE**	**RMSE**	**Average**
Oxygen-based	12.571480302781522	12.721854177677583	12.64666724
Copper-based	13.0656990330298	12.762817548280848	12.914258291
Iron-based	9.432939649342208	9.03169978125998	9.232319715

Table 5: The RMSE value comparison of RM and XGBoost framework

3.3 Multiple Linear Regression (MLR)

Multiple Linear Regression is used as a benchmark model to compare the two principal models (XGBoost/RF). Figure 8 clearly shows there is bias in the best-fit. The results obtained for the MLR as follow for the Copper-based,

Average of RMSE: 19.919025054782633
Average of best_score: 0.623255687589374
Average of R2: 0.6182334992524054

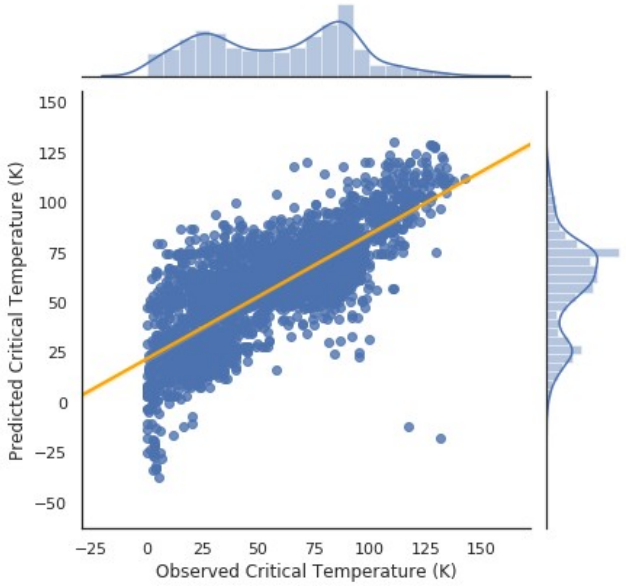

Figure 8: This shows the Predicted critical temperature Vs Observed critical temperature in Copper-based superconductors using MLR

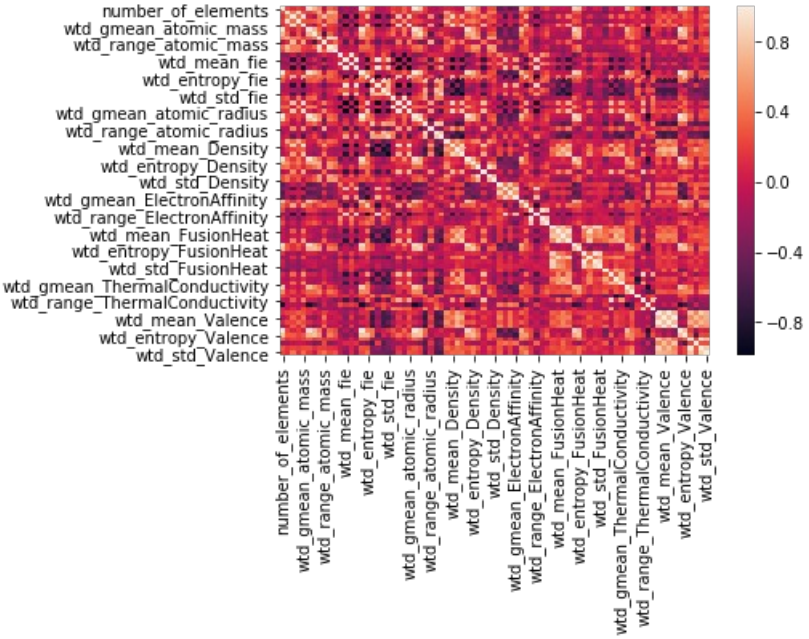

Figure 9: This shows the correlation matrix of the original dataset.

Figure 9 shows that the correlation matrix is symmetric as expected and also possible to see that some variable are perfectly positively correlated but other variables are negatively correlated. Please see the color code next to it.

Correlated Variables	Values
weighted_std_ ThermalConductivity Vs critical_temp	0.721271
range_ThermalConductivity Vs critical_temp	0.687654
range_ atomic_radius Vs critical_temp	0.653759
std_ThermalConductivity Vs critical_temp	0.653632
weighted_mean_Valence Vs critical_temp	-0.632401
weighted_entropy_atomic_mass Vs critical_temp	0.626930
weighted_gmean_Valence Vs critical_temp	-0.615653
weighted_entropy_atomic_radius Vs critical_temp	0.603494
number_of_elements Vs critical_temp	0.601069
range_fie Vs critical_temp	0.600790

Table 6: This shows top 10 correlated variables of Cu/O/Fe-based superconductors

Table 6 shows the top 10 highly correlated features with the critical temperature of Cu/O/Fe-based superconductors. These features include *Thermal Conductivity, Atomic Radius, Valence electrons, Atomic mass, Number of elements and First ionization energy*. Hence it is clear when predicting the critical temperature these are the *seven feature* that contributes the most. But as expected RMSE has increased to 14.27 from 12.91 after the application of PCA.

Analyze with PCA

Out-of-Sample estimation.

