Prediction of Critical Temperature of Superconductors using Tree Based Methods with K-Fold Cross Validation

Peeyush Kumar Singh[1], Pratham Kumar[2]

[1]*Student, Department of Computer Science and Engineering, HMR Institute of Technology and Management, Delhi, India*
[2]*Student, Department of Information Technology, College of Engineering Roorkee, Uttarakhand, India*

---***---

Abstract – *Superconductors are materials that show the phenomenon of superconductivity which is a state of matter that conducts current with zero resistance. Hence, an electric current can exist indefinitely in such a material. Critical temperature is that temperature below which they show superconductivity (their resistance drops to zero). These materials have significant practical applications but it is very difficult to explain the critical temperature. For this, empirical models are used by researchers in synthesizing such materials. In this paper, various tree-based methods namely Random Forest, CatBoost, LightGBM and XGBoost are employed with K-fold cross validation to achieve a prediction of critical temperature as close as possible to the true value. The findings demonstrate the effectiveness of these models based on root-mean-squared-error. Random Forest, CatBoost, LightGBM and XGBoost achieved 9.05 K, 8.95 K, 8.86 K and 8.85 K of root-mean-squared error which suggests that they can be used to predict the critical temperature by greatly reducing the range of temperatures in which it may lie. It is noted that the model is only valid for giving predictions for superconductors.*

***Key Words*: Superconductors, Critical Temperature, Random Forest, CatBoost, LightGBM, XGBoost, K-fold Cross-Validation, Machine Learning**

1.INTRODUCTION

First discovered by Dutch physicist Heike Kamerlingh Onnes in 1911, the phenomenon of superconductivity that is the existence of materials that conduct current with zero resistance when they become colder than their critical temperature, has found many practical applications in the modern world. They are widely used in particle accelerators, electric motors and generators, Maglev trains, Magnetic Resonance Imaging (MRI), RF and Microwave filters, etc. Thus, such materials are at the core of many modern-day technologies and are of great importance.

The Critical temperature, below which superconductivity is observed for a given superconductor is an important factor in determining the practicality of a superconductor but it is very hard to determine. Due to this, researchers employ empirical models to determine the critical temperature. With the advancement in Machine learning techniques, it has now become possible to considerably narrow down the range of temperatures among which critical temperature may lie. Tree based architectures have emerged as a convincing technique for this purpose as they can predict the Critical temperature with an impressive root-mean-squared-error.

1.1 Random Forest

A Random Forest is, at its core, a supervised learning technique that is an ensemble of Decision trees. It was developed by Leo Breiman and Adele Cutler. It is a meta-estimator that aggregates many Decision trees. Each tree draws a random sample from original dataset when generating a split and uses averaging to improve accuracy. The trees run in parallel therefore, there is no interaction among them. Each tree is different as not all features are considered while making an individual tree. It is immune to the curse of dimensionality. Its ease of use and flexibility has seen it being adopted widely. It can handle both regression and classification tasks. A regression technique that utilizes Random Forest is available in Scikit-Learn library which is used here.

1.2 CatBoost

CatBoost, developed by Yandex (2017) is an open-source library for gradient boosting on decision trees. The name comes from two words “Category” and “Boosting”. The idea of boosting is to sequentially combine many weak models and thus through greedy search strategy, create a strong model. CatBoost builds symmetric trees and supports all kinds of features be it numerical, categorical or text. Its performance, robustness and ease of use makes it a good choice for regression and classification tasks although it is still used relatively less than the well-known XGBoost method.

1.3 LightGBM

Light Gradient Boosting Machine, otherwise known as LightGBM developed by Microsoft (2016) is an open-source library for gradient boosting on decision trees. The difference between this and other boosting frameworks is that it grows tree leaf-wise while other methods do it level-wise. It is termed ‘Light’ due to the speed of its execution. It can handle large size of data and takes much lower memory to run compared to other methods. It uses two novel techniques namely Gradient-based One Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) and fulfils the limitations of the histogram-based algorithms.

1.3 XGBoost

XGBoost is also an ensemble learning method which utilizes decision trees and implements machine learning algorithms under the Gradient boosting framework. It stands for Extreme Gradient Boosting. It was developed by Chen and Guestrin (2016). The trees are created in sequential form in XGBoost. All the independent variables are assigned weights and fed to a decision tree. The weight of variables wrongly predicted are increased and fed to second decision tree. Due to this, a strong and precise model is built. The xgboost library offers both the classification and regression techniques which is used here. It gives reasonable error of 8.853 K based on root-mean-squared-error.

