# Predicting Superconductivity and Critical Temperature with Chemical Composition and Machine Learning

Final Project Report

**SuperconducTeam:**

Alex Oranday, Ben Harris, Kevin Allen, Priyansh Lunia

May 12, 2021

# Contents

**1 Introduction 1**

1.1 History of Superconductivity Theory . . . 2

1.2 The Many-body Hamiltonian . . . 4

1.3 Data Science Objectives and Core Science Question . . . 5

1.4 Report Breakdown . . . 6

**2 Data Science Pipeline 7**

**3 Data Acquisition & Wrangling 11**

3.1 SuperCon & Materials Project . . . 11

3.2 Chemical Composition Matrix . . . 12

3.3 Training, Validation, & Testing Sets . . . 13

**4 Classification 14**

**5 $T_c$ Prediction 17**

5.1 Results . . . 17

**6 Discussion and Conclusion 22**

**A Common Terms 26**

**B Replication of Roter and Dordevic 27**

B.1 Replication of Roter and Dordevic's Classification Results . . . 27

B.2 Replication of Roter and Dordevic's Regression Results . . . 28

# 1 Introduction

Superconductivity has been in the spotlight of condensed matter research since its discovery by Kamerlingh Onnes in 1911 [5]. He observed that the electrical resistance of metals such as mercury, lead, and tin would disappear below a certain material-dependent threshold called the critical temperature, $T_c$. It is this property of vanishing electrical resistance that is known as superconductivity.

The prospect of high-temperature superconductors (HTS), has immense potential for technical applications — HTS could replace all conventional conductors as they have zero electrical resistance. Some applications include energy storage, efficient energy transmission, transportation, and use in particle accelerators and nuclear fusion reactors for magnetic confinement.

Despite over a century of research, it remains a challenge both theoretically and experimentally to understand the connection between superconductivity and a material's chemistry and structure [12]. This complexity comes from the fact that existing theories of superconductivity can only describe a portion of existing superconductors and do not work for high-temperature superconductors. It is possible to take advantage of the abundance of discovered superconductors and the accessibility of materials databases and apply machine learning methods to gain a better understanding of the correlations that tie superconductivity to the material's chemistry and structure. In this work we take advantage of open databases to apply machine learning methods for classifying superconductors and predicting their critical temperature.

To this aim we follow the method proposed by Roter and Dordevic in their 2020 paper which proposed a simple method for superconductor classification and critical temperature prediction. They claim that their models' performance was on par with the most accurate models proposed in the field, using only the chemical composition of the materials as features [8]. For this project, we seek to build upon their paper, and provide further insight into the role that chemical composition plays in superconductivity.

## 1.1 History of Superconductivity Theory

Superconductivity has come a long way since its discovery at the start of the 20th century. According to Schrieffer, an important author in the theory that described superconductivity, this phenomenon can be initially described as electrons condensing into a macro-molecule that is capable of motion as a whole as a super-fluid; as the temperature nears $T_c$, a second order phase transitions occurs from a normal state (or normal fluid) to a super-fluid associated with a rigidity, with respect to a perturbation such as a magnetic field [10]. To fully understand how the theory for superconductivity was reached, it is important to look at the history behind the description of this phenomenon.

Historically the next breakthrough after the discovery of superconductivity occurred in 1933 by Meissner and Ochsenfel with the discovery of the perfect diamagnetism of bulk superconductors in weak magnetic fields — this is known as the Meissner effect. This effect indicates that a magnetic field is excluded from entering a superconductor and any field within a non-superconducting sample will be expelled once temperature drops through $T_c$, implying that superconductivity will be destroyed by a critical magnetic field $H_c$. This field is related to the free energy difference between the normal and superconducting states at zero field [13].

Returning to the idea of the rigidity of a superfluid, the London brothers proposed a phenomenological theory related to the electromagnetic behavior of superconductors. This was based on the two-fluid type concept mentioned by Schrieffer that have two different fluid densities and velocities. They also introduced a parameter related to the penetration of fields within the surface of a superconductor called the London penetration depth [4]. Parameters such as the coherence length were obtained from proposing a non-local generalization of the London equation — this is analogous to the mean free path $\ell$ in non-local electrodynamics of normal metals [13].

An important advancement towards understanding superconductivity was Landau's contribution to the theory of phase transitions, which describes symmetry breaking at low temperature phase transitions. This work provided inspiration for another theory, the Ginsburg-Landau theory [4]. This focuses on the superconducting electrons, rather than excitations, and considers that the superfluid density may vary in space [10].

It wasn't until the 1950s that a satisfactory theory was constructed to describe conventional superconductors. The first aspect to understand of this theory, is the existence of an energy gap of order $k_B T_c$ between the ground state and the quasi-particle excitations [13]. The presence of a band gap in a superconductor may be explained by the idea of Cooper pairs. By focusing on the interaction between two electrons, Cooper showed that an arbitrarily small attraction between electrons costs less energy for the electrons to pair up rather than float in a Fermi sea. This interaction overcomes the Coulomb repulsion. The amount of energy needed to break the pairs is know as binding energy and this corresponds to the superconducting gap. The attraction between the electrons was associated with an electron-phonon interaction, and was later described in what is know as BCS theory by Bardeen, Cooper and Schrieffer [4, 13].

BCS theory explains how the positions of the atomic cores are modified by the electrons' electrostatic field and how phonons produce an attractive interaction between these electrons, given by a potential $V_0$. The pairs that form have opposite spin, thus they obey Bose-Einstein statistics as they form a Bose-type condensate under a critical temperature. As this interaction is mediated by phonons, the critical temperature scales with the Debye temperature $T_D$ instead of the Fermi temperature, $T_F$. BCS theory gives the equation:

$$T_c = \frac{2e^{\gamma_E}}{\pi} \frac{\hbar \omega_D}{k_B} e^{\frac{-1}{V_0 \rho_F}} \tag{1.1}$$

where $V_0 \rho_F \ll 1$, $\gamma_E$ is the Euler constant, $\omega_D$ is the Debye frequency, $V_0$ is the attractive interaction, and $\rho_F$ is the density of states. The density of states indicates the pairing is limited to the electrons that are near the Fermi surface [11].

## 1.2 The Many-body Hamiltonian

Most previous studies of superconductor classification and critical temperature prediction using machine learning have relied on assembling a dataset of material features, either calculated or measured experimentally [9]. These features include electronic, thermodynamic, and quantum-mechanical quantities of the material — features which would have some relevance to or correlation with superconductivity. At first glance, the chemical composition of a material as features for training does not appear to contain relevant information to be an accurate predictor. It is the purpose of this section to motivate why using the chemical composition is not only valid, but potentially a powerful predictor.

As described by Townsend, the Hamiltonian, $\hat{H}$, is the generator of time translations of a system in quantum mechanics. In other words, it governs how a quantum state evolves over time. Furthermore, the Hamiltonian is also the energy operator, corresponding to the total energy of the system. By solving the time-independent Schrödinger equation, the eigenstates and eigenenergies of the system can be computed. These eigenenergies have importance in quantum mechanics as they correspond to the spectrum of possible energies a system can have. [14]

The many-body Hamiltonian is the general Hamiltonian for a system with many interacting particles. It is straightforward as it is composed of the kinetic energies of all the particles in the system as well as the potential energies between each pair of interacting particles. A system's many-body Hamiltonian, which is composed of $N_e$ electrons and $N_n$ nuclei is given by:

$$\begin{aligned}\hat{H} &= \hat{T}_e + \hat{T}_n + \hat{V}_{ne} + \hat{V}_{ee} + \hat{V}_{nn} \\ &= -\frac{\hbar}{2}\left(\sum_i^{N_e} \frac{1}{m_e}\nabla_i^2 + \sum_I^{N_n} \frac{1}{m_n}\nabla_I^2\right) \\ &\quad + \frac{e^2}{4\pi\epsilon_0}\left[-\sum_i^{N_e}\sum_I^{N_n} \frac{Z_I}{|\vec{r}_i - \vec{R}_I|} + \frac{1}{2}\sum_i^{N_e}\sum_{j\neq i}^{N_e} \frac{1}{|\vec{r}_i - \vec{r}_j|} + \sum_I^{N_n}\sum_{J\neq I}^{N_n} \frac{Z_I Z_J}{|\vec{R}_I - \vec{R}_J|}\right]\end{aligned} \tag{1.2}$$

where $\hat{T}_e$ and $\hat{T}_n$ are the kinetic energies of the electrons and nuclei, respectively; $\hat{V}_{ne}$, $\hat{V}_{ee}$, and $\hat{V}_{nn}$ are the potential energies of the electron-nuclei interactions, electron-electron interactions, and nuclei-nuclei interactions, respectively. The indices $i, j$ run over the electrons, and the indices $I, J$ run over the nuclei. Although eigenstates and eigenenergies of the many-body Hamiltonian are impossible to solve analytically, it is in principle possible to derive every chemical property of the material using the Hamiltonian. Each of the five terms of Eq. 1.2 can be inferred from the chemical composition of the material (in addition to knowledge of the structure of the material). [3]

Thus, by using only the chemical composition as training features, we are essentially 'encoding' all the chemical properties of relevance of the material. Although this 'encoding' is not explicit, and the Schrödinger equation is not solvable analytically, machine learning is a powerful tool for learning patterns, correlations, and connections between chemical composition and superconducting properties.

