UCLouvain

École polytechnique de Louvain

Study of Superconductors Using Machine Learning

Author: **Najlae SAHBI**
Supervisor: **Gian-Marco RIGNANESE**
Readers: **Pierre Paul DE BREUCK, Christoph DE VLEESCHOUWER**
Academic year 2022–2023
Master [120] in Data Sciences Engineering

Contents

Abstract

The development of simulation and modelling methods has made the creation of databases dedicated to materials research possible. This has enable scientists to apply machine learning algorithms to solid-state materials research. One area of research is the development of superconducting materials. This work analyses more than ten thousand superconducting compounds and uses classification and regression algorithms to predict the temperature at which the electrical resistivity vanishes.
Key words: machine learning - classification - regression -critical temperature - superconductors.

Chapter 1

Introduction

The 21^{st} century has been characterized by an exponential growth of available data as well as vast improvements in computing power. This made it ideal for the development of machine learning methods. Generally speaking, these are methods that leverage data to accomplish a specified task. The most well-known applications today concern image recognition, self-driving cars, and speech processing (to name but a few examples). However, data-driven fields such as biology and chemistry have also been increasingly employing these statistical methods.

Another field of research that has been increasingly using machine learning methods is the one of solid-state materials, whereby algorithms are used in the study of the properties of materials.

There are 118 known elements and an infinite number of ways in which these can be combined, thus it should come as no surprise that most great discoveries in materials science happen either as the result of a happy accident (the discovery of Teflon for example) or as the fruit of years of careful study and experimentation. The search space in materials science for any problem is infinitely big, estimated to be in the scale of 10^{100}, hence the increasing interest in using computational methods to explore it more rapidly and more effectively than would be possible otherwise.

We can trace back the introduction of computational methods in the materials science research field back to the late 1960s when Density Functional Theory (DFT) was first developed by Nobel prize winners Walter Kohn and Pierre Hohenberg [1]. DFT has since become an essential modeling tool in electronic structure calculations, only gaining more popularity since the 1990s with the increase in available computing power.

With the development of large-scale computing, modeling and simulation methods,

more solid-state data has become available, making it possible to use machine learning algorithms for materials science study. Such algorithms are able to detect patterns in large datasets, as well as underlying hidden laws.

The way these algorithms function, as shown on Figure 1.1, is through an iterative process. One the data has been retrieved, prepared through various processing and featurization steps, a model is trained and evaluated. It is not unusual to return to the data processing step while tuning the model of choice, just as one usually trains several models and compares their performances to select to best one. Once that is done the final results can be visualized and analyzed.

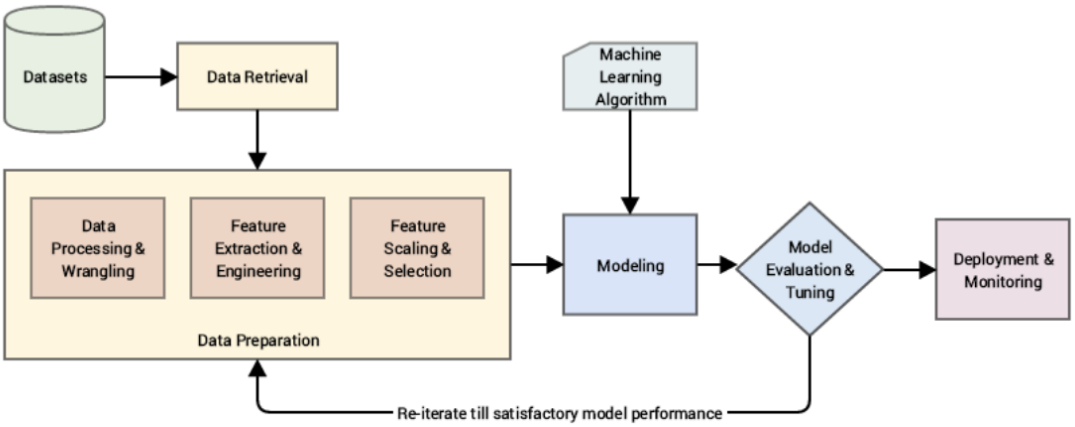

Figure 1.1: The supervised leanrning workflow

A general review of the literature on the applications of machine learning to solid state materials yields on consensus: the greatest challenge currently is to predict the structure of a crystalline solid from its chemical composition [2]. Some search algorithms (such as simulated annealing and genetic algorithms)[2] have yielded some promising results but they are computationally expensive to implement as the many arrangements of atoms in the three-dimensional space make for a large search space.

A materials science in which machine learning algorithms have been successful is in the prediction of the properties of compounds such as the bulk modulus (i.e. the resistance to compression) of compounds, the topological states of condensed matter, and even the prediction of structures [2]. I have chosen for this work to focus on the phenomena of superconductivity. Superconductors are materials that are perfect electrical conductors (no electrical resistance) beyond a certain temperature called the critical temperature. There is few research tackling the application of machine learning to superconductivity research.

In this work I first discuss the concept of superconductivity: its origins, the theory behind it as we understand it today and its importance and applications. An important aspect of both machine learning and materials research is the data used, which is why I spend two sections discussing the data itself and the transformations applied to it. Afterwards I develop a classification model that categorizes superconductors based on a predetermined optimal critical temperature value. Using the results from the classification model, a regression model is then developed to make critical temperature predictions. Finally, an attempt is made at predicting whether or not a compound is a superconductor, using data retrieved from the Materials Project website.

Chapter 2

Superconductivity

It has been established that, as the temperature of an electrical conductor increases, so does its electrical resistance. On the other hand, when the temperature decreases, the electrical resistance does as well. For most materials, the theoretical electrical resistance (theoretical because the absolute zero temperature, 0 K, cannot be reached) is not null. That is not the case for superconductors (previously called supraconductors), so called because of the vanishing of their electrical resistance beyond a characteristic temperature called the critical temperature.

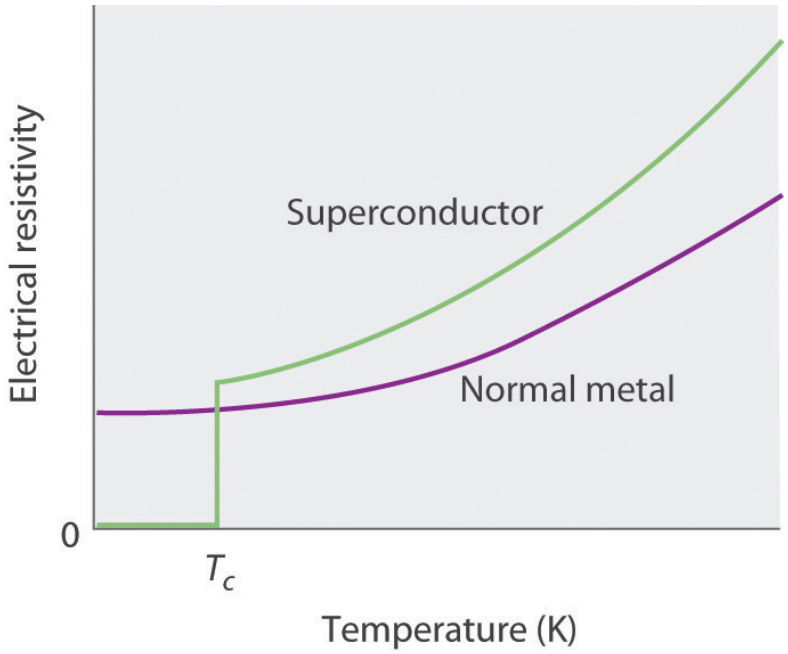

Figure 2.1: The electrical resistivity as a function of the temperature

In the early 20^{th} century, Dutch physicist Heike Kamerlingh Onnes was conducting experiments at low temperatures. Specifically, he was interested in the evolution of the electrical resistance of metal wires when the temperature was increasingly lowered. Some physicists thought that in such a case, the electrons would not flow in the material and thus the resistance would increase the lower the temperature gets. Onnes's findings showed the exact opposite: the electrical resistance actually got lower with the temperature. He was at first conducting his investigation using liquid hydrogen but he managed to, in 1908, synthesize liquid helium leading to a veritable revolution in low temperature physics. Then three years later, Onnes discovered that a mercury wire immersed in liquid nitrogen (i.e. at around 4.2K) has a nonexistent resistance [3]. Thus the phenomenon of superconductivity (called supraconductivity at the time) was discovered.

Since the discovery of the first superconducting material, many more have been discovered: Niobium in 1932 [4], ytrium barium copper oxide compounds (YBCO) in 1987[5], and more recently carbonaceous sulfure hydride on 2020.

Superconductor research is on-going and for good reasons. Indeed, superconductors have many practical applications. They are used in medical imaging in MRIs where the magnetic field is created by superconductors. Superconductors are also used in transportation in maglev trains. And because of their characteristic property, scientists are interested in their use in electric cars to reduce electric power loss [6].

While we have been able to observe the superconductivity phenomena, it was not until forty years later that a theory was developed to adequately explain it. In 1957, Bardeen, Cooper and Schrieffer proposed the BCS theory (named after them) explaining superconductivity on a microscopic level. They received the Nobel Prize for it and it is used today to explain some superconductors, but not all. The theory explains superconductivity as a destabilization of the electronic structure near the Fermi surface, resulting from the formation of Cooper pairs. Cooper pairs are electrons that overcome the Coulomb repulsion and bound together at low temperatures [7]. The creation of many of such pairs attract positive charge and deform the lattice, resulting in an uninhibited electric flow.

The compounds whose superconductivity can be explained by the BCS theory are called conventional superconductors, and the others are called unconventional.
Another characteristic of superconductors is the Meissner effect. When a superconductor is cooled below its critical temperature, it complete expels magnetic fields. The vanishing electrical resistance and the Meissner effect both characterize

superconductor. The maximum strength of the magnetic field that can be applied to a superconductor before it no longer remains superconducting is called the critical field and it is denoted in the literature as H_c. Compounds with a unique H_c are classified as type I superconductors and these are the ones the BCS theory explains best. Compounds behaving as superconductors below a maximum magnetic field and as normal conductors above another magnetic field are type II superconductors and we still do not understand how they work.

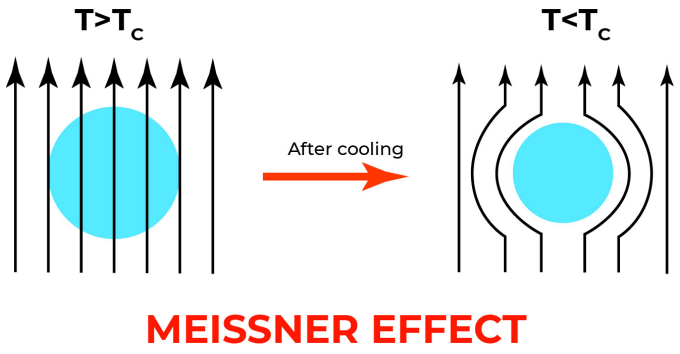

Figure 2.2: Diagram of the Meissner effect

This inexplicability of the superconducting phenomena makes it an ideal field for machine learning applications. Indeed, the models may reveal some yet unknown link between certain characteristics of the material and its critical temperature as well confirm what the theory suggests.