2. LITERATURE REVIEW

[1] The paper discusses a XGBoost based model to predict the critical temperature of a superconductor. The paper presents a best performing model which has a cross validated root-mean-squared-error of 9.5K. The performance metric was 25-fold cross validated root-mean-squared-error. The paper also covered feature importance and concluded that properties such as thermal conductivity, valence, electron affinity and atomic mass could be more important in predicting critical temperature.

3. METHODOLOGY

The proposed approach consists of following stages: Data collection and preprocessing, Model creation & Randomized Search for hyperparameters, cross validation and score tabulation.

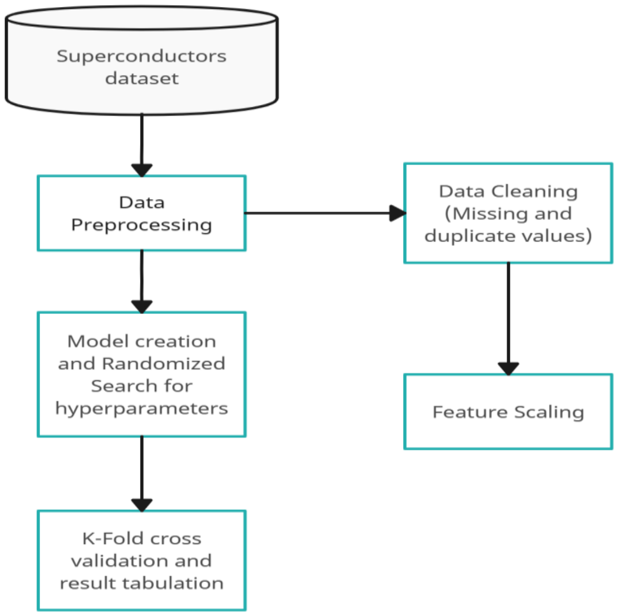
Superconductors dataset
Data Preprocessing
Data Cleaning (Missing and duplicate values)
Model creation and Randomized Search for hyperparameters
Feature Scaling
K-Fold cross validation and result tabulation

Fig- 1: Workflow Diagram

(I) *Data Collection and Preprocessing*

To conduct this research, a dataset comprising of the superconductors and their relevant features was obtained from the UCI Machine Learning Repository. The dataset consists of 21263 instances of superconductors and 82 features with 82nd feature being the critical temperature. A correlation map was built as shown in the figure below. The dataset was then split into X set containing all the features and Y set containing the critical temperature. The feature set X was then scaled using standard scaler of the Scikit-Learn library.

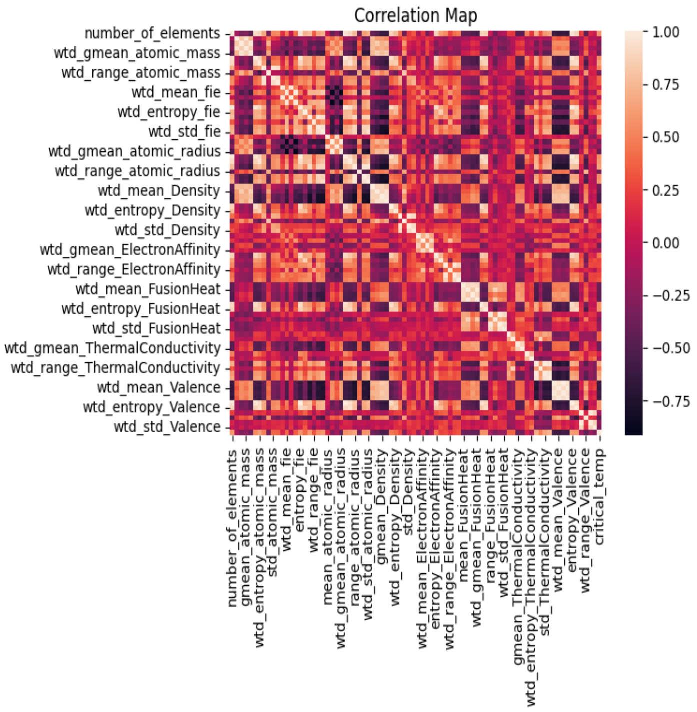

Fig- 2: Correlation Map

(II) *Model creation & Randomized search for hyperparameters*

Each model along with its grid of hyperparameters was built. Experimentation with various such grids was done before choosing the final grid. A Randomized Search was performed on the final grid to get the values of hyperparameters. In Randomized search, not all parameter values are tried out. Only a fixed number of settings is sampled from the distribution. These hyperparameters are then used to build the final model.