## 1.3 Data Science Objectives and Core Science Question

Since we are concerned with investigating the connections between chemical composition and superconductivity, we choose to explicitly define two problems that we are going to tackle using machine learning. The first problem is the classification problem, which is concerned with determining if a given material is a superconductor or non-superconductor. The second problem is the regression problem, which involves predicting the critical temperature of a superconductor. For both of these problems, the only features that will be used are the material's chemical composition. These two problems, which are subsequently referred to as classification and $T_c$ prediction, form the basis of this project. Having explicitly defined the two problems that we are tackling in this work, we can now form our two central data science objectives:

1. Train a **classifier** to determine whether or not a proposed material is a superconductor or not.

2. Train a **regressor** to predict the critical temperature of a superconductor.

Pursuing these data science objectives are designed to help answer our core science question, which is: why does it make *physical* sense for chemical composition to be so effective in predicting superconductivity and critical temperature?

## 1.4 Report Breakdown

The subject of this report is to present on the machine learning of chemical composition to predict superconductivity in materials and predict the $T_c$ of superconductors. The following chapter outlines the data science pipeline we followed. Section 3 goes through our data acquisition and wrangling step, specifically explaining how we form our datasets for training. The subsequent two sections, 4 and 5, report on the results for classification and $T_c$ prediction, respectively. Note that the details of the machine learning models used for both problems are explained in Appendix A. Finally, Section 6 provides some concluding and evaluative remarks as well as suggestions for future studies.

As stated earlier, this work builds upon [8]. Although their results are encouraging, upon closer inspection of their methods, we found an abundance of problematic machine learning practices. In an effort to illustrate the importance of proper machine learning practice, we replicated their method and discuss its potential issues further in Appendix B.

# 2 Data Science Pipeline

The data science pipeline we followed is fully outlined in the table below.

| | Main Objective | Sub-objective |
|---|---|---|
| **Data Wrangling** | **Goals** | |
| | 1. Create the chemical composition matrix.<br>2. Create training, validation, and testing sets with random entries. | 1. Normalize the rows of this matrix to sum to 1. |
| | **Steps** | |
| | 1. Collect data from both SuperCon and Materials Project.<br>2. Process formulae and find differing entries between the two data sets.<br>3. Process the formulae to match the form of the chemical composition matrix.<br>4. Split into training, validation, and testing data. | |

|  | Main Objective | Sub-objective |
|---|---|---|
| **Exploratory Data Analysis** | **Goals** |  |
|  | 1. Explore the abundance of each element in each compound, as well as the whole dataset.<br>2. Explore the sparsity of the chemical composition matrix. |  |
|  | **Techniques** |  |
|  | 1. Count quantities for each element. |  |
| **Modeling** | **Goals** |  |
|  | 1. Explore $k$-NN models for classifying superconductors.<br>2. Explore Bagged Trees models for predicting a superconductor's critical temperature. | 1. Explore Bagged Trees models for classifying superconductors.<br>2. Explore Boosted Trees and Random Forest models. |

| | Main Objective | Sub-objective |
|---|---|---|
| | **Techniques** | |
| | 1. Produce a competitive $k$-NN classification model.<br>2. Produce a competitive Bagged Trees regression model.<br>3. Explore various $k$ values for $k$-NN models.<br>4. Explore different quantities of decision trees for Bagged Trees models. | 1. Produce competitive models for Bagged Trees, Boosted Trees, and Random Forests. |
| **Validation** | **Goals** | |
| | 1. Maximize classification accuracy to at least 90%.<br>2. Minimize root-mean-square error (RMSE) for predicted critical temperatures. | 1. Limit the use of the testing set to the end of the study. |

|  | **Main Objective** | **Sub-objective** |
|---|---|---|
|  | **Techniques** |  |
|  | 1. Train/Validation/Test Split. |  |
| **Communication** | **Reporting** |  |
|  | 1. Performance of the best model.<br>2. Interesting findings as a result of each applied model. |  |

# 3 Data Acquisition & Wrangling

## 3.1 SuperCon & Materials Project

The SuperCon database contains the most extensive and up-to-date collection of known superconductors with reference to journal entries [2]. Many machine learning projects involving superconductivity rely heavily on information made available from the SuperCon database, and their free-to-use data was the baseline of our project [8, 9, 12].

From the SuperCon database, we retrieved the chemical formulae of all the available superconductors and their corresponding critical temperatures. At the time of download, we obtained 33,248 superconductor entries. However, 7,059 entries did not have a listed $T_c$, 5 entries had a $T_c$ greater than 273 K, and 5,612 other entries that contained errors in their chemical formulae. These errors could take the form of invalid element symbols, such as Z, L, or Ox.

We aimed to use the chemical compositions as our predictors, and thus we were required to remove all 5,612 entries that contained chemical formulae errors. We also planned to use the same dataset of superconductors for our classification and regression problems. For this reason we removed the 7,064 entries with a high or missing $T_c$. After removing these entries, we were left with 20,572 superconductors.

For our non-superconductors, we collected chemical formulae from the Materials Project. The Materials Project is an open web-based database of materials, material properties, and material analysis [7]. Their goal was to transition material studies into the computational and information age. More information on the Materials Project can be viewed in [7]. Since the purpose of the SuperCon database is to be the most up-to-date with regard to known superconductors, we assumed that any material present in the Materials Project database but not in the SuperCon database is a non-superconductor [2]. We acquired 20,572 non-superconductors to match the amount of superconductors we gathered and created a balanced dataset. For each entry in our database, we assigned a label of ‘0’ or ‘1’ to denote classes, where ‘0’ indicates a non-superconductor and a ‘1’ indicates a superconductor.

Thus, we have created a database for both the classification problem and the critical temperature prediction problem. The dataset for the classification problem has 41,144 chemical formulae, 50% non-superconductors and 50% superconductors, with the 0/1 labeling. The dataset for the critical temperature prediction problem uses 20,572 superconductor formulae with a matching $T_c$ for every entry. Now, we can create our so-called chemical composition matrix, which is described in the next section.

## 3.2 Chemical Composition Matrix

Having obtained the chemical formulae, we created the chemical composition matrix, which is shown in Table 3.1. This matrix involves splitting the elements present in each material and assigning the index of each element as a column. Each row of this matrix is each chemical formulae entry from our datasets. For consistency, we normalize each row by dividing by its corresponding row sum such that molar ratios are preserved and values in each row sums to 1.

| **Formula** | **Ba** | **La** | **Cu** | **O** | **...** | **Ne** | **Label** |
|---|---|---|---|---|---|---|---|
| La1.85Ba0.15Cu1O4 | 0.021 429 | 0.264 286 | 0.142 857 | 0.571 429 | ... | 0.0 | 1 |
| La1.85Sr0.15Cu1O4 | 0.0 | 0.264 286 | 0.142 857 | 0.571 429 | ... | 0.0 | 1 |
| Lu1.8Ba0.2Cu1O4 | 0.028 571 | 0.0 | 0.142 857 | 0.571 429 | ... | 0.0 | 1 |
| Y0.9Ba0.1Cu1O4 | 0.016 667 | 0.0 | 0.166 667 | 0.666 667 | ... | 0.0 | 1 |
| Ba0.15La1.85Cu1O4 | 0.021 429 | 0.264 286 | 0.142 857 | 0.571 429 | ... | 0.0 | 1 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| Li5Mn7O16 | 0.0 | 0.0 | 0.0 | 0.571 429 | ... | 0.0 | 0 |
| Mn2BeO4 | 0.0 | 0.0 | 0.0 | 0.571 429 | ... | 0.0 | 0 |
| U2Cr3Si | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0 |
| BeSe | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0 |
| K15Cr7N19 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0 |

**Table 3.1:** An overview of the normalized chemical composition dataset, for the classification problem. ‘Label’ indicates whether the material is a superconductor (1) or non-superconductor (0). For the $T_c$ prediction problem, the ‘Label’ column is replaced by a column containing ‘$T_c$’.