Chapter 3

The Dataset

The dataset used throughout this project is the SuperCon database and it is a comprehensive list of superconducting components. Originally published and maintained by the National Institute of Materials Science (NIMS), the database was not available on their website until December 16, 2022, when a new version, denoted MDR SuperCon Datasheet (doi: https://doi.org/10.48505/nims.3739), was published. There was however an older version of the database, simply denoted SuperCon, available via github.

While most of the work has been redone using the newer version of the database (MDR SuperCon Datasheet), I include a comparison between the older and the newer versions to highlight the improvements made to the latter and justify why it is the better choice for the machine learning models developed later on.

Note: the older dataset will denoted **SuperCon** whereas the newer one will be denoted **MDR SuperCon** throughout the rest of the paper.
After the description and the comparison of the two sets of data, I describe the steps undertaken to make the data ready for machine learning applications. I then present a few statistics characterizing it.

3.1 Data Description

The **Supercon** database contains 16 414 observations and only 2 columns:

- `name`: contains the chemical formula of the compounds;
- `Tc`: contains the critical temperature of the superconductor, superconductors with no reported critical temperature have been assigned a critical temperature of 0 K.

The **MDR SuperCon** database on the other hand counts 26 323 observations and 7 columns:

- `num`: the identifier of the observation;
- `name`: the common formula (i.e. the empirical formula) of the material;
- `element`: the chemical formula of the component;
- `str3`: common name of the compound used to search on the database;
- `utc`: the unit of the reported critical temperature (all reported in Kelvin);
- `tc`: the critical temperature of the compound, all superconductors with no reported critical temperature have been assigned a critical temperature of 0 K;
- `journal`: the source the experimental results are derived from.

The MDR Supercon contains more observations than its predecessor. Additionally, it contains more complete information: indeed, the number of compounds without a reported critical temperature (i.e. those for which the reported T_c in the database is equal to 0K) is lower in the MDR SuperCon (71 observations) than in the SuperCon (3 966 observations).

There is a greater range of critical temperatures in the newer database, with temperatures ranging from 0.00027 K to 323.0K. The average critical temperature is also higher, almost double what it is for the SuperCon. Which could prove to be helpful with the interest in high-temperature superconductors. Indeed, the MDR SuperCon contains more than twice the number of reported high-temperature superconductors than the SuperCon.
Table 3.1 summarizes these comparisons.

Table 3.1: Comparison of the old (SuperCon) and new (MDR SuperDon) databases

	SuperCon	**MDR SupeCon**
Number of observations	16 414	26 323
Number of variables	2	7
Observations with no reported T_c	3 966	71
Maximum T_c (in K)	143.0	323.0
Minimum T_c (in K)	0.0005	0.0002675
Mean T_c (in K)	17.97	32.02
Number of unique observations	16 414	24 561
Number of high-temperature compounds	1239	2 836
Number of compounds with different T_c (in K)	0	7016

3.2 Data Processing

From here on out, every operation will be done on the MDR SuperCon. Before it can be used in any machine learning model, the data needs to be processed first. A brief look-over one of the columns of interests (the one containing the chemical formulas) shows that it includes non-stoichiometric compounds. Many superconductors are non-stoichiometric, wherein doping greatly influences the critical temperature. In an effort to be as precise as possible, the compounds are considered stoichiometric and the variables denoting non-stoichiometry considered to be equal to 0. I also set aside the compounds with no reported critical temperature (i.e. those with $T_c = 0$ K).

As observed during the exploration of the data, the database contains several identical compounds with different reported critical temperatures. A way to explain that would be to see what pressures the compounds were subject to when their electrical resistance vanished but as we only have the reported critical temperatures, this is only a hypothesis. To circumvent the issue, the average of the critical temperatures was taken for every compound with more than one.
Finally I make sure to remove any redundancy from the data. To do so I remove any repeating observations: rows where the empirical and chemical formulas, as well as the critical temperature are identical.
The final dataset contains 17 144 observations and is ready for machine learning.

3.3 Data Distribution

Plotting the critical temperatures in the processed database, we observe the result on Figure 3.1 where the bin size is 2K. The most noticeable characteristics of the data distribution is its skewedness. Indeed, we observe that most of the critical temperatures are very low, with more than half of them falling below 20K.

We also see that the number of observations greater than 150K is negligible in comparison to the rest. Indeed, only 3 compounds have critical temperatures beyond that point, hence why the horizontal axis does not extend all the way to the maximum value of 323.0 K.

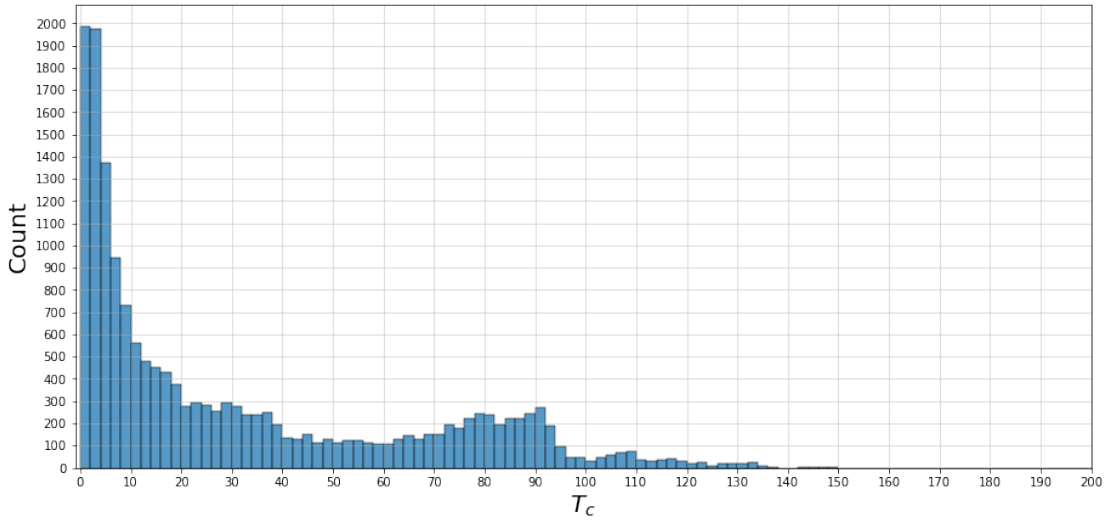

Figure 3.1: The distribution of the T_c (in K)

Rather than trying to determine the distribution of the original critical temperatures, I took their natural logarithm (as seen on Figure 3.3). The distribution already looks more bell-shaped than on Figure 3.1. The `fitter` is then used to fit several known probability distributions to the data and the one with the best fit is selected.

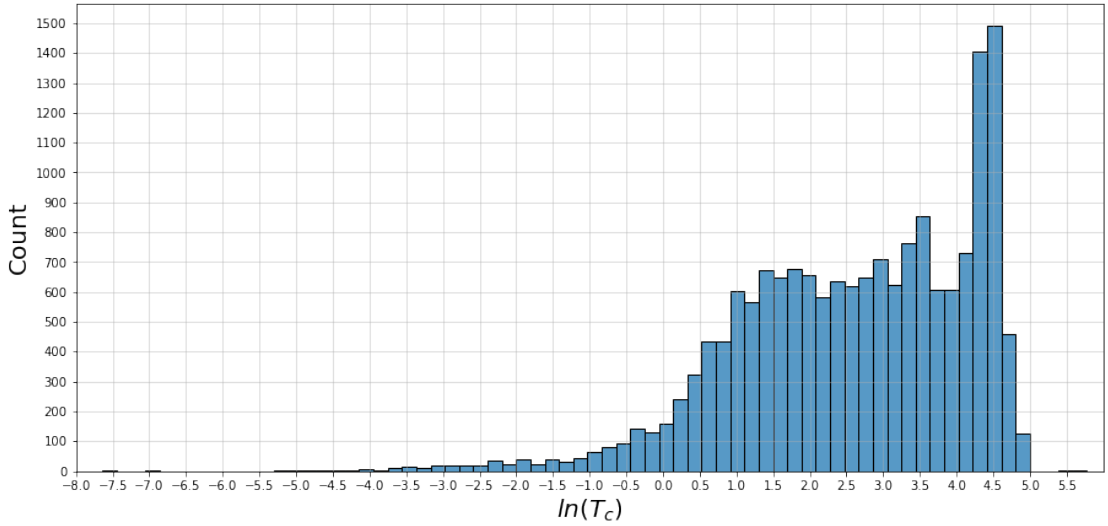

Figure 3.2: The distribution of the natural log T_c (in K)

From the results obtained after running the function, five distributions were considered to best fit the natural logarithm of the critical temperature:

- the skewed normal distribution;
- the Laplace asymmetric distribution;
- the generalized logistic distribution;
- the skewed cauchy distribution;
- the logarithm of the gamma distribution.

The criteria for comparing the fit of each distribution is summarized in Table 3.2 on which we see that the one with the smallest sum square error and the highest Akaike and Bayesian Information criteria (AIC and BIC respectively) is the skewed normal distribution.

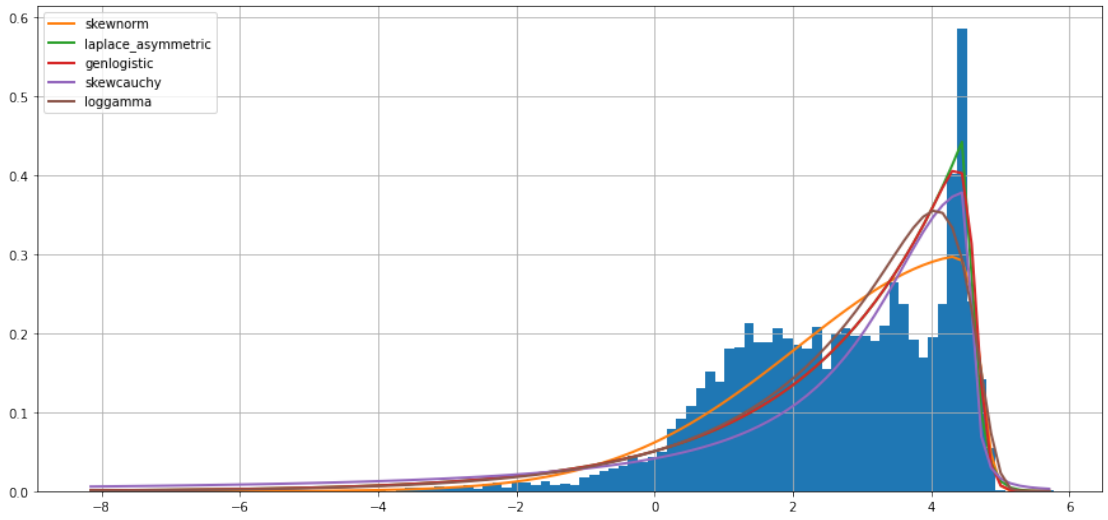

Figure 3.3: The distribution of $ln(T_c)$ (in K)

Table 3.2: Criteria for distributions fitted to $ln(T_c)$

	sum square error	aic	bic
skewnorm	0.174	1152.3	-197105.9
Laplace asymmetric	0.205	810.7	-194313.3
genlogistic	0.219	835.3	-193153.1
skewcauchy	0.268	722.9	-189710.4
loggamma	0.272	911.5	-189432.2

Chapter 4

Methods

Having explored the dataset and dealt with the more obvious cleaning tasks, we now move onto the task of supplementing it with more useful information. At the beginning of this section, the only information we have is the chemical composition of the compounds as well as their critical temperatures. Here I present the data mining tools used to featurize the data and a description of the features resulting from the process.