1. Standardizing the features
2. The data is randomly divided into 2/3 to train and 1/3 to test.
3. set n_components =30 to cover 99% of the variance in the data
4. Fit the normalized feature space
5. Apply the transformation for both training and the test dataset.
6. The model is fitted using the train data.
7. The critical temperature is predicted using test data.
8. The root mean square error (RMSE), best_score and R2 value are calculated for 5 iteration.
9. Found the average values of RMSE, best_score and R2 values.

			With PCA	Without PCA
	Random Forest	XGBoost	RF/XGB	RF/XGB
Superconductor type	RMSE	RMSE	Average	Average
Oxygen-based	13.36003517413	13.277237502456	13.31863634	12.64666724
Copper-based	13.400985880948	13.464481724431	13.4327338	12.914258291
Iron-based	9.2741501918660	9.1000616962995	9.187105944	9.232319715

Table 7: This table shows the RMSE values after the application of PCA (and before) for both RF and XGBoost framework.

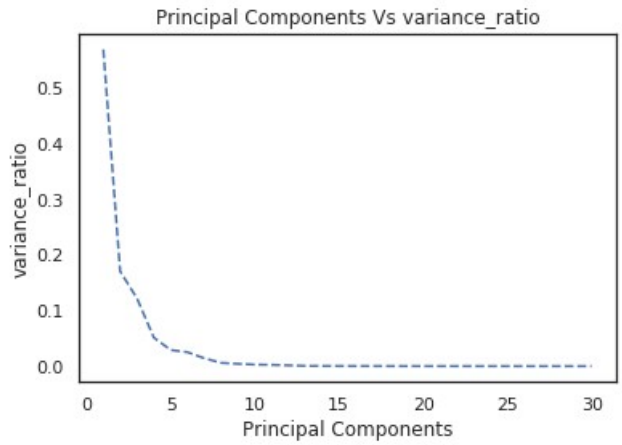

Figure 10: This shows the how the variance ration changes with the principal components.

Conclusion and Evaluation

This study has shown application of two main Machine Learning algorithm to predict the Cu/O/Fe-based Superconductor dataset available at UCI Machine Learning Repository. The Root-Mean-Square-Error of around 12.64K obtained for Oxygen-based, 12.91K obtained for Copper-based and 9.23K obtained for Iron-based superconductors. We observed Copper-based and Oxygen-based superconductors form two clusters approximately close to 25K and 85K, meaning most of the Cuprate and Oxygen based superconductors are falling either around 25K or 85K. But this was very contrast to the Fe-based superconductors, where we observed a single elliptic shape kde plot at around 15~18K. Meaning, out of 2339 Fe based superconductors that we analyzed, most of them are falling close to the 15~18K. Even though we had 81 features at the beginning, it was realized that the 23 features are enough to predict 98% accuracy, see Figure 10. In this experiment we decided to exclude the application of outliers detection, because nothing can be treated as outliers or more precisely all these data are collected from the literatures, after many experimental verifications.

The experiment was tested out with several machine learning models before deciding which one to choose for this specific dataset. We have tried KNN, SVM, Multilayer Perceptron (MLP) as well. None of them performed well, including MLP. Since MLP took more than 6hrs to run a single iteration (without PCA application), we decided to terminate it, yet the performance was not good. Performance of KNN was better compare to SVM and MLP, it obtained RMSE of 13.89 for Oxygen-based, 14.44 for Cupper-based, and 9.52 for Iron-based. The SVM has performed very poorly in our case, although O. Owolabi et al[2] reported it worked pretty well for the Fe-based superconducors. It is worth noting here that O. Owolabi et al tested only 30 superconductors, however we had tested 2339 Fe-based superconductors. Furthermore, we tested by changing the *regression* problem into *classification*. Hence the dataset was divided into three classes based on the critical temperature, which were then denoted by either class 0, 1 or 2 and tested for the XGBoost. The performance of such model was very poor in quality.

Moreover, we wanted to test, is there any significant improvement if we include the number of Cu atoms/O atoms/Fe atoms as features. This was not included in the work of K.Hamidieh et al[1]. Hence we tried this test with the XGBoost/FR for cuprate. We did not find any significant improvement in the performance of the models. We obtained 13.54 RMSE value for the XGBoost and 13.59 for RF. To run such model, Cu must be simply eleminated from the feature space if we run for cuprate for example.

As a future work, it is recommended to analyze the data that include more features. For example, at the beginning we wanted to include the pressure data into the existing dataset, but unfortunately NIMS database has been closed for the maintenance purposes. Moreover if we could include electronic band structure or Fermi surface of iron-based superconductors, it could give more insight into the problem.

References

1. Kam Hamidieh, A data-driven statistical model for predicting the critical temperature of a superconductor, ELSEVIER, Computational Materials Science, 154 (2018) 346–354

2. O. Owolabi, O.Akande, O.Olatunji, Prediction of Superconducting Transition Temperatures for Fe- Based Superconductors using Support Vector Machine, ISSN 2225-0638, Vol.35, 2014.

3. D. D. Landis, J. Hummelshøj, S. Nestorov, J. Greeley, M. Du lak, T. Bligaard, J. K. Nørskov, and K. W. Ja- cobsen, The Computational Materials Repository, Comput. Sci. Eng. 14, 51–57 (2012).

4. T. Chen, C. Guestrin, XGBoost: A Scalable Tree Boosting System, arXiv:1603.02754 (2016)

Additional package installed:
Install XGBOOST in the Anaconda envo using: conda install -c conda-forge xgboost