The models we used take a row from this matrix as input and would output a 0 or 1 in the case of classification, or a predicted critical temperature in the case of regression.

## 3.3 Training, Validation, & Testing Sets

We split our datasets by 70% for training, 15% for validation, and 15% for testing. The exact number of entries for the classification and regression problem are shown in Table 3.2. Before the split, the datasets were shuffled randomly so that the entries present in each category after the split is near evenly distributed. The extra 2 entries in the testing sets compared to the validation sets was a design choice to deal with rounding errors.

| | **Training** (70%) | **Validation** (15%) | **Testing** (15%) |
|---|---|---|---|
| **Classification** | 28,800 | 6,171 | 6,173 |
| **Regression** | 14,400 | 3,085 | 3,087 |

**Table 3.2:** The number of entries in the training, validation, and testing sets for both the classification problem and the regression problem.

# 4 Classification

Our proposed classification technique differs from that proposed by Roter and Dordevic in a few key areas. First, we normalized our chemical composition matrices in such a way that each row would sum to one. We did so to maintain the integrity of elemental ratios and to not give an advantage to high number elements. Furthermore, we chose to forgo PCA as we saw no physical motivation for the action.

Roter and Dordevic built $k$-NN, Bagged Trees, Boosted Trees, and Support Vector Machine models for classification. Their most accurate model was $k$-NN at a reported 96.5% [8]. We also implemented $k$-NN, Bagged Trees, and Boosted Trees, but opted instead to use Random Forest as our fourth and final model. See Appendix A for details regarding the machine learning methods mentioned here. After completing training and optimizing on the validation set, we received the accuracy report as shown in Table 4.1 when classifying our test data.

At first glance, our results appear to mimic Roter and Dordevic. Like them we found every model to have $> 92\%$ accuracy. However, this greatly surprised us as we had previously attributed much of their accuracy to problematic methodology, as shown in Appendix B. Specifically, we were especially surprised by the $k$-NN results. Compared to the other three models, $k$-NN is incredibly simple, especially in this case where $k$=1. It looks at the nearest neighbor in Euclidean space and achieves a 93% accuracy. This indicates a stronger correlation than we anticipated.

Our best model was the Random Forest classifier which matched Roter and Dordevic's reported 96% accuracy [8]. Additionally, given the ensemble nature of Random Forest algorithms, we were able to compile the feature importance for each element in the chemical composition matrix space. This was done by taking an unweighted mean feature importance across the respective decision trees of the Random Forest model.

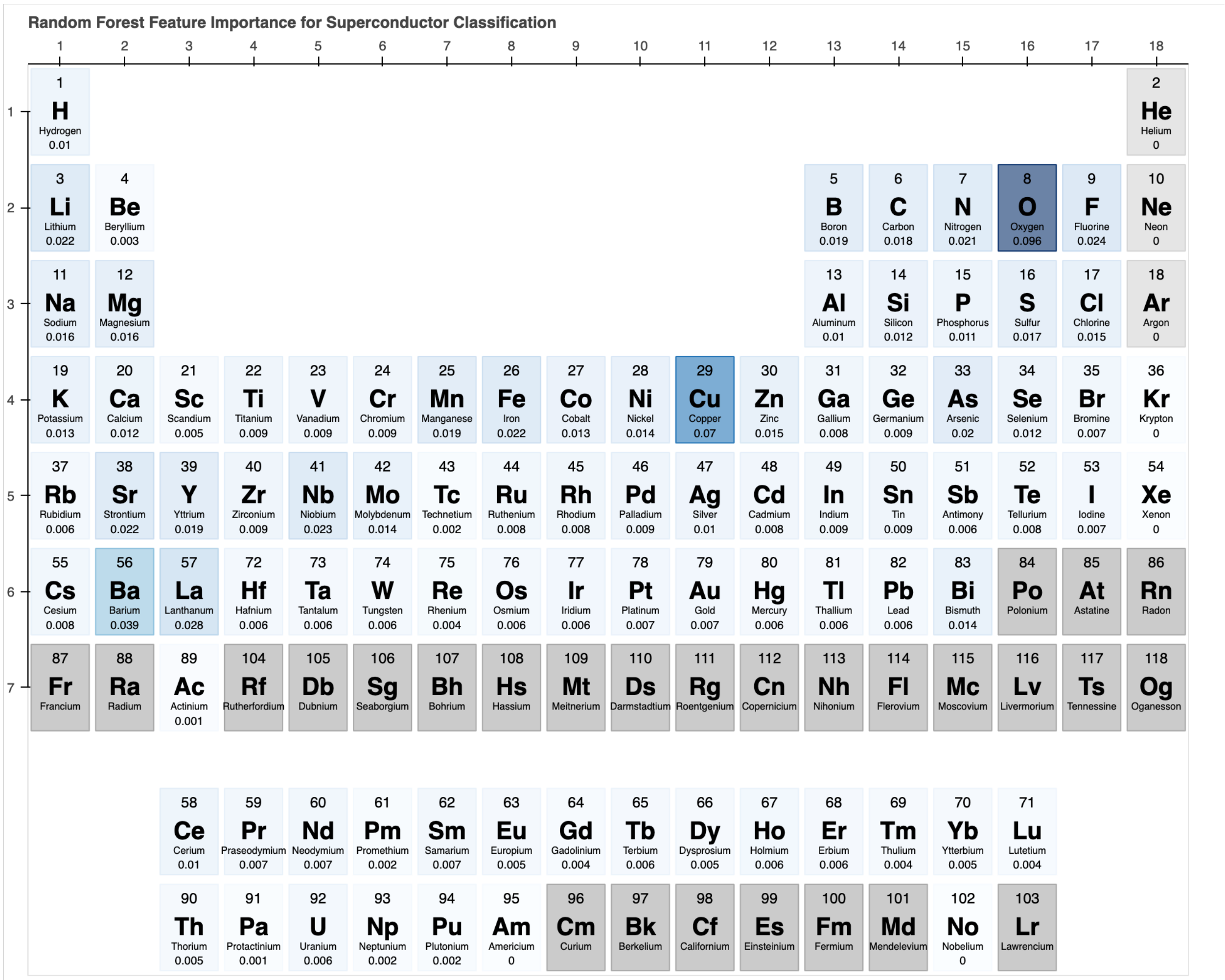
The five most important features for classification were Oxygen (9.636%), Copper (7.008%), Barium (3.940%), Lanthanum (2.772%), and Flourine (2.412%), as shown in Fig. 4.1.

| **Model** | **Class** | **Precision** | **Recall** | **F1-Score** |
|---|---|---|---|---|
| Random Forest | 0 | 0.97 | 0.95 | 0.96 |
| | 1 | 0.95 | 0.97 | 0.96 |
| | Accuracy | | | 0.96 |
| Boosted Trees | 0 | 0.95 | 0.94 | 0.94 |
| | 1 | 0.94 | 0.95 | 0.94 |
| | Accuracy | | | 0.94 |
| Bagged Trees | 0 | 0.97 | 0.92 | 0.95 |
| | 1 | 0.93 | 0.97 | 0.95 |
| | Accuracy | | | 0.95 |
| $k$-NN ($k$=1) | 0 | 0.98 | 0.88 | 0.92 |
| | 1 | 0.89 | 0.98 | 0.93 |
| | Accuracy | | | 0.93 |

**Table 4.1:** Validation statistics by model with improved chemical composition classification. Validation data was split from the balanced dataset.