4.1 Matminer

As mentioned in the introduction, the amount of data pertaining to materials research has been increasing with the development of simulation and modeling methods. This data is generally stored in databases maintained by different organisms. A lack of centralization of data makes it rather cumbersome for scientists to conduct their research. Matminer is data mining tool that seeks to make it easier on users to access whatever data they may require by providing an interface that makes it so one does not need to learn how to use the several APIs (Application Programming Interface) of every data source. At the time of writing this, Matminer allows the data retrieval from four materials databases[8]: Citrination, the Materials project, the Materials Data Facility, and the Materials Platform for Data Science.

Matminer has also been designed with machine learning in mind. While it does not directly implement the various supervised and unsupervised learning algorithms directly, it was created to integrated with well-known data science tools. Indeed, it is very easy to use libraries such as `pandas`, `numpy`, and `scikit-learn` in tandem with the data extracted from one of the data sources.

A crucial step in the data science process is the featurization of the data. It

is when the data at hand is transformed or augmented to be machine readable. There is no unique way to do either thing as this step is very much application-dependent. It is the case as well in materials research, where the information we can add depends on the information we already have. With this in mind, Matminer currently has 39 feature extraction modules [8] that are grouped by the type of the input data:

- composition: data about the chemical composition of compounds;
- crystal structure: data about how the compounds occupy the three-dimensional space;
- density of electronic states: data about the energy states of compounds;
- band structure: information about the energy levels of electrons;
- atomic site: statistics about the location of atoms in a compound.

Figure 4.1 displays the capabilities of Matminer: from data retrieval to feature extraction, to machine learning applications and visualization tools.

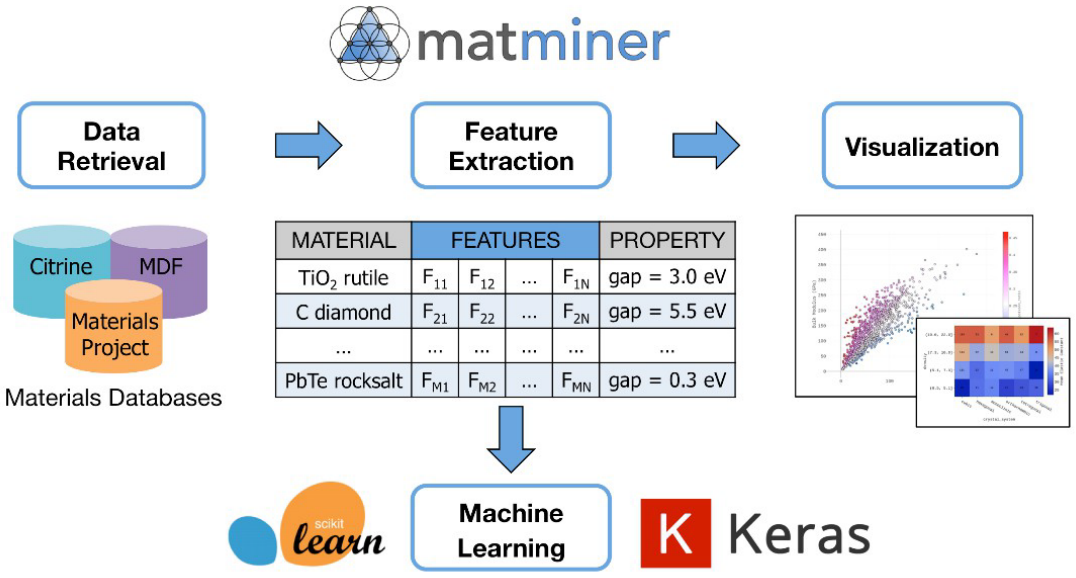

Figure 4.1: Functioning of Matminer

4.2 Features

The **MDR SuperCon** dataset contains only information about the chemical compositions of the compounds. As such the Matminer featurizer that was used

was based on that. With what I had on hand, the featurizer that worked best was the `ElementProperty` one. It provides statistics (the mean, the standard deviation, the maximum value, the minimum value and the range) of various element properties such as the electronegativity, the thermal conductivity, and the melting point. After featurization, I had a database with 182 columns. 20 of these newly generated columns contained more than a quarter of NAs and were subsequently dropped from the dataframe. Additionally, 178 rows were also found to have missing values and were removed as well. By the end of the process, the database had 16 721 observations ready to be used for machine learning.

Chapter 5

Classification Model

Critical temperature prediction is a problem that naturally lends itself to a regression approach which will be developed in the next chapter. In this one however, I approach the problem as a classification task. This approach was based on the categorization of superconductors as low-temperature or high-temperature compounds. Indeed, the rule says that if a material exhibits zero electrical resistance at temperatures greater than the boiling point of liquid nitrogen (which is at 77 K), then it is a high-temperature superconductor.
In this chapter, a random forest classification model is trained on the dataset containing the electronic properties of each compound. The performance of the classifier is first evaluated when trying to classify the observations into low- and high-temperature superconductors. Different values for the threshold critical temperature are then tested to see how it affects the performance of the model, evaluated on several metrics. Finally, the hyper-parameters of the model are fine-tuned and the variables with the highest effect on the critical temperature analyzed.

5.1 The Classification Algorithm

One of the most widely used classification algorithms is the random forest method. It is an ensemble learning algorithm that is composed of many decision trees [9]. As far as classification methods go, classification trees are easy to use and offer an intuitive approach to the categorization of data. They perform poorly however on large datasets where the interactions between variables may be complex. They are also not exempt from the problem of overfitting. Random forests on the other hand manage to overcome those problems by employing ensemble learning and introducing some randomness. Indeed, each generated decision tree votes for a class and the output is therefore the class that has the most votes. Moreover, the construction of decision trees involves the generation of independent and identically

distributed random vectors that help against the problem of overfitting. How a random forest classifier works is summarized in Figure 5.1.
Random forests are quite adept at dealing with non-linear datasets where the relationship between the explanatory variables and the target may not be linear. Another advantage is its ability to handle data without needing it to be extensively processed: it renders good predictions without having to, for instance, normalize or scale the data.

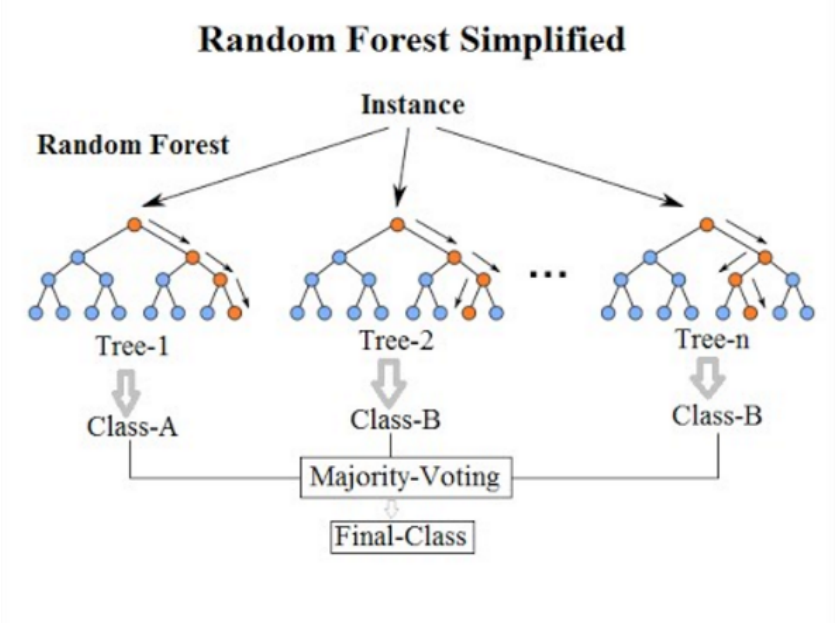

Figure 5.1: Diagram detailing the functioning of a random forest classifier

5.1.1 Performance Metrics

One cannot simply train an algorithm without having the means to measure its performance. In the case of classification algorithms, there are several metrics one can use, all relatively easy to understand from the error matrix. The rows on the matrix indicate the predicted class while the columns indicate the true one. If the positive class is predicted and that is the real class, the prediction is considered to be a true positive ("TP"). If on the other hand the class predicted is the negative one when the true class is positive, the prediction is considered to be a false negative ("FN"). The same reasoning is applied to understand what false negative ("FP") and true negative ("TN") mean.

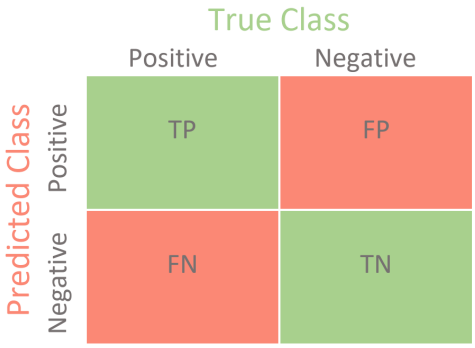

Figure 5.2: Error matrix

The simplest evaluation metric is the accuracy and it is the ratio of the correct predictions and the total number of predictions made:

$$accuracy = \frac{tp + tn}{tp + tn + fn + fp} \tag{5.1}$$

The accuracy is useful but it can be misleading in the instances where the classes are unbalanced. Therefore, other metrics need to be considered:

$$precision = \frac{tp}{tp + fp} \tag{5.2}$$

$$recall = \frac{tp}{tp + fn} \tag{5.3}$$

And since this is a binary classification problem, I also used the F_1 score, which is defined as the harmonic mean of the precision and the recall.

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{5.4}$$

5.2 Defining the Classes

In this section the processed and featurized data (as described in Chapter 4) is split into two classes based on the value of the critical temperature. The first split is made at the boiling temperature of liquid nitrogen (77K), the temperature that separates high-temperature superconductors and low-temperature ones. Afterwards, several more threshold temperatures are tried to see if a more optimal separation point can be found.

5.2.1 High-Temperature Superconductors Cutoff

I split the data in two and labeled it thusly:

- **Label 0:** the critical temperature of the compound is strictly smaller than 77K;
- **Label 1:** the critical temperature of the compound is greater or equal to 77K.

This leads to a very unbalanced dataset as seen on Figure 5.3 where only 2683 observations can be classified as high-temperature superconductors, while the rest are low temperature ones.

