T.R.

GEBZE TECHNICAL UNIVERSITY

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

MACHINE LEARNING GUIDED DISCOVERY OF HIGH ENTROPY CERAMICS

BARIŞ AYYILDIZ

SUPERVISOR
PROF. YUSUF SINAN AKGÜL

GEBZE
2023

T.R.
GEBZE TECHNICAL UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

MACHINE LEARNING GUIDED DISCOVERY OF HIGH ENTROPY CERAMICS

BARIŞ AYYILDIZ

SUPERVISOR
PROF. YUSUF SINAN AKGÜL

2023
GEBZE

GEBZE TECHNICAL UNIVERSITY	GRADUATION PROJECT JURY APPROVAL FORM

This study has been accepted as an Undergraduate Graduation Project in the Department of Computer Engineering on 16/06/2023 by the following jury.

JURY

Member (Supervisor)	:	Prof. Yusuf Sinan AKGÜL
Member	:	Prof. Yakup Genç

ABSTRACT

This research presents a deep learning model for predicting the composition of super-hard high entropy ceramics (HECs). Instead of relying on crystal structure information, a composition-based approach is employed, leveraging the broader availability and applicability of composition-based descriptors. The deep learning model is trained using hardness values, enabling high-throughput screening of HECs. Processing parameters are not included due to the lack of uniform reporting. The results demonstrate the effectiveness of the deep learning model in accurately predicting desirable HEC compositions with superior hardness properties, offering valuable insights for the design and discovery of advanced functional materials.

ACKNOWLEDGEMENT

For this important research idea and his lead I am grateful to my supervisor Prof. Yusuf Sinan Akgül. And I would like to thank Mr. Russlan Jaafreh from the Sungkyunkwan University for all of his support.

Barış Ayyıldız

LIST OF SYMBOLS AND ABBREVIATIONS

Symbol or Abbreviation		Explanation
HECs	:	High Entropy Ceramics
R2	:	R Squared
SSE	:	Sum of Squared Errors

CONTENTS

LIST OF FIGURES

1. INTRODUCTION

In this project, I applied advanced machine learning techniques to gain insights into high entropy ceramics (HECs) and predict the hardness factor based on a given composition. Through the utilization of machine learning models, I extracted valuable information to enhance our understanding of the properties and behavior of HECs.

1.1. Success Criteria

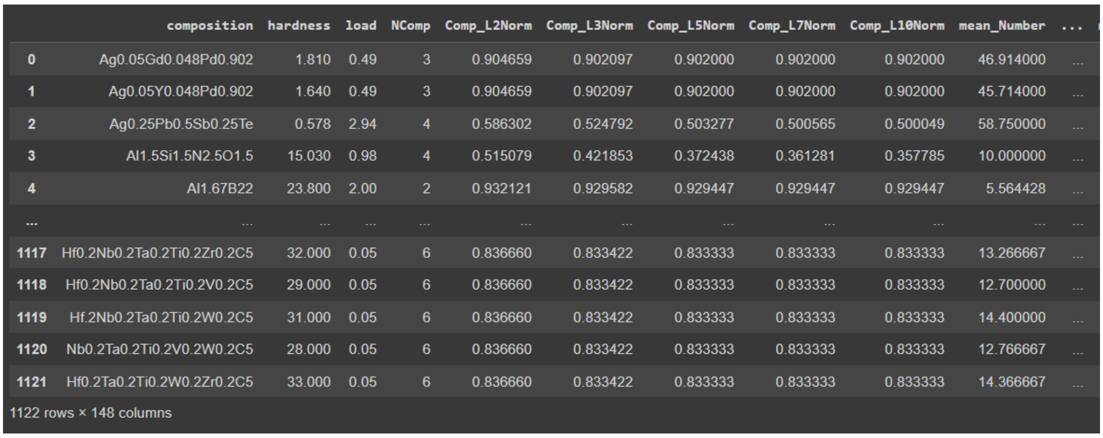
1. Collecting at least 1000 data with mininum 20 features
2. Getting 80% of accuracy in regression model
3. Estimating hardness factor for new compositions under 30ms on Google Colab using the Intel Xeon Processer 2.20 GHz

2. DATA PREPROCESSING

2.1. Dataset

I was provided with a dataset along with a paper, and you can find the link to access it in the Appendices section. I was provided with a dataset along with a paper, and you can find the link to access it in the Appendices section.