While these results are interesting, they were expected as Oxygen and Copper have been linked to superconductivity [10]. However, we were surprised by the lack of noble gas representation. Helium, Argon, and Neon had feature importances of zero while Xenon and Krypton had importances of $1.917 \times 10^{-7}\%$ and $1.917 \times 10^{-8}\%$ respectively. Neither Radon nor Oganesson were present in the training chemical composition matrix inputs and thus were irrelevant regarding feature importances. We know that no superconductors contain noble gases so we expected them to have a high feature importance in classification, as the models would be filtering out compounds with noble gases present. However, given this is not the case, we theorized that our models are finding patterns within ratio analysis and not labeling based on a process of elimination.

To test this hypothesis, we implemented a binary chemical composition matrix. Rather than including a normalized ratio of chemical composition we placed a 1 for elements present in the compound and a 0 for those not present. When classification models were trained on



Random Forest Feature Importance for Superconductor Classification

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1<br>H<br>Hydrogen<br>0.01 | | | | | | | | | | | | | | | | | 2<br>He<br>Helium<br>0 |
| 2 | 3<br>Li<br>Lithium<br>0.022 | 4<br>Be<br>Beryllium<br>0.003 | | | | | | | | | | | 5<br>B<br>Boron<br>0.019 | 6<br>C<br>Carbon<br>0.018 | 7<br>N<br>Nitrogen<br>0.021 | 8<br>O<br>Oxygen<br>0.096 | 9<br>F<br>Fluorine<br>0.024 | 10<br>Ne<br>Neon<br>0 |
| 3 | 11<br>Na<br>Sodium<br>0.016 | 12<br>Mg<br>Magnesium<br>0.016 | | | | | | | | | | | 13<br>Al<br>Aluminum<br>0.01 | 14<br>Si<br>Silicon<br>0.012 | 15<br>P<br>Phosphorus<br>0.011 | 16<br>S<br>Sulfur<br>0.017 | 17<br>Cl<br>Chlorine<br>0.015 | 18<br>Ar<br>Argon<br>0 |
| 4 | 19<br>K<br>Potassium<br>0.013 | 20<br>Ca<br>Calcium<br>0.012 | 21<br>Sc<br>Scandium<br>0.005 | 22<br>Ti<br>Titanium<br>0.009 | 23<br>V<br>Vanadium<br>0.009 | 24<br>Cr<br>Chromium<br>0.009 | 25<br>Mn<br>Manganese<br>0.019 | 26<br>Fe<br>Iron<br>0.022 | 27<br>Co<br>Cobalt<br>0.013 | 28<br>Ni<br>Nickel<br>0.014 | 29<br>Cu<br>Copper<br>0.07 | 30<br>Zn<br>Zinc<br>0.015 | 31<br>Ga<br>Gallium<br>0.008 | 32<br>Ge<br>Germanium<br>0.009 | 33<br>As<br>Arsenic<br>0.02 | 34<br>Se<br>Selenium<br>0.012 | 35<br>Br<br>Bromine<br>0.007 | 36<br>Kr<br>Krypton<br>0 |
| 5 | 37<br>Rb<br>Rubidium<br>0.006 | 38<br>Sr<br>Strontium<br>0.022 | 39<br>Y<br>Yttrium<br>0.019 | 40<br>Zr<br>Zirconium<br>0.009 | 41<br>Nb<br>Niobium<br>0.023 | 42<br>Mo<br>Molybdenum<br>0.014 | 43<br>Tc<br>Technetium<br>0.002 | 44<br>Ru<br>Ruthenium<br>0.008 | 45<br>Rh<br>Rhodium<br>0.008 | 46<br>Pd<br>Palladium<br>0.009 | 47<br>Ag<br>Silver<br>0.01 | 48<br>Cd<br>Cadmium<br>0.008 | 49<br>In<br>Indium<br>0.009 | 50<br>Sn<br>Tin<br>0.009 | 51<br>Sb<br>Antimony<br>0.006 | 52<br>Te<br>Tellurium<br>0.008 | 53<br>I<br>Iodine<br>0.007 | 54<br>Xe<br>Xenon<br>0 |
| 6 | 55<br>Cs<br>Cesium<br>0.008 | 56<br>Ba<br>Barium<br>0.039 | 57<br>La<br>Lanthanum<br>0.028 | 72<br>Hf<br>Hafnium<br>0.006 | 73<br>Ta<br>Tantalum<br>0.006 | 74<br>W<br>Tungsten<br>0.006 | 75<br>Re<br>Rhenium<br>0.004 | 76<br>Os<br>Osmium<br>0.006 | 77<br>Ir<br>Iridium<br>0.006 | 78<br>Pt<br>Platinum<br>0.007 | 79<br>Au<br>Gold<br>0.007 | 80<br>Hg<br>Mercury<br>0.006 | 81<br>Tl<br>Thallium<br>0.006 | 82<br>Pb<br>Lead<br>0.006 | 83<br>Bi<br>Bismuth<br>0.014 | 84<br>Po<br>Polonium | 85<br>At<br>Astatine | 86<br>Rn<br>Radon |
| 7 | 87<br>Fr<br>Francium | 88<br>Ra<br>Radium | 89<br>Ac<br>Actinium<br>0.001 | 104<br>Rf<br>Rutherfordium | 105<br>Db<br>Dubnium | 106<br>Sg<br>Seaborgium | 107<br>Bh<br>Bohrium | 108<br>Hs<br>Hassium | 109<br>Mt<br>Meitnerium | 110<br>Ds<br>Darmstadtium | 111<br>Rg<br>Roentgenium | 112<br>Cn<br>Copernicium | 113<br>Nh<br>Nihonium | 114<br>Fl<br>Flerovium | 115<br>Mc<br>Moscovium | 116<br>Lv<br>Livermorium | 117<br>Ts<br>Tennessine | 118<br>Og<br>Oganesson |
| | | | 58<br>Ce<br>Cerium<br>0.01 | 59<br>Pr<br>Praseodymium<br>0.007 | 60<br>Nd<br>Neodymium<br>0.007 | 61<br>Pm<br>Promethium<br>0.002 | 62<br>Sm<br>Samarium<br>0.007 | 63<br>Eu<br>Europium<br>0.005 | 64<br>Gd<br>Gadolinium<br>0.004 | 65<br>Tb<br>Terbium<br>0.006 | 66<br>Dy<br>Dysprosium<br>0.005 | 67<br>Ho<br>Holmium<br>0.006 | 68<br>Er<br>Erbium<br>0.006 | 69<br>Tm<br>Thulium<br>0.004 | 70<br>Yb<br>Ytterbium<br>0.005 | 71<br>Lu<br>Lutetium<br>0.004 | |
| | | | 90<br>Th<br>Thorium<br>0.005 | 91<br>Pa<br>Protactinium<br>0.001 | 92<br>U<br>Uranium<br>0.006 | 93<br>Np<br>Neptunium<br>0.002 | 94<br>Pu<br>Plutonium<br>0.002 | 95<br>Am<br>Americium<br>0 | 96<br>Cm<br>Curium | 97<br>Bk<br>Berkelium | 98<br>Cf<br>Californium | 99<br>Es<br>Einsteinium | 100<br>Fm<br>Fermium | 101<br>Md<br>Mendelevium | 102<br>No<br>Nobelium<br>0 | 103<br>Lr<br>Lawrencium | |

**Figure 4.1:** The table displays the feature importance of each element in our Random Forest classification model. Specific element feature importance values are shown at the bottom of the respective cell. Blue tones indicate the relative importance with darker tones indicating increased importance. Light grey tones indicate that the element received a feature importance of zero. Dark gray tones indicates the element was not found in the chemical composition matrix.

this, we saw significant decreases in accuracy. Additionally, we created a custom validation dataset which only included compounds where all constituent elements are also present in known superconductors. When testing on the custom shared validation set we saw no significant decrease in model accuracy. Both of these results support our hypothesis.

# 5 $T_c$ Prediction

We only explored the Bagged Trees algorithm for critical temperature prediction. We used the chemical composition matrix described in Section 3 as our input for training, validation, and testing.