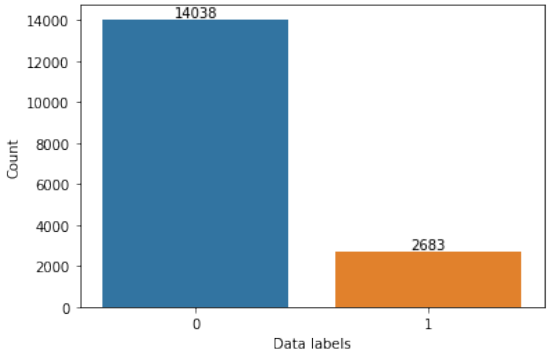

Figure 5.3: Labeled classes with cutoff = 77K

As seen on table 5.1, the random forest classifier performed very well on the training set across all metrics. Predictably, the testing scores were lower but still quite high, with a precision of 85

Table 5.1: Training and testing scores with cutoff = 77K

	Training	Testing
Accuracy	0.99	0.95
Precision	0.99	0.85
Recall	0.99	0.82
F1	0.99	0.84

5.2.2 Cutoff Temperature Selection

Following that first classification model, I decided to train several random forest classifiers in order to determine the optimal value of the cutoff temperature. Therefore, several classifiers were evaluated on threshold temperatures ranging from 4K

to 150K and the performance metrics defined above are then plotted as a function of it.

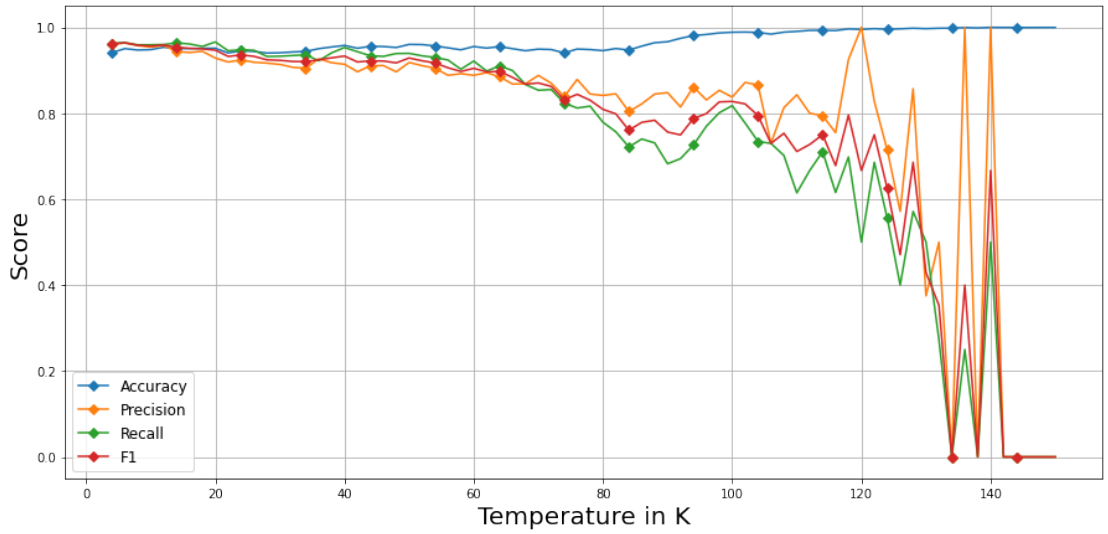

Figure 5.4: Performance of classifier as a function of cutoff temperature (in K)

As seen on Figure 5.4, the precision, recall and f1 scores all steadily decline as the temperature approaches 100K, beyond which they appear to vary in an erratic manner (large increases followed by sharp decreases). The accuracy on the other hand keeps increasing until it reaches a value of 1 when the classes become more and more unbalanced. I have decided to remove from consideration all temperatures greater than 60K as my goal is to find one where all the performance metrics behave in more deterministic manner.

We can spot on Figure 5.5 two interesting points: the first one at 24 K and the the second at 40K. The accuracy and precision scores are very close (the exact same in the case of the precision) whereas the recall and f1 scores at $T_c = 24$K is slightly higher.

Moreover when the categorization of the data is done based on the threshold set at 24K (data labeled 1 when $T_c \geq 24$ and 0 when $T_c < 24$), the resulting classes are more balanced than when it is set at 40K, as can be seen on Figures 5.6 and 5.7.
As a result, the optimal cutoff temperature for the classification model has been chosen to be equal to 24 K.

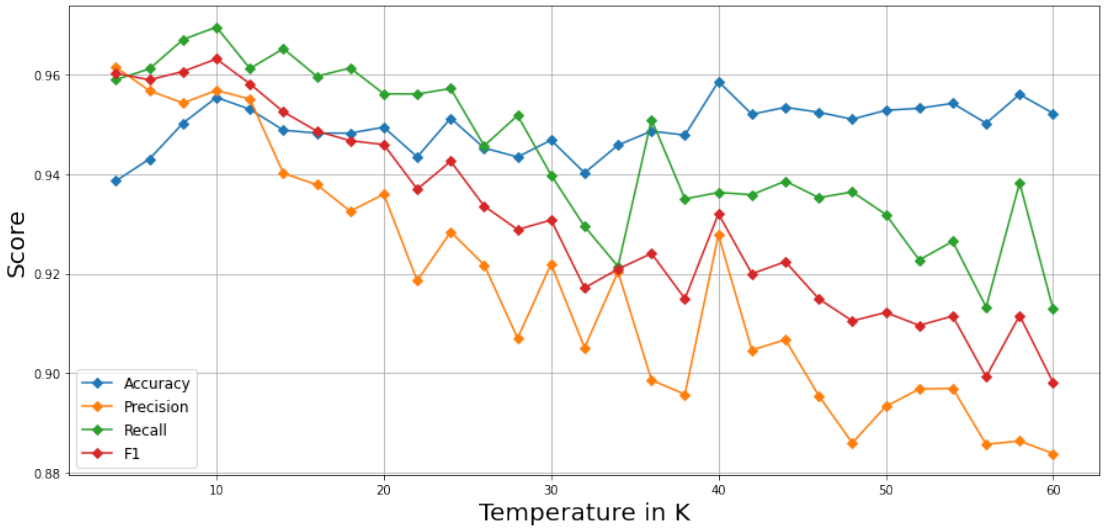

Figure 5.5: Performance of classifier as a function of cutoff temperature $\leq$ 60K

Table 5.2: Performance scores at 24K and 40K

	Acuracy	Precision	Recall	F1
24K	0.951	0.928	0.957	0.942
40K	0.958	0.928	0.936	0.932

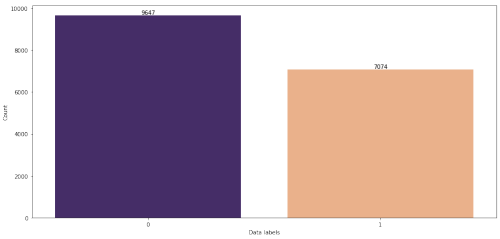

Figure 5.6: Classes at 24K

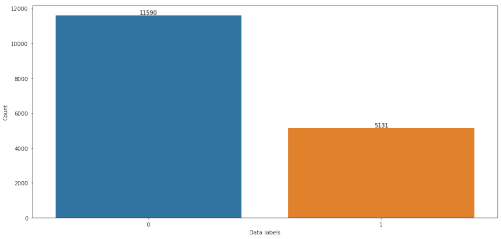

Figure 5.7: Classes at 40K

5.3 Tuning the Hyper-parameters

Having decided on the point on which the data will be split, we move onto tuning the hyper-parameters of the random forest classifier. Implemented using the `sklearn` package, we have the ability to change a few hyperparameters. The ones I chose the focus on are `n_estimators` which is the number of decision trees generated and `max_features` which the number of explanatory variables integrated in the

model.
We see on Figure 5.8 that, in general, all scores increase with the number of decision trees. It is however interesting to note that when `n_estimators` = 300, the accuracy, recall and F1 experience a local maximum before a decline at `n_estimators` = 350. Such lack of monotonicity has been observed and studied by Boulesteix and Probst [10] where they advocate against tuning the number of decision trees in a random forest classifier. Nevertheless, based on Figure 5.8, I have decided on a classification model with 300 decision trees.

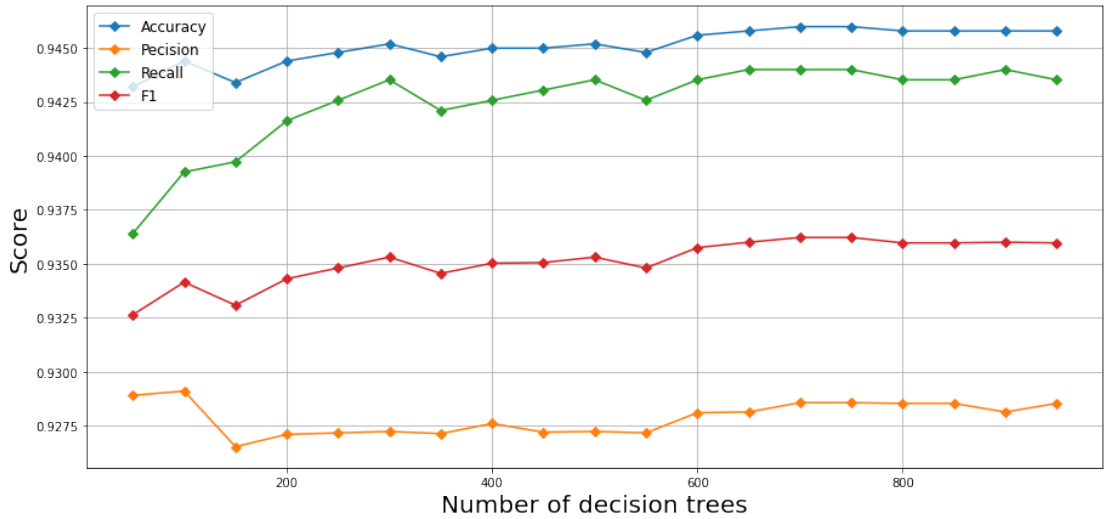

Figure 5.8: Performance of classifier as a function the number of decision trees

The next parameter to be tuned is the number of features. According to Hastie et al. [11], when it comes to classification problems, the optimal number of features given N features is generally $\sqrt{N}$. I plotted the performance of the classifier as a function of the number of features on Figure 5.9 and could not identify with reasonable certainty an optimal number. Therefore, taking as a reference the workd of Hastie et al. [11], the theoretical optimal number of features is $\sqrt{177} \approx 13$.