(III) *K-Fold cross validation and result tabulation*

Cross validation is a technique to estimate the performance of the model on unseen data. The dataset is split into K folds. In each iteration, data from K-1 folds is used to estimate the model and evaluated on the remaining fold. This is repeated K times so that each fold is used once for validation and an average evaluation metric is computed.

The final models are cross validated using 3-Fold, 5-Fold and 10-Fold cross validation using the KFold and cross_val_score methods of Scikit-Learn library. The results are then tabulated. In the evaluation of the models, the metric used is RMSE (root-mean-squared error) which is defined as:

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

Fig- 3: RMSE Formula

4. EVALUATION & RESULTS

(a) First, a Random Forest Model was created using the Random Forest Regressor using the Scikit-Learn library. A randomized search was performed on its hyperparameters and thus the model was built using following: n-estimators=100, max_depth=50 and max_features=0.3. After this, 3-fold, 5-fold and 10-fold cross validation was done and the results are tabulated below. These scores are impressive and already outperforms the score mentioned in [1].

CV	RMSE
3-fold	9.484991 K
5-fold	9.210530 K
10-fold	9.050967 K

Table- 1: RMSE for Random Forest

This result was taken as a benchmark for our boosting algorithms.

(b) A CatBoost model was then created using the CatBoost Regressor. Performing the randomized search on its hyperparameters yielded the following: iterations=2000, depth=10, l2_leaf_reg=1 and lastly, learning_rate=0.05. These parameters were then used to perform 3-fold, 5-fold and 10-fold cross validation. The results are as follows:

CV	RMSE
3-fold	9.369437 K
5-fold	9.103698 K
10-fold	8.959809 K

Table- 2: RMSE for CatBoost

(c) Afterwards, a LightGBM model was created using the LightGBM Regressor. Randomized search on its hyperparametersgave the following: n_estimators=750, num_leaves=100, min_split_gain=0.01, learning_rate=0.05, colsample_bytree=0.5. These parameters were then used to perform 3-fold, 5-fold and 10-fold cross validation. The results are as follows:

CV	RMSE
3-fold	9.342223 K
5-fold	9.051495 K
10-fold	8.864332 K

Table- 3: RMSE for LightGBM

(d) Lastly, XGBoost model was created using the XGBoost Regressor. Randomized search on its hyperparameters yielded the following: n_estimators=750, max_depth=20, learning_rate=0.01,colsample_bytree=0.3,alpha=2, lambda=0, gamma=2 and min_child_weight=10. These parameters were then used to perform 3-fold, 5-fold and 10-fold cross validation. The results are as follows:

CV	RMSE
3-fold	9.294267 K
5-fold	9.020392 K
10-fold	8.853131 K

Table- 4: RMSE for XGBoost

Finally, taking the best score of all models which is at 10-Fold, we get the following result:

Model	RMSE (10-Fold)
Random Forest	9.050967 K
CatBoost	8.959809 K
LightGBM	8.864332 K
XGBoost	8.853131 K

Table- 5: RMSE (10-Fold) for Various Tree-based models

This shows that all four models outperform the one mentioned in [1]. Furthermore, among all these XGBoost performs best although it is closely followed by LightGBM and CatBoost. It is to be noted that the LightGBM model is among the fastest gradient boosting methods and its result being on par with the XGBoost model (within very small margin of error) makes it a favorable model to consider for real world usage. It is evident that all of these models perform well for predicting the critical temperature of the superconductors.

5. CONCLUSION

It is shown that the proposed models can predict the critical temperatures of superconductors reasonably well. These can greatly reduce the range of temperature within which it may lie. Future studies may concentrate on dataset expansion and rigorous feature engineering with extensive search for more optimal hyperparameters. The models can be easily used for research purposes.

REFERENCES

[1] Hamidieh, K. (2018). A data-driven statistical model for predicting the critical temperature of a superconductor. *Computational Materials Science*, *154*, 346-354.

[2] Hamidieh,Kam. (2018). Superconductivty Data. UCI Machine Learning Repository. https://doi.org/10.24432/C53P47.

[3] https://xgboost.readthedocs.io/en/stable/

[4] https://catboost.ai/

[5] https://lightgbm.readthedocs.io/en/latest/index.html