## 5.1 Results

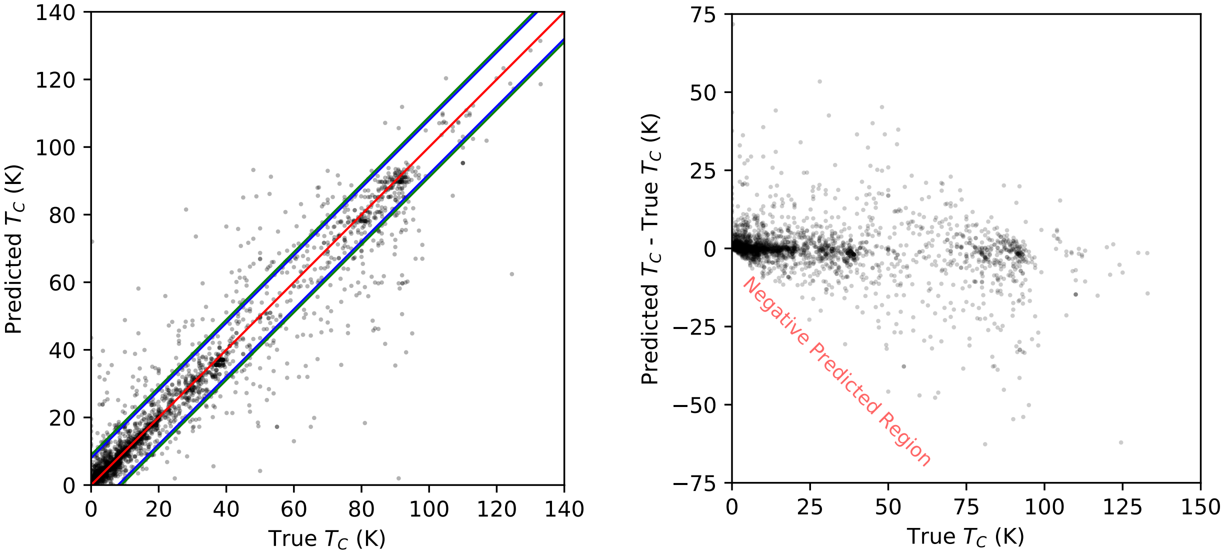
We define our resolution as the predicted critical temperature minus the true critical temperature:

$$\Delta T_c \equiv T_c^{\text{Predicted}} - T_c^{\text{True}}.$$

In the perfect case, for $\Delta T_c$ we would achieve a standard deviation of 0 K and all values would appear at 0 K. However, this is not something we would hope for or ever expect to achieve, for this would indicate that there is a mistake in the model.

The distribution we produced when recreating Roter and Dordevic's methods is precise with respect to $\Delta T_c$, but would be the result of overfitting [1]. The model was able to train to perform as best as possible for the given data, but has no indication of how it will perform when applied to new data. Following the use of a training, validation, and testing split, we expected a much more robust model at a higher RMSE and standard deviation [1]. A more detailed discussion on the recreated results are given in Appendix B.

We tested this hypothesis using our training, validation, and testing dataset where we achieved a RMSE of approximately 7.97 K on our testing set. As expected, this value is significantly greater than the 4.11 K we obtained by using all samples during training. This trend follows with the resolution. We found $\Delta T_C$ had a mean value of 0.1380 K and a standard deviation of 8.1891 K. The exact distribution, shown in Fig. 5.3a, has a similar shape to the overfitted results with predictable changes in terms of the width and height. One would expect a shorter height due to the decrease in samples from splitting and a wider distribution when applying a model to new samples.



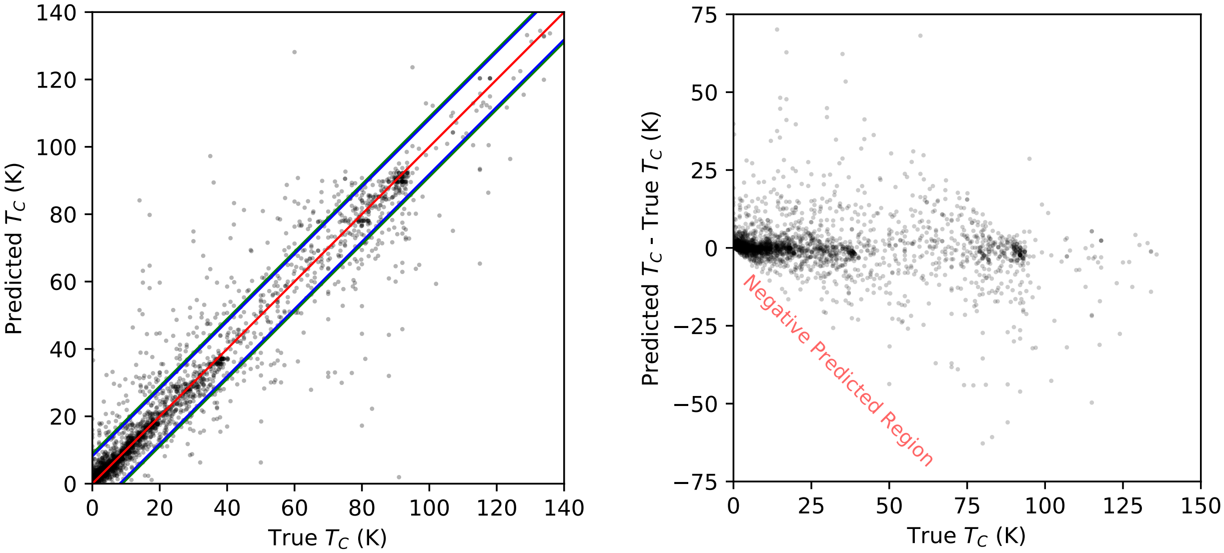
**(a)**

**(b)**

**Figure 5.1:** These plots are the result of our testing set. **(a)** Scatter plot of predicted critical temperature versus the true critical temperature. The red line is $y = x$ and only used for visualization. The green lines are shifted by ± 8.91 to represent the RMSE achieved by Roter and Dordevic [8]. The blue lines are shifted by ± 7.97 to represent the RMSE achieved by our model. **(b)** Scatter plot of Predicted $T_C$ - True $T_C$ versus True $T_C$. The red shaded region is the area where the predicted critical temperature is less than 0 K. Both plots make use of transparent points to better visualize density.

An important goal we considered is to not produce any negative predicted temperatures. In Fig. 5.2b, we show that our approach produced no negative temperatures. The red region in this plot indicates where negative predicted temperatures would occur. This is a notable result as it shows that our model has learned a physical barrier to potential temperatures.

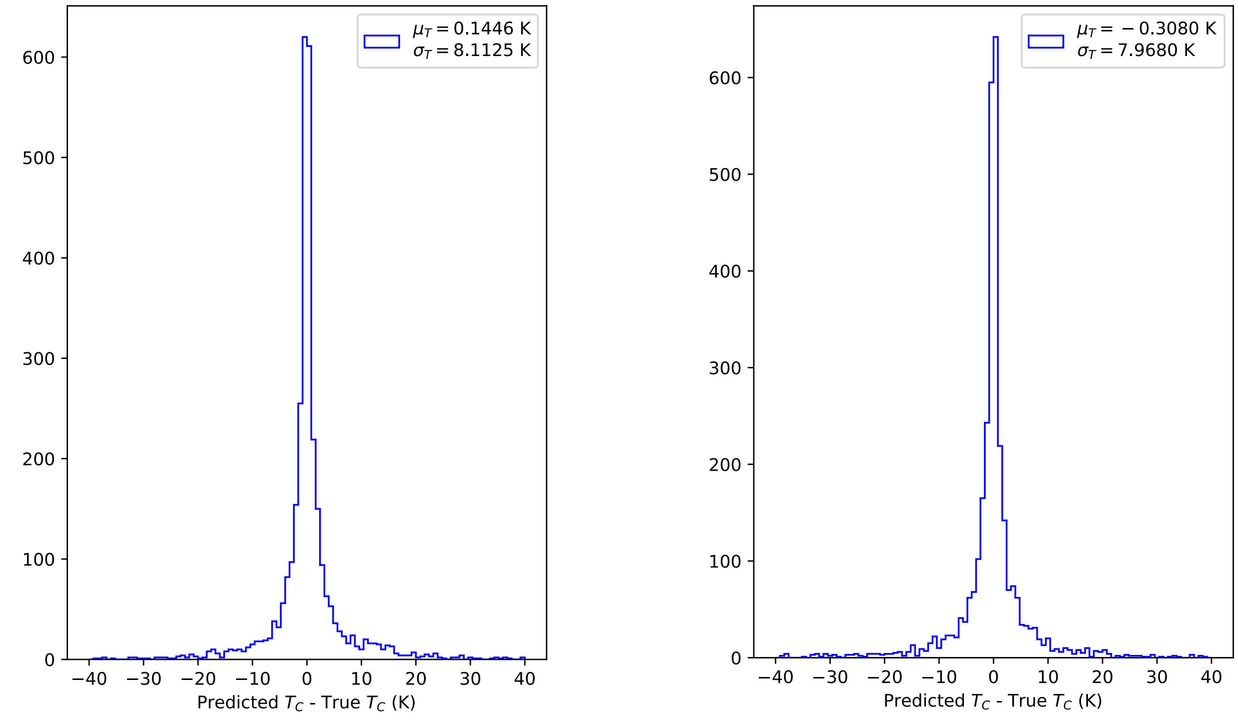
Another notable result is the feature importance of the elements when predicting the critical temperature. This is shown in Fig. 5.4. The most important element is copper at 66.0%, with oxygen as the second most important at 8.7%. This result is interesting for the same reasons that were explained in Section 4.