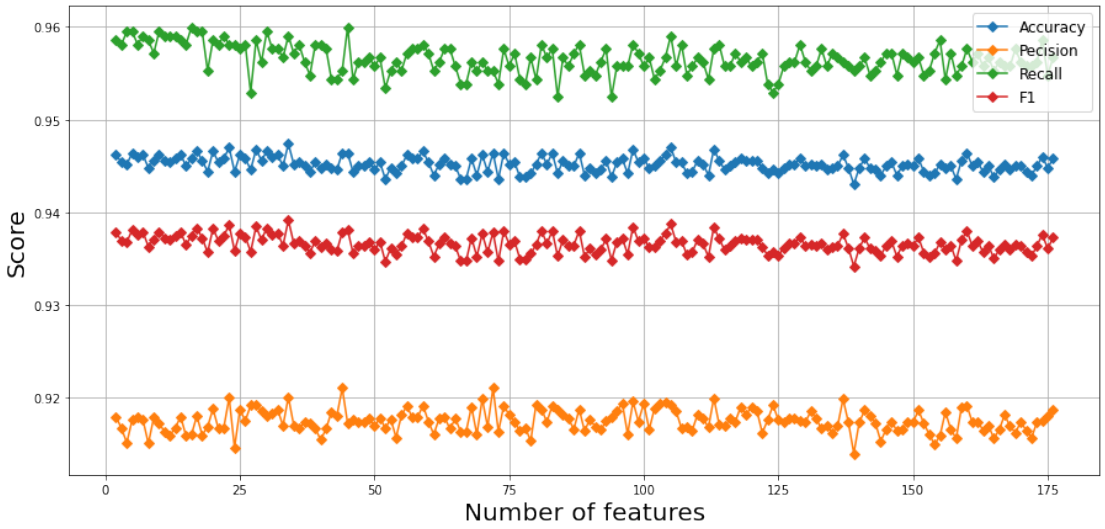

Figure 5.9: Performance of classifier as a function of the number of features

5.4 Best Explanatory Variables

The Random Forest Classifier, as implemented in the `sklearn` library, performs an implicit feature selection, hence its superior performance on larger datasets and its robustness to overfitting. Indeed, the classifier can be used to evaluate the importance of features via the Gini importance. This criteria measures the evolution of the model fit when a variable is removed, thus the more the performance dicreases, the more important the removed variable was.
Displayed on Figure 5.10 are the 13 most relevant features to the classification of compounds into low-temperature ($T_c < 24$K) and high-temperature ($T_c \geq 24$K). These are, in decreasing order of importance:

1. the range of the Mendelev number;
2. the standard deviation of the thermal conductivity;
3. the range of the thermal conductivity;
4. the range of the ground state volume;
5. the range of the atomic radius;
6. the average deviation of the ground state volume;
7. the range group;

8. the column range;
9. the range of the Mendelev number;
10. the range of electronegativity;
11. the maximum Mendelev number;
12. the minimum electronegativity;
13. the range of the covalent radius;
14. the minimum Mendelev number.

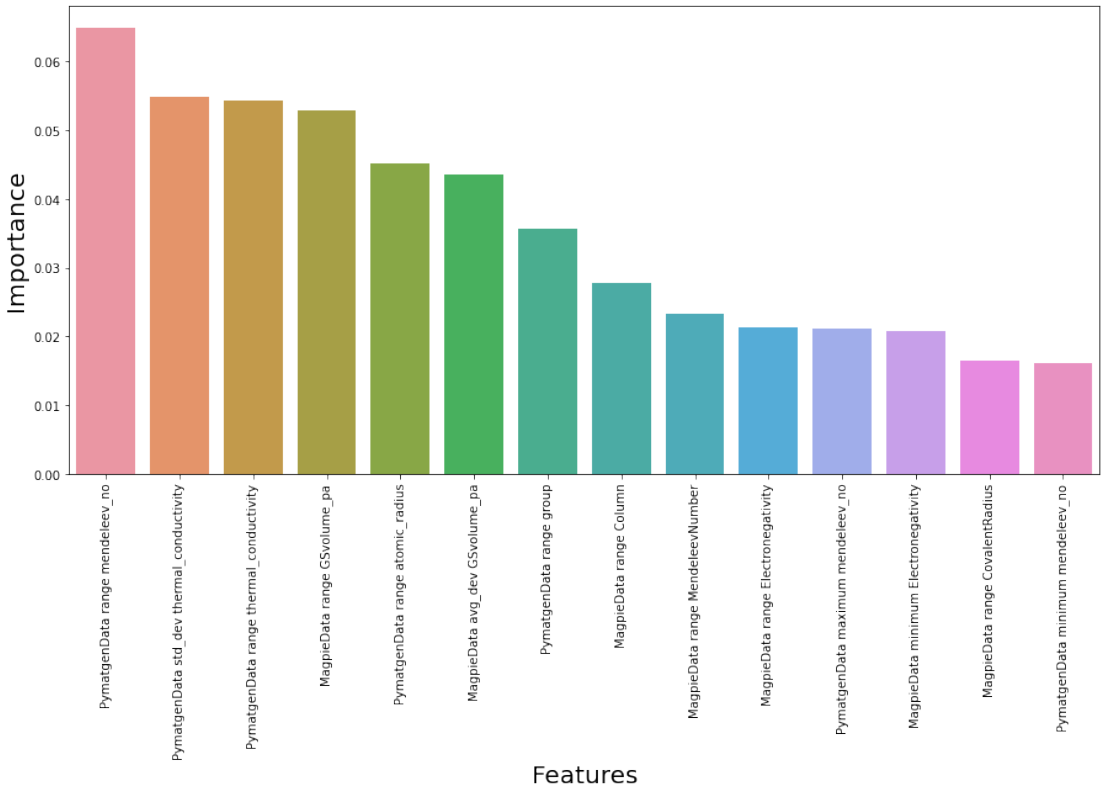

Figure 5.10: The important feature to the classification task

5.5 Final Model

Putting together the resulst of all the analyses of the previous sections, I built a final Random Forest Classifier. It takes into account the optimal temperature

threshold, the best number of decision trees, the best number of features and the most relevant features. The model is then trained on 70% of the data, tested on the other 30% and its performance is evaluated. It has a higher precision than the model trained in section **5.2.2** but performs similarly across the other metrics, indicating that the most important parameter to this classification problem is the way the data is categorized.

Table 5.3: Summarizing the performance of the final classification model

	Accuracy	Precision	Recall	F1
Performance	0.951	0.933	0.953	0.943

Chapter 6

Regression Model

The approach adopted to solving the critical temperature prediction problem in this chapter is regression. A first simple linear model is developed and its performance is compared to a random forest regression model developed afterwards. All the work is done on the dataset as it has been defined in the Methods chapter. The models based on the featurized data are also compared to models in the literature where the input consisted only of the chemical composition.
Unlike in the case of the classification model, the target variable for the regression is the natural logarithm of the critical temperature ($ln(T_c)$). As shown in Chapter 3, taking such a transformation gives a more uniform distribution. Moreover according to Gelman and Hill [12], the coefficients are more easy to interpret as they represent proportional differences.

Given the results obtained from the previous section, namely the identification of the optimal critical temperature at which to split the data into high- and low-temperature superconductors, I have developed two different models for each type of compound: one that predicts on compounds whose critical temperatures are higher or equal to 24K and one that predicts on those who critical temperatures are lower than 24K. I also train a regressor on the entire data without distinguishing between low- and high-temperatures.

6.1 Linear Regressor

The most simple type of regression is the linear one where the target variable can be expressed as a linear combination of explanatory variables. The major disadvantage of this method is the assumption that the relation between the variables is linear. It is nonetheless a good benchmark to measure up other regression models' performances to.

The first linear model is trained on compounds whose critical temperature is strictly smaller than 24K and the second one is trained on compounds whose critical temperature is greater or equal to 24K. Their performances are listed in Table 6.1.
The model for $T_c < 24$ has similar training and testing metrics indicating that it does not overfit. Its RMSEs (both testing and training) correspond to an error of about 2K, which is relatively small.
On the other hand, the model for $T_c \geq 24$ performs terribly. Its training metrics seem to indicate a better performance than model for $T_c < 24$, as it has a higher R^2 and a lower RMSE but the testing metrics indicate otherwise. Indeed, there is both a drastic decrease in R^2 and a severe increase of the RMSE. Performing a kfold cross validation ($k = 20$) on the model yields a mean RMSE of 181116.843, indicating the model is not appropriate for the problem at hand.

Table 6.1: Comparison of the performance of the linear regression model for low- and high-temperature

	Training R^2	Testing R^2	Training RMSE	Testing RMSE
Model $T_c < 24$	0.535	0.506	0.831	0.859
Model $T_c \geq 24$	0.608	-42934487980.3	0.289	95386.8

6.2 Random Forest Regressor

As the performance of the linear regressor left a lot to be desired, a new model needs to be developed. Similarly to what was done in the CLassification chapter, a Random Forest model was chosen. The benefits of the Random Forest regressor are similar to those of Random Forest classifier (ability to handle complex relationships between the variables, robustness to overfitting and flexibility in the types of input accepted).

As was previously done, three general models are developed: one for predicting low critical temperatures (< 24), one for predicting high critical temperatures, and one that does not split the data in two. The models are then refined and their performance compared.

6.2.1 Model for $T_c < 24$

A random forest regression model is trained to predict the critical temperatures below 24K. It performs better than the linear regression model that was developed on the same data in the previous section. We see on Table 6.2 that the random

forest regressor has higher R^2 values and lower errors than the linear regression model.

Table 6.2: Comparison of the performance of the random forest and linear regressors for $T_c < 24$

	Linear model	Random forest
Training RMSE	0.831	0.229
Testing RMSE	0.859	0.567
Validation RMSE	5.860	0.547
Training R^2	0.535	0.965
Testing R^2	0.506	0.784
Validation R^2	188.46	0.798

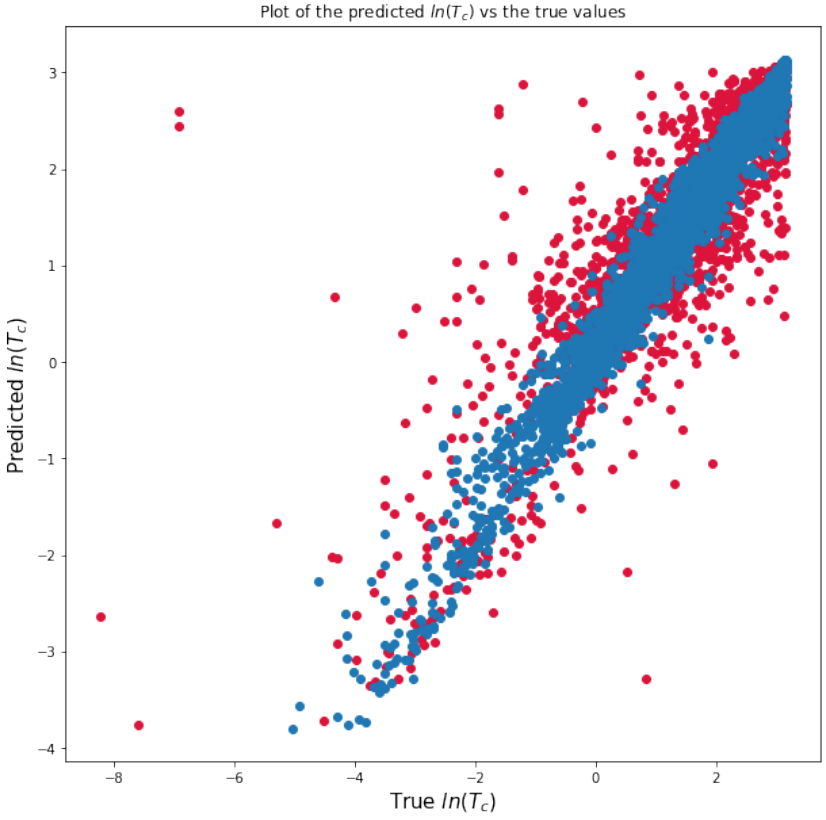

Figure 6.1: Scatter plot of the predicted $ln(T_c)$ vs the true $ln(T_c)$ where the red dots are from the testing set

We see on Figure 6.1 the predicted values of $ln(T_c)$ vs the true ones. The blue

dots are the ones derived from the training data and the red dots are derived from the testing one. The predictions made on the testing data exhibit some outliers but all the values predicted are positive with the smallest predicted temperature equal to 0.021 K.
Cross validation was performed (with the number of folds set at 10) and the score obtained was RMSE = 0.547, which confirms the established performance of the model.

6.2.2 Model for $T_c \geq 24$

A random forest regression model is trained to predict the critical temperatures greater than 24K. When the linear model was trained for this same task, it performed better in training than the model predicting the lower critical temperatures but showed severe signs of overfitting when the testing performance metrics were considered.

The random forest regression model on the other hand performs adequately on both datasets (training and testing). Indeed, while the training metrics are slightly better than the testing one, such a variation is not alarming.
Interestingly, the error of the random forest regressor on higher temperatures prediction is lower than that of the one on lower temperature prediction.

Table 6.3: Comparison of the performance of the random forest and linear regressors for $T_c \geq 24$

	Linear model	Random forest
Training RMSE	0.289	0.0840
Testing RMSE	95386.8	0.219
Validation RMSE	1.027e5	0.206
Training R^2	0.608	0.967
Testing R^2	-4.310e10	0.777
Validation R^2	4.958e11	0.800

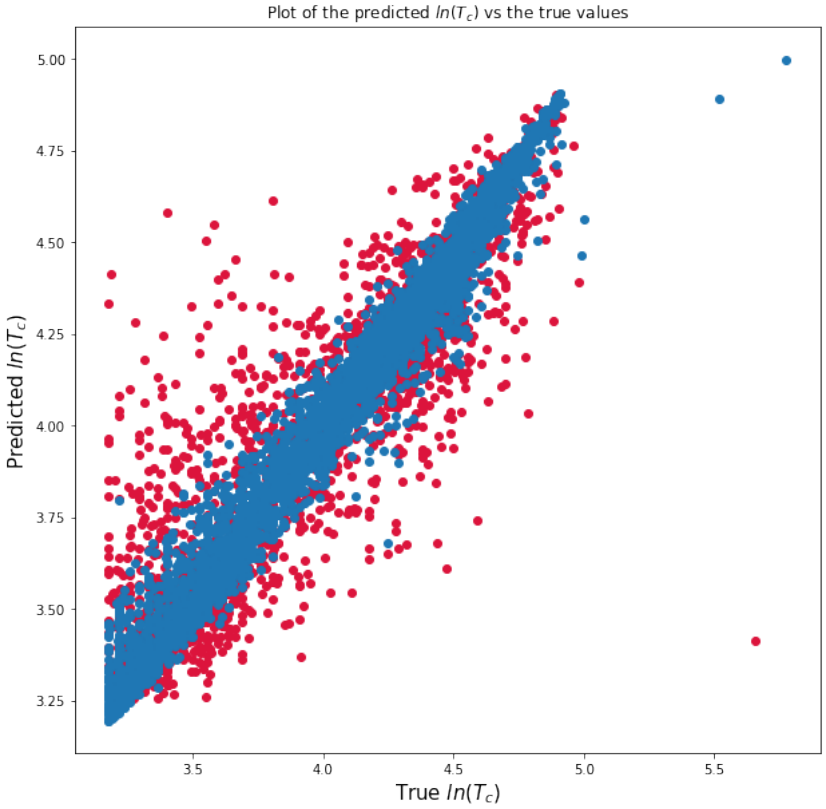

Figure 6.2: Scatter plot of the predicted $ln(T_c)$ vs the true $ln(T_c)$ where the red dots are from the testing set

We see on Figure 6.2 the predicted values of $ln(T_c)$ plotted against the true ones. The blue dots are the ones derived from the training data and the red ones are derived from the testing subset. There are a couple of outliers in both categories but overall the distributions are similar. And the error obtained on the cross validation is an RMSE of 0.206.

6.2.3 General Random Forest Regression Model

To see if there is any merit to splitting the data at the optimal threshold temperature, a Random Forest Regression model is trained on the data in its entirety. The performance metrics are then evaluated and shown on Table 6.4. This general model has a lower error than the others and a higher R^2 indicating that, with the set of features used, splitting the superconductors in two to try to predict their critical temperatures may not be the best method.

Table 6.4: The performance of the general random regressor

	Performance
Training RMSE	0.192
Testing RMSE	0.494
Validation RMSE	0.478
Training R^2	0.987
Testing R^2	0.902
Validation R^2	0.910

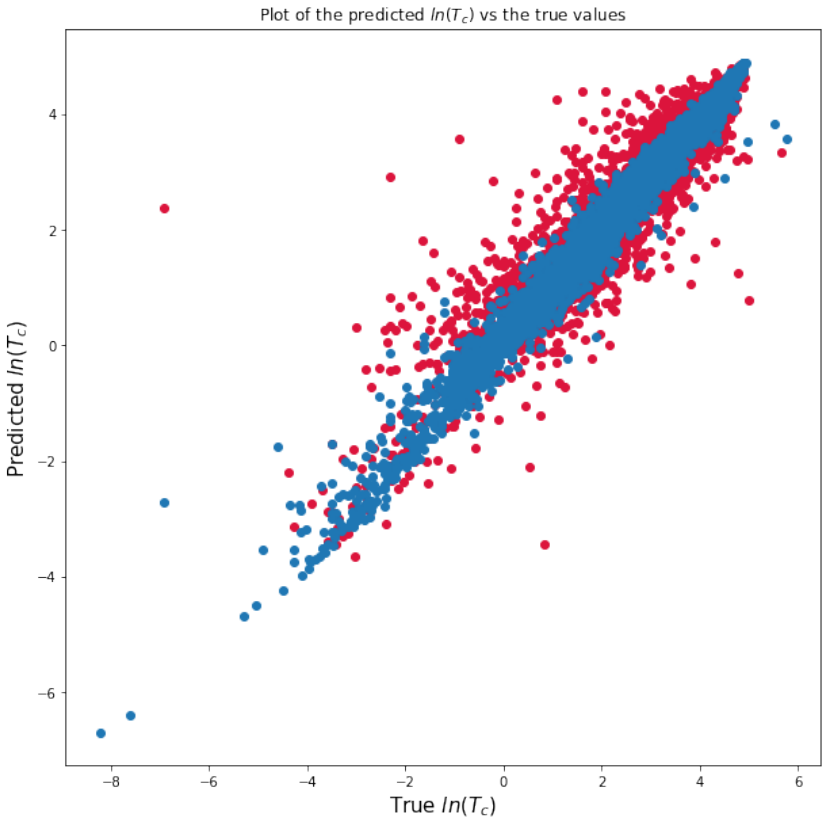

Figure 6.3: Scatter plot of the predicted $ln(T_c)$ vs the true $ln(T_c)$ where the red dots are from the testing set

We see on Figure 6.3 the predicted values of $ln(T_c)$ plotted against the true ones. The blue dots are the ones derived from the training set and the red ones were calculated on the testing set. The scatter plot looks very much like Figures 6.1 and 6.2 put together.

6.2.4 Feature Importance

Having trained the three random forest regressors and obtained satisfactory performance metrics, I thought it interesting to compare the relevant predictors for each task. We seen on figures 6.4, 6.5, and 6.6 the top 10 most significant features arranged in a bar plot for the low-temperature prediction, the high-temperature prediction and the general temperature prediction respectively. The information is also gathered in Table 6.4 to make the comparison easier.

Various statistics of the thermal conductivity are significant features in both temperature-based cases (the mean and minimum for model $T_c < 24$K and the standard deviation for model $T_c \geq 24$K). The same is true for the number of unfilled orbitals and the average atomic mass.

Some features appear more significant in one temperature-based model but not the other: the average deviation of the electronegativity is used in model $T_c \geq 24$K but not in Model $T_c < 24$K.

A noticeable difference among the important features of model $T_c \geq 24$ is the coefficient of the average number of unfilled orbitals. Such a variable appears to matter a lot more than the other in the prediction of high critical temperatures.
In terms of easy to interpret features, the general regression model performed the best. Its 10 most significant features are in line with what we understand of the theory of superconductivity.

Table 6.5: The 10 most relevant predictors for each regression task

Rank	**Model** $T_c < 24$	**Model** $T_c \geq 24$
1	Mean thermal conductivity (0.054)	Average number of unfilled orbitals (0.391)
2	Mode number (0.042)	Standard deviation of the thermal conductivity (0.087)
3	Mendelev number range (0.033)	Standard deviation of the group (0.028)
4	Mode of the atomic weight (0.028)	Mean block (0.027)
5	Mean atomic weight (0.027)	Standard deviation of the atomic mass (0.022)
6	Minimum thermal conductivity (0.024)	Standard deviation of the atomic radius (0.022)
7	Standard deviation of the group (0.022)	Average deviation of the covalent radius (0.018)
8	Mean atomic mass (0.019)	Average deviation of group space volume (0.015)
9	Mean number of unfilled orbitals (0.019)	Average deviation of the column (0.014)
10	Column average deviation (0.018)	Average deviation of electronegativity (0.0134)

Table 6.6: The 10 most relevant predictors for the general classification task

Rank	**General regression model**
1	Range of the thermal conductivity (0.229)
2	Mode of the number of unfilled orbitals (0.093)
3	Mode of the melting temperature (0.073)
4	Average deviation of the group space volume (0.053)
5	Range of the covalent radius (0.050)
6	Mean number of d valence electrons (0.028)
7	Mean atomic radius (0.018)
8	Average deviation of the average magnetic moment (0.018)
9	Average deviation of the electronegativity (0.016)
10	Standard deviation of the group (0.011)