**(a)**

**(b)**

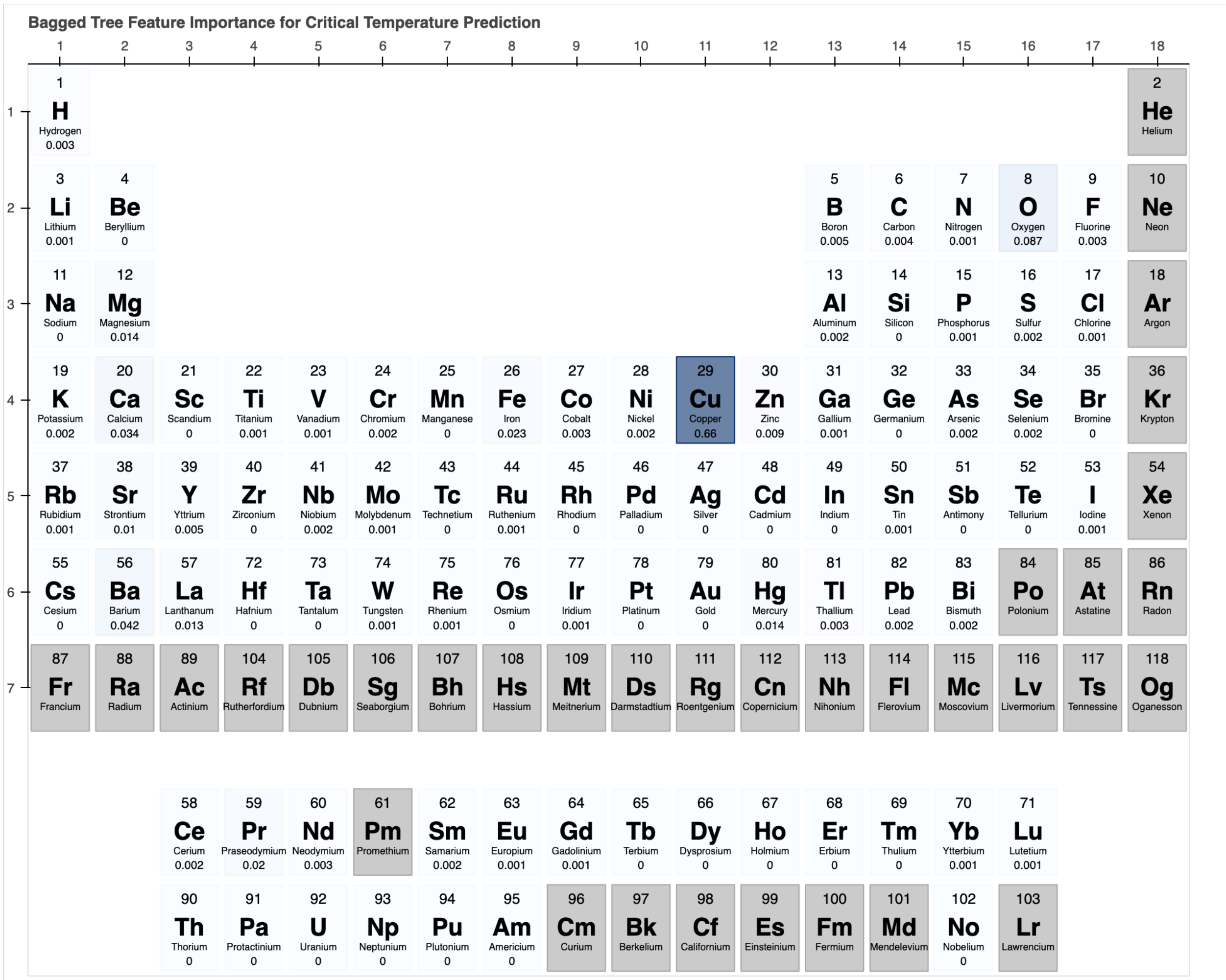
**Figure 5.2:** These plots are the result of our validation set. **(a)** Scatter plot of predicted critical temperature versus the true critical temperature. The red line is $y = x$ and only used for visualization. The green lines are shifted by ± 8.91 to represent the RMSE achieved by Roter and Dordevic [8]. The blue lines are shifted by ± 8.11 to represent the RMSE achieved by our practical model. **(b)** Scatter plot of Predicted $T_C$ - True $T_C$ versus True $T_C$. The red shaded region is the area where the predicted critical temperature is less than 0 K. Both plots make use of transparent points to better visualize density.



**(a)**

**(b)**

**Figure 5.3: (a)** Resolution histogram of our validation set on a model trained on a training set. **(b)** Resolution histogram of our testing set using the same model as that produced (a).