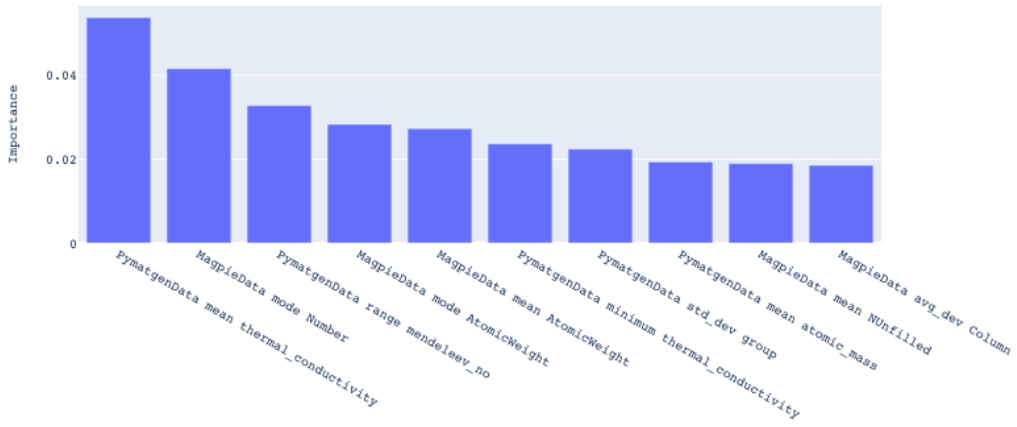

Figure 6.4: The most significant 10 features in the prediction of $T_c < 24K$

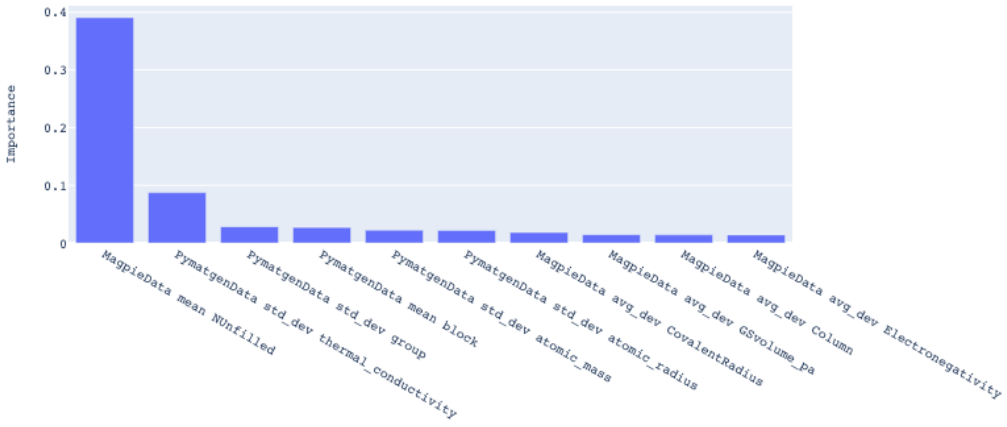

Figure 6.5: The most significant 10 features in the prediction of $T_c \geq 24K$

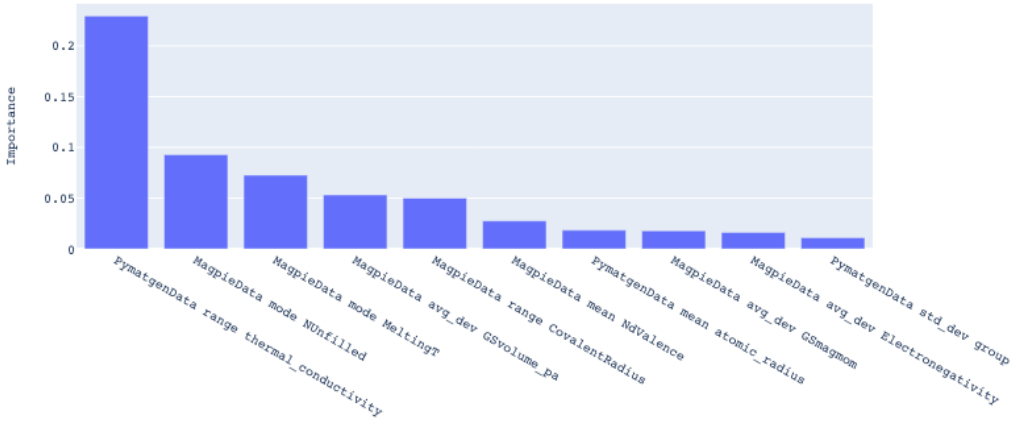

Figure 6.6: The 10 most significant features in the general prediction algorithm

6.3 Performance comparison

In an article published in 2020, Roter and Dordevic [13] employed a combination of supervised and unsupervised learning methods to predict the critical temperature of superconductors. The interesting about the article is that they did not use any physical descriptors in their models, only the chemical composition . Using only the chemical composition, they construct what they call a chemical composition matrix as seen on Table 6.5 where the elements are in the columns, the chemical formulas of the compounds are in the rows and the cells contain the proportion of the element in the compound. The final matrix they obtained was rather large (30 000 by 96) and was very sparse.

Table 6.7: Chemical composition matrix used by Roter and Dordeviv (2020

	H	He	Li	Be	$\cdots$	N
Compound 1	0.03	0	0	0	$\cdots$	0
Compound 2	0	0.2	0	0	$\cdots$	0.03
Compound 3	0	0	0.4	2	$\cdots$	0
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

They used a variant of the Random Forest algorithms where all features are considered at each node split in a tree: bagging trees [9]. The resulting model had

an R^2 score of 0.93 and an RMSE of 2.19 (corresponding to $\approx$ 8.19K). While it does have a higher R^2 score, it has a higher error than all models developed with the featurized data.
Put together, the overall best performing critical temperature prediction model is the one that was trained on the entire data.

Table 6.8: Comparison of the performance of the prediction algorithms (including the one developed by Roter and Dordevic [13])

	RMSE	R^2
Model with no features	2.19	0.93
Model with features ($T_c < 24$)	0.567	0.784
Model with features ($T_c \geq 24$)	0.219	0.777
Model with features (general)	0.478	0.910

Chapter 7

Identification of New Superconductors

The overall goal of using machine learning for materials science is not just to be able to predict the properties of a known compound, rather it is to be able to identify a compound exhibiting a certain property. Indeed, the traditional material discovery pipeline starts with identifying what properties one desire to obtain. Then based on scientific knowledge, come months (sometimes even years) of synthesizing prototypes and testing them. And there is no guarantee that the product obtain in the end is the one that was searched for all along. Which is why the idea of integrating machine learning methods to such a process is an attractive concept: instead of spending years and money on various prototypes, one could directly obtain the composition or the crystal structure of the goal material [14]. Such techniques are still in development, although there have been some positive results in narrowly defined problem such as the discovery of new stable compounds in the V-O systems [15].

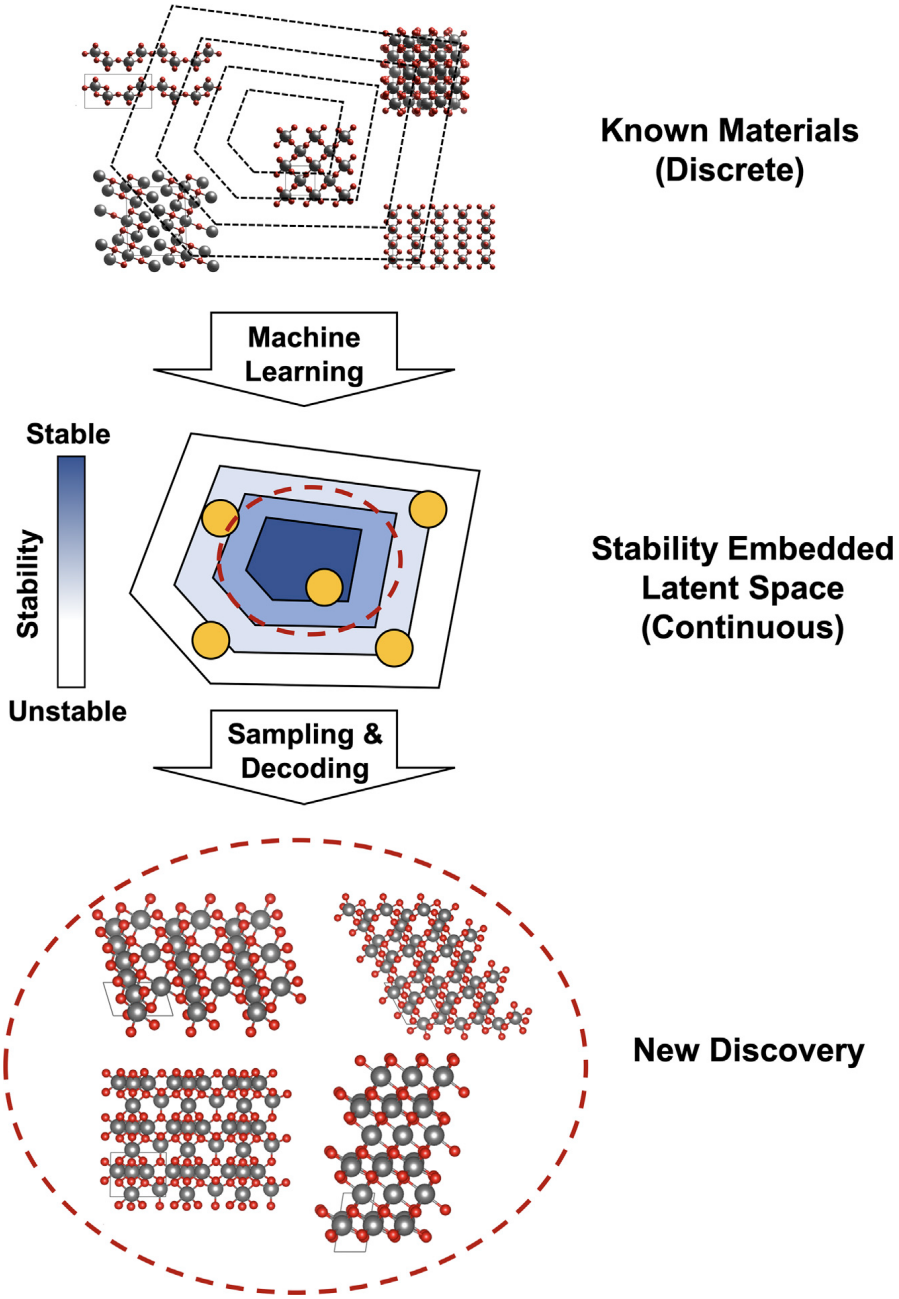

Figure 7.1: Inverse material design schematic flow as explained by Noh et al. (2019)

In this chapter, as the Random Forest algorithm preformed well on previous tasks, it is also used to predict whether a material is a superconductor or not. Data considered non-superconducting was imported from the Materials Project,

processed and the combined with superconducting data and classified using the Random Forest Classifier.

The Materials Project is an open-access database for materials research. Its founder, Kristin Persson, also heads the initiative at Berkley to run DFT calculations. As a result, the database is quite large, containing more than 130 000 inorganic compounds. Access to the database can be through their API or via the Matminer wrapper.

7.1 The Data Processing

The most useful information extracted from the unprocessed MDR SuperCon database was the chemical compositions and the associated critical temperatures. At this stage I cannot extract additional critical temperatures but I can obtain more chemical formulas. I downloaded all the chemical formulas of inorganic compounds available on the Materials Project database. The result was a list of 154 718 elements. Mirroring the data processing done at the beginning of this work, the chemical formulas were transformed into `pymatgen` objects to later make the featurization easier.

The main hypothesis here is that if a compound appears in the Materials Project database but not in the MDR SuperCon, it cannot be a superconductor. Therefore the superconductors are removed from the downloaded database, leaving a list of 152 077 non-superconducting compounds (there were 2 641).

To preemptively deal with the problem of unbalanced classes and taking into consideration the fact that columns and rows with missing values may need to be removed from the dataset, I randomly select a little over twice the number of superconductors (i.e. 34 000 observations) from the database of non-superconducting elements.

Using the Matminer toolkit, the resulting data is featurized using once again `ElementProperty` to obtain data about the chemical composition. I remove the columns made up of more than a quarter of missing values and then the rows with missing values. The final featurized dataset contains 33 958 observations and 179 variables.
The MDR SupreCon observations are labeled as 1 and the non-superconductors are labled as 0.

7.2 The Model

Supeconductors represent a proportion of only 10% of all the compounds listed in the Materials Project database. This is representative of reality as we understand it: there are a lot less superconducting compounds than they are non-superconducting ones. I chose not to represent such an extreme unbalance. Instead I first trained a Random Forest Regressor where a third of the data is labeled 1 and then another one where half is labeled 1. Their performances are presented and contrasted below.

7.2.1 Unbalanced Classes

The first Random Forest classifier was trained on the unbalanced dataset. Its performance as a function of the number of features included is shown in Figure 7.2 where we see that the accuracy, the precision, and the F1 score are high when the number of features is at 42.
Using that parameter, the model is trained and the training, testing and validation score are listed in Table 7.1. We might think that there are the indicators of a slight overfitting as the training RMSE is much smaller than the testing and validation errors, but these all correspond to values of about ≈ 1.15K.

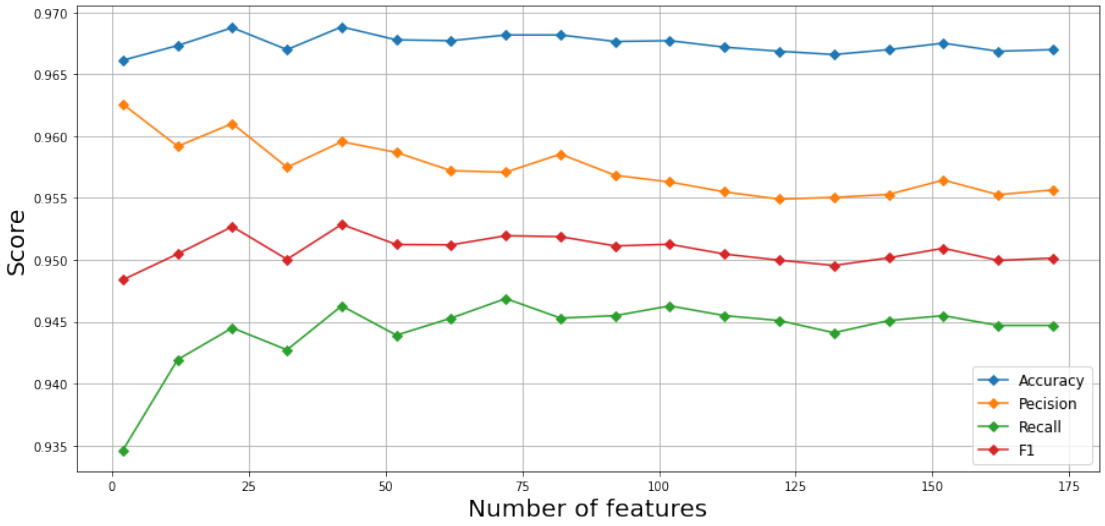

Figure 7.2: Performance of the classifier as a function of the number of features

Table 7.1: Performance of the classification on the unbalanced dataset

	Random Forest - Unbalanced
Training accuracy	0.999
Testing accuracy	0.966
Training precision	0.997
Testing precision	0.958
Testing recall	0.939
Training F1 score	0.998
Testing F1 score	0.949

The most relevant features to the model are displayed in Figure 7.3 and are the same ones identified in the Random Forest Regressor on all the critical temperatures, further cementing their importance to the superconductivity phenomenon.

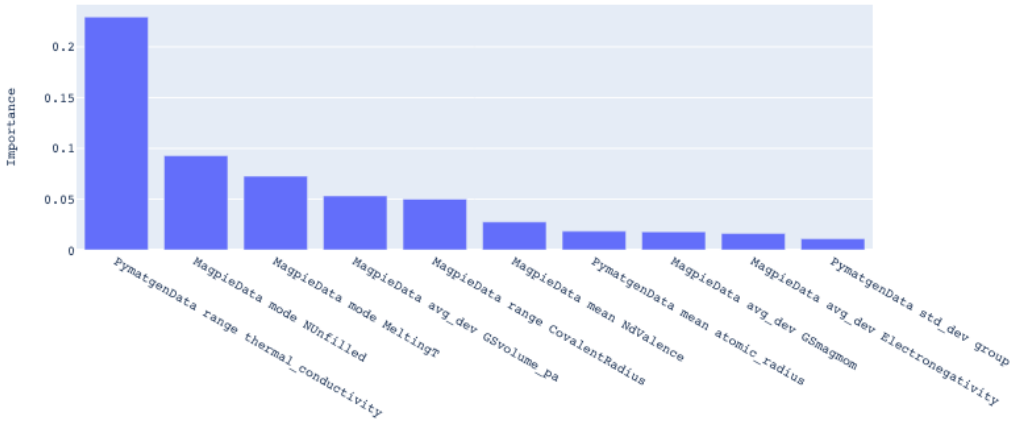

Figure 7.3: The 10 most significant features extracted by the classification algorithm trained on the unbalanced data

7.2.2 Balanced Classes

The second Random Forest classifier was trained on the balanced data set where the proportion of superconductors and non-superconductors is the same. The model performance can be measured using the metrics in Table 7.2 on which we see that the difference between the scores obtained on the balanced data and these is so small as to be negligible.

Table 7.2: Performance of the classification on the balanced dataset

	Random Forest - Balanced
Training accuracy	0.999
Testing accuracy	0.967
Training precision	0.927
Testing precision	0.963
Testing recall	0.9491
Training F1 score	0.938
Testing F1 score	0.932

Chapter 8

Conclusion

One of the most crucial components of machine learning is the data used to train the models. For the longest time, data was scarce in the materials science research field. Information about known compounds is not always homogeneous: one may for instance know the shear modulus of a compound because experiments have been conducted to measure but not of another. Luckily the development of modeling and simulation methods as well as the increase in computing power has made it possible to obtain large quantities of theoretical data about a large number of materials.

This has led to the creation of mostly open-source database containing theoretically calculated as well as experimental information about solid materials and to the creation of tools that enable almost anyone to use access them. The MDR Supercon is such a database and Matminer is such a tool.

The MDR Supercon is a large database with more than twenty thousand observations originally but with no other useful information than the chemical composition of the compounds and their critical temperatures. After rigorous processing of the data, the number of observations decreased to a little over seventeen thousand. Using the Matminer data mining tool, the information in the database was supplemented with statistics of element characteristics, yielding a database with over 180 variables. The data was processed yet again to deal with any missing information before it was deemed ready for use in supervised learning algorithms. All the algorithms used were of the Random Forest variety.

The problem I tried to solve is the one critical temperature prediction. First the problem was approached as a classification task. A threshold critical temperature was selected and the compounds were labeled based on whether their critical temperature falls below or above the threshold. The compounds were then classified as low-temperature or high-temperature. Various values for the threshold tempera-

ture were tried before one was found that yielded the best performing model: 24K. Afterwards other parameters of the model were tuned to maximize the performance The model trained on what was judged to be an optimal set of parameters had an accuracy of 0.951, a precision of 0.933, a recall score of 0.953, and an F1 score of 0.943.

Afterwards the problem was approached as a regression task. Instead of predicting the critical temperature directly, its logarithm was computed to obtain a more uniform distribution and it was used as the target variable. Using the optimal threshold critical temperature found in previous sections, the data was first split into two categories: the first category with the critical temperatures below 24K and the second category with the critical temperatures above 24K. A first linear model was developed, and while it performed adequately on the prediction of low-temperature superconductors (the RMSE was of 0.831 and the R^2 had a value of 0.535), it suffered from overfitting when trained on the high-temperature compounds (the performance metrics were much worse on the testing set).

The second regression model that was tried was the Random Regression one. It yielded an RMSE of 0.547 in the case of low critical temperature prediction and RMSE of 0.206 for the high critical temperature prediction. In this case, the model performed better for the higher temperatures than the lower ones.
The most significant features for each model were extracted, and similarly to the performance metrics, they differed for low-temperature superconductors and high-temperature ones.

The value of using more than just the chemical composition to predict the critical temperatures of compounds was judged by comparing a model developed by Roter and Dordevic [13] where the input was a matrix of element vectors, each containing the proportion of each element in the compound. The error for both models was lower but their R^2 was higher indicating that either approach is valid on its own. A way to improve on the various predictions would be to include the pressure measurements along with the critical temperature ones. A combination of both parameters could uncover interesting trends among the data.
Finally an attempt at identifying superconductors among non-superconductors was made. The chemical formulas of the compounds from the Materials Project database were retrieved and every one that does not match up with a compound in the MDR SuperCon was treated as a non-superconductor. A Random Forest classifier was trained on datasets where the proportion of the classes was varied. The models performed well on balanced and unbalanced data.

Bibliography

[1] Kieron Burke. *The ABC of DFT*. University of California, April 2007.

[2] Jonathan Schmidt, Mário R. G. Marques, Silvana Botti, and Miguel A. L. Marques. Recent advances and applications of machine learning in solid-state materials science. *npj Computational Materials*, 5(1):83, August 2019.

[3] Aron Walsh. The quest for new functionality. *Nature Chemistry*, 7(4):274–275, April 2015.

[4] W. Meißner, H. Franz, and H. Westerhoff. Messungen mit Hilfe von flüssigem Helium. XXII Widerstand von Metallen, Legierungen und Verbindungen. *Annalen der Physik*, 409(6):593–619, 1933.

[5] M. K. Wu, J. R. Ashburn, C. J. Torng, P. H. Hor, R. L. Meng, L. Gao, Z. J. Huang, Y. Q. Wang, and C. W. Chu. Superconductivity at 93 K in a new mixed-phase Y-Ba-Cu-O compound system at ambient pressure. *Physical Review Letters*, 58(9):908–910, March 1987.

[6] H. Oyama, T. Shinzato, K. Hayashi, K. Kitajima, T. Ariyoshi, and T. Sawai. Application of superconductors for automobiles. pages 22–26, 10 2008.

[7] Leon N. Cooper. Bound Electron Pairs in a Degenerate Fermi Gas. *Physical Review*, 104(4):1189–1190, November 1956.

[8] Logan Ward, Alexander Dunn, Alireza Faghaninia, Nils E.R. Zimmermann, Saurabh Bajaj, Qi Wang, Joseph Montoya, Jiming Chen, Kyle Bystrom, Maxwell Dylla, Kyle Chard, Mark Asta, Kristin A. Persson, G. Jeffrey Snyder, Ian Foster, and Anubhav Jain. Matminer: An open source toolkit for materials data mining. *Computational Materials Science*, 152:60–69, September 2018.

[9] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

[10] Philipp Probst and Anne-Laure Boulesteix. To tune or not to tune the number of trees in random forest? 2017. Publisher: arXiv Version Number: 1.

[11] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, second edition edition.

[12] Andrew Gelman and Jennifer Hill. *Data analysis using regression and multilevel/hierarchical models*. Analytical methods for social research. Cambridge University Press, Cambridge ; New York, 2007. OCLC: ocm67375137.

[13] B. Roter and S.V. Dordevic. Predicting new superconductors and their critical temperatures using machine learning. *Physica C: Superconductivity and its Applications*, 575:1353689, August 2020.

[14] Benjamin Sanchez-Lengeling and Alán Aspuru-Guzik. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365, July 2018.

[15] Juhwan Noh, Jaehoon Kim, Helge S. Stein, Benjamin Sanchez-Lengeling, John M. Gregoire, Alan Aspuru-Guzik, and Yousung Jung. Inverse Design of Solid-State Materials via a Continuous Representation. *Matter*, 1(5):1370–1384, November 2019.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**