Bagged Tree Feature Importance for Critical Temperature Prediction

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1<br>H<br>Hydrogen<br>0.003 | | | | | | | | | | | | | | | | | 2<br>He<br>Helium |
| 2 | 3<br>Li<br>Lithium<br>0.001 | 4<br>Be<br>Beryllium<br>0 | | | | | | | | | | | 5<br>B<br>Boron<br>0.005 | 6<br>C<br>Carbon<br>0.004 | 7<br>N<br>Nitrogen<br>0.001 | 8<br>O<br>Oxygen<br>0.087 | 9<br>F<br>Fluorine<br>0.003 | 10<br>Ne<br>Neon |
| 3 | 11<br>Na<br>Sodium<br>0 | 12<br>Mg<br>Magnesium<br>0.014 | | | | | | | | | | | 13<br>Al<br>Aluminum<br>0.002 | 14<br>Si<br>Silicon<br>0 | 15<br>P<br>Phosphorus<br>0.001 | 16<br>S<br>Sulfur<br>0.002 | 17<br>Cl<br>Chlorine<br>0.001 | 18<br>Ar<br>Argon |
| 4 | 19<br>K<br>Potassium<br>0.002 | 20<br>Ca<br>Calcium<br>0.034 | 21<br>Sc<br>Scandium<br>0 | 22<br>Ti<br>Titanium<br>0.001 | 23<br>V<br>Vanadium<br>0.001 | 24<br>Cr<br>Chromium<br>0.002 | 25<br>Mn<br>Manganese<br>0 | 26<br>Fe<br>Iron<br>0.023 | 27<br>Co<br>Cobalt<br>0.003 | 28<br>Ni<br>Nickel<br>0.002 | 29<br>Cu<br>Copper<br>0.66 | 30<br>Zn<br>Zinc<br>0.009 | 31<br>Ga<br>Gallium<br>0.001 | 32<br>Ge<br>Germanium<br>0 | 33<br>As<br>Arsenic<br>0.002 | 34<br>Se<br>Selenium<br>0.002 | 35<br>Br<br>Bromine<br>0 | 36<br>Kr<br>Krypton |
| 5 | 37<br>Rb<br>Rubidium<br>0.001 | 38<br>Sr<br>Strontium<br>0.01 | 39<br>Y<br>Yttrium<br>0.005 | 40<br>Zr<br>Zirconium<br>0 | 41<br>Nb<br>Niobium<br>0.002 | 42<br>Mo<br>Molybdenum<br>0.001 | 43<br>Tc<br>Technetium<br>0 | 44<br>Ru<br>Ruthenium<br>0.001 | 45<br>Rh<br>Rhodium<br>0 | 46<br>Pd<br>Palladium<br>0 | 47<br>Ag<br>Silver<br>0 | 48<br>Cd<br>Cadmium<br>0 | 49<br>In<br>Indium<br>0 | 50<br>Sn<br>Tin<br>0.001 | 51<br>Sb<br>Antimony<br>0 | 52<br>Te<br>Tellurium<br>0 | 53<br>I<br>Iodine<br>0.001 | 54<br>Xe<br>Xenon |
| 6 | 55<br>Cs<br>Cesium<br>0 | 56<br>Ba<br>Barium<br>0.042 | 57<br>La<br>Lanthanum<br>0.013 | 72<br>Hf<br>Hafnium<br>0 | 73<br>Ta<br>Tantalum<br>0 | 74<br>W<br>Tungsten<br>0.001 | 75<br>Re<br>Rhenium<br>0.001 | 76<br>Os<br>Osmium<br>0 | 77<br>Ir<br>Iridium<br>0.001 | 78<br>Pt<br>Platinum<br>0 | 79<br>Au<br>Gold<br>0 | 80<br>Hg<br>Mercury<br>0.014 | 81<br>Tl<br>Thallium<br>0.003 | 82<br>Pb<br>Lead<br>0.002 | 83<br>Bi<br>Bismuth<br>0.002 | 84<br>Po<br>Polonium | 85<br>At<br>Astatine | 86<br>Rn<br>Radon |
| 7 | 87<br>Fr<br>Francium | 88<br>Ra<br>Radium | 89<br>Ac<br>Actinium | 104<br>Rf<br>Rutherfordium | 105<br>Db<br>Dubnium | 106<br>Sg<br>Seaborgium | 107<br>Bh<br>Bohrium | 108<br>Hs<br>Hassium | 109<br>Mt<br>Meitnerium | 110<br>Ds<br>Darmstadtium | 111<br>Rg<br>Roentgenium | 112<br>Cn<br>Copernicium | 113<br>Nh<br>Nihonium | 114<br>Fl<br>Flerovium | 115<br>Mc<br>Moscovium | 116<br>Lv<br>Livermorium | 117<br>Ts<br>Tennessine | 118<br>Og<br>Oganesson |
| | | | 58<br>Ce<br>Cerium<br>0.002 | 59<br>Pr<br>Praseodymium<br>0.02 | 60<br>Nd<br>Neodymium<br>0.003 | 61<br>Pm<br>Promethium | 62<br>Sm<br>Samarium<br>0.002 | 63<br>Eu<br>Europium<br>0.001 | 64<br>Gd<br>Gadolinium<br>0.001 | 65<br>Tb<br>Terbium<br>0 | 66<br>Dy<br>Dysprosium<br>0 | 67<br>Ho<br>Holmium<br>0 | 68<br>Er<br>Erbium<br>0 | 69<br>Tm<br>Thulium<br>0 | 70<br>Yb<br>Ytterbium<br>0.001 | 71<br>Lu<br>Lutetium<br>0.001 | | |
| | | | 90<br>Th<br>Thorium<br>0 | 91<br>Pa<br>Protactinium<br>0 | 92<br>U<br>Uranium<br>0 | 93<br>Np<br>Neptunium<br>0 | 94<br>Pu<br>Plutonium<br>0 | 95<br>Am<br>Americium<br>0 | 96<br>Cm<br>Curium | 97<br>Bk<br>Berkelium | 98<br>Cf<br>Californium | 99<br>Es<br>Einsteinium | 100<br>Fm<br>Fermium | 101<br>Md<br>Mendelevium | 102<br>No<br>Nobelium<br>0 | 103<br>Lr<br>Lawrencium | | |

**Figure 5.4:** The table displays the feature importance of each element in our Bagged Tree regression model. Specific element feature importance values are shown at the bottom of the respective cell. Blue tones indicate the relative importance with darker tones indicating increased importance. Light grey tones indicate that the element received a feature importance near zero. Dark gray tones indicates the element was not found in the chemical composition matrix.

# 6 Discussion and Conclusion

An important aspect of using machine learning algorithms to predict properties in materials is being able to encompass all the necessary information from the materials as to be able to tell the different atomic and crystalline structures apart [9]. A simple approach to this problem is by taking into account only chemical composition for the classification of superconductors and the prediction of their critical temperatures.

By exploring our dataset using a simple chemical composition encoding scheme, we found high accuracy in both the classification and prediction problems. The best performing model for both classification and regression was the bagged trees model, for which we have achieved a classification accuracy of 95% and a prediction RMSE of 7.97 K.

One of the important aspects of this model is its ability to predict critical temperature with high accuracy based solely on chemical composition. In principle, any property can be calculated given chemical composition. Knowing chemical composition you can obtain its specific Hamiltonian, as shown in Eq. 1.2; what is needed is where each nuclei sits in thermodynamic equilibrium, not taking into account any quantum fluctuations. When applying this to a theory such as BCS, the model begins to break down as BCS is a poor approximation to obtaining the critical temperature. In order to obtain more accurate predictions we would have to go beyond the BCS treatment, for example with better variational wavefunctions than the BCS wavefunction and variational quantum monte carlo which keep superconducting correlations.

A possible approach to understanding this model is by looking at the BCS theory superconducting transition temperature which is given by Eq. 1.1. The relevant parameter here is the Debye frequency which is related to the Debye temperature and at the same time related to the molecular weight and number of atoms present [3]:

$$T_D = \frac{\hbar\omega_D}{k_B} = \frac{h3n}{4k\pi}\left[\frac{N\rho}{M}\right]^{1/3} v_m$$

This relation indicates that in BCS theory heavier compounds will have a lower frequency and thus a lower $T_C$. If the element is lighter it will have higher frequency and higher $T_C$ — this is directly related to how these systems have a stronger coupling.

Despite using a method that does not include any relevant features for the theory of superconductivity such as resistivity, superfluid density, band structure features [8], our model is performing better than expected to so for a simple input. By using chemical composition we will not be able to obtain any theoretical model that may help in writing a new theory of superconductivity, but it does show a correlation among the compounds the present superconductivity.

Even though SuperCon has a limited number of known superconductors compared to non-superconductors, a clear trend is present in our data with an underlying pattern only the model is able to uncover. This model will help understand what materials or combination of materials are more likely to produce a superconducting material with either high or low critical temperature. However, we have to be aware of the potential presence of experimental bias due to the prevalence of more heavily studied superconductors, such as those that contain oxygen and copper.

Nevertheless, this project has provided some interesting insights regarding the role chemical composition plays in classifying superconductors and predicting critical temperature. We hope that this will be further explored for algorithms that can better incorporate the chemical composition and that this work provides a base for a more detailed analysis of the feature space to be done. Future work will require studying Isomaps and local linear embedding for dimensional reduction as well as methods such $k$-means clustering and further methods such as deep learning to gain a better understanding how this model is predicting critical temperature and classifying superconductors.

# References

[1] (1995), Journal of chemical information and computer sciences. **35** (5).

[2] (2021), "Supercon database," .

[3] Ashcroft, N. W., N. D. Mermin, *et al.* (1976), *Solid state physics*, Vol. 2005 (holt, rinehart and winston, new york London).

[4] Blundell, S. J. (2009), *Superconductivity: a very short introduction* (OUP Oxford).

[5] van Delft, D. (2012), Physica C: Superconductivity **479**, 30.

[6] Gron, A. (2019), *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (OReilly Media, Inc.).

[7] Jain, A., S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, and K. a. Persson (2013), APL Materials **1** (1), 011002.

[8] Roter, B., and S. Dordevic (2020), Physica C: Superconductivity and its Applications **575**, 1353689.

[9] Schmidt, J., M. R. Marques, S. Botti, and M. A. Marques (2019), npj Computational Materials **5** (1), 1.

[10] Schrieffer, J. R. (2018), *Theory of superconductivity* (CRC Press).

[11] Skomski, R. (2008), *Simple models of magnetism* (Oxford University Press on Demand).

[12] Stanev, V., C. Oses, A. G. Kusne, E. Rodriguez, J. Paglione, S. Curtarolo, and I. Takeuchi (2018), npj Computational Materials **4** (1), 1.

[13] Tinkham, M. (2004), *Introduction to superconductivity* (Courier Corporation).

[14] Townsend, J. S. (2012), *A Modern Approach to Quantum Mechanics, 2nd ed.* (University Science Books).

# A Common Terms

1. **Random Forest**: The random forest algorithm works by creating an ensemble of decision trees and classifying via the mean output of the set. Random forests improve upon a single decision by reducing tendencies to overfit. [6, p. 197]

2. **Bagging**: Bagging algorithms work by sampling with replacement from the data and creating a variety of new datasets. A different decision tree is then trained on each dataset and the output of is formed by the average of the forests predictions. [6, p. 192]

3. **Gradient Boosting**: Gradient boosting adds predictors to an ensemble and attempts to correct residual errors. [6, p. 203]

4. ***k*-NN**: The $k$-nearest neighbor algorithm works by evaluating the Euclidean distance between the input vector and dataset vectors. The prediction comes from averaging the labels of the $k$ nearest points to the input. [6, p. 22]

5. **Principal Component Analysis**: A popular dimensional reduction algorithm, principal component analysis identifies and projects the data onto the closest hyperplane. [6, p. 219]

# B Replication of Roter and Dordevic

## B.1 Replication of Roter and Dordevic's Classification Results

To replicate Roter and Dordevic's results, we used our un-normalized and imbalanced chemical composition dataset. We then ran PCA and reduced the initial vector space of size 81 down to 15 components. See Appendix A for more details on PCA. Like Roter and Dordevic we trained using the entire dataset, and tested on the same data used for training [8]. These test results are shown in Table B.1.

| Model | Class | Precision | Recall | F1-Score |
|---|---|---|---|---|
| Boosted Trees | 0 | 0.78 | 0.62 | 0.69 |
| | 1 | 0.96 | 0.98 | 0.97 |
| | Accuracy | | | 0.95 |
| Bagged Trees | 0 | 1.00 | 1.00 | 1.00 |
| | 1 | 1.00 | 1.00 | 1.00 |
| | Accuracy | | | 1.00 |
| $k$-NN ($k$=5) | 0 | 0.90 | 0.67 | 0.77 |
| | 1 | 0.96 | 0.99 | 0.98 |
| | Accuracy | | | 0.96 |

**Table B.1:** Testing accuracy by model with a direct adaption of Roter and Dordevic's methodology.

As we have previously stated, the lack of a train test split calls the validity of these accuracies into question [6]. However, these results verify that we correctly understood Roter and Dordevic's technique.

To correct Roter and Dordevic's methodology, we started with a balanced chemical composition dataset. Unlike the imbalanced data, this has an equal number of superconductors and non-superconductors. Once again we ran PCA and reduced the vector space from 81 to 15. We them conducted a 70:15:15 train, test, validate split. We trained on the designated training data and validated on the validation data.

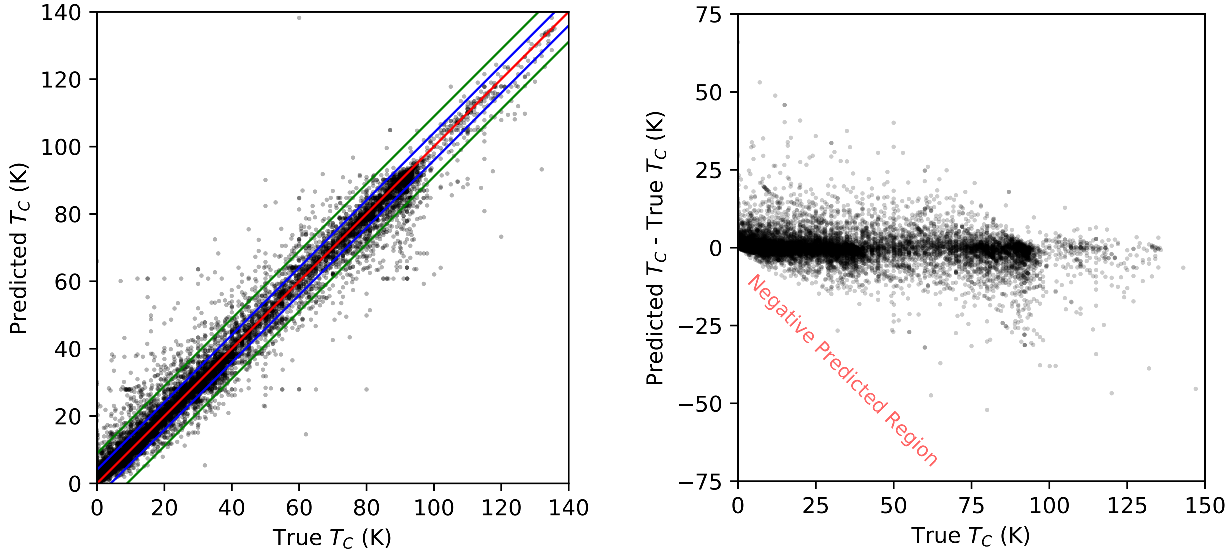
| Model | Class | Precision | Recall | F1-Score |
|---|---|---|---|---|
| Boosted Trees | 0 | 0.86 | 0.90 | 0.88 |
| | 1 | 0.90 | 0.85 | 0.87 |
| | Accuracy | | | 0.88 |
| Bagged Trees | 0 | 0.89 | 0.96 | 0.92 |
| | 1 | 0.95 | 0.87 | 0.91 |
| | Accuracy | | | 0.92 |
| $k$-NN ($k$=5) | 0 | 0.92 | 0.91 | 0.91 |
| | 1 | 0.91 | 0.91 | 0.91 |
| | Accuracy | | | 0.91 |

**Table B.2:** Testing accuracy by model on Roter and Dordevic's technique with a train-test-validate split implemented.

Table B.2 provides a clearer view on the effectiveness of Roter and Dordevic's method. While they cited $k$-NN as the most effective [8], we found Bagged Trees performed better on a separated test set. Once train-test splits were implemented, the best model accuracy dropped to 92%.

## B.2 Replication of Roter and Dordevic's Regression Results

First we will go over our attempted reproduction of Roter and Dordevic's results. We achieved a root-mean-square error (RMSE) of approximately 4.11 K. This is significantly smaller than the RMSE of 8.91 K presented by Roter and Dordevic and we are suspicious that this may involve the missing 9,428 samples from SuperCon [8]. In other cases, the results of our attempt agree with this smaller RMSE [1]. We found a $\Delta T_C$ with a mean value of $-0.0093$ K and a standard deviation of 4.1125 K. The exact distribution is shown in Figure B.2.



**(a)**

**(b)**

**Figure B.1:** These plots are the result of our attempt to recreate Roter and Dordevic's results. **(a)** Scatter plot of predicted critical temperature versus the true critical temperature. Red line is $y = x$ and only used for visualization. The green lines are shifted by ± 8.91 to represent the RMSE achieved by Roter and Dordevic [8]. The blue lines are shifted by ± 4.11 to represent the RMSE achieved by our recreation. **(b)** Scatter plot of Predicted $T_C$ - True $T_C$ versus True $T_C$. The red shaded region is the area where the predicted critical temperature is less than 0 K. Both plots make use of transparent points to better visualize density.

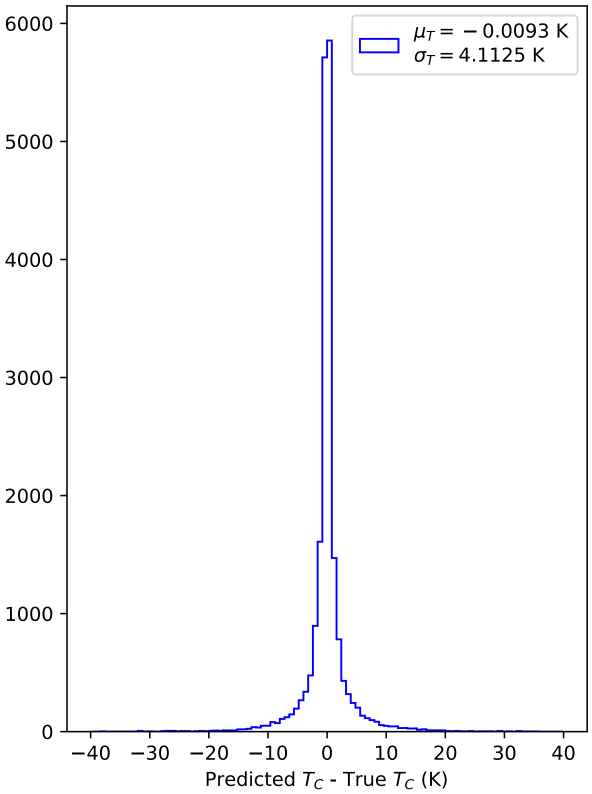


**Figure B.2:** Resolution histogram when recreating the results of Roter and Dordevic.