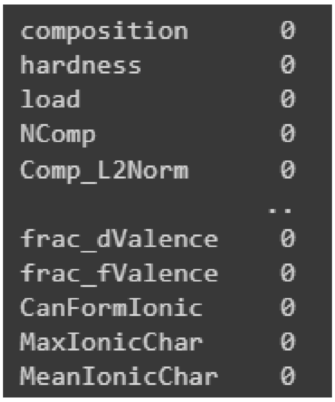
The dataset used for training my models contains 1122 rows and 148 columns. One of the columns is named ”composition,” which represents the name of the composition in each row. Another column is named ”hardness,” which serves as the output variable that I aim to predict using my machine learning models.



	composition	hardness	load	NComp	Comp_L2Norm	Comp_L3Norm	Comp_L5Norm	Comp_L7Norm	Comp_L10Norm	mean_Number	...
0	Ag0.05Gd0.048Pd0.902	1.810	0.49	3	0.904659	0.902097	0.902000	0.902000	0.902000	46.914000	...
1	Ag0.05Y0.048Pd0.902	1.640	0.49	3	0.904659	0.902097	0.902000	0.902000	0.902000	45.714000	...
2	Ag0.25Pb0.5Sb0.25Te	0.578	2.94	4	0.586302	0.524792	0.503277	0.500565	0.500049	58.750000	...
3	Al1.5Si1.5N2.5O1.5	15.030	0.98	4	0.515079	0.421853	0.372438	0.361281	0.357785	10.000000	...
4	Al1.67B22	23.800	2.00	2	0.932121	0.929582	0.929447	0.929447	0.929447	5.564428	...
...	...	...	...	...	...	...	...	...	...	...	...
1117	Hf0.2Nb0.2Ta0.2Ti0.2Zr0.2C5	32.000	0.05	6	0.836660	0.833422	0.833333	0.833333	0.833333	13.266667	...
1118	Hf0.2Nb0.2Ta0.2Ti0.2V0.2C5	29.000	0.05	6	0.836660	0.833422	0.833333	0.833333	0.833333	12.700000	...
1119	Hf.2Nb0.2Ta0.2Ti0.2W0.2C5	31.000	0.05	6	0.836660	0.833422	0.833333	0.833333	0.833333	14.400000	...
1120	Nb0.2Ta0.2Ti0.2V0.2W0.2C5	28.000	0.05	6	0.836660	0.833422	0.833333	0.833333	0.833333	12.766667	...
1121	Hf0.2Ta0.2Ti0.2W0.2Zr0.2C5	33.000	0.05	6	0.836660	0.833422	0.833333	0.833333	0.833333	14.366667	...

1122 rows × 148 columns

Figure 2.1: Dataset



```
composition      0
hardness         0
load             0
NComp            0
Comp_L2Norm      0
                ..
frac_dValence    0
frac_fValence    0
CanFormIonic     0
MaxIonicChar     0
MeanIonicChar    0
```

Figure 2.2: Dataset Missing Values

2.2. Preprocessing

Next, I explored the dataset to identify the input features that are most relevant for predicting hardness. To achieve this, I performed a correlation analysis between the input features and the target variable. I considered features with a correlation coefficient greater than 0.5 or less than -0.5 as potentially influential for hardness prediction.

To gain further insights into the relationships among the selected features, I generated a correlation matrix and visualized it using a heatmap. This visualization helped me identify patterns and potential interactions between the features, which assisted in the selection of informative input features for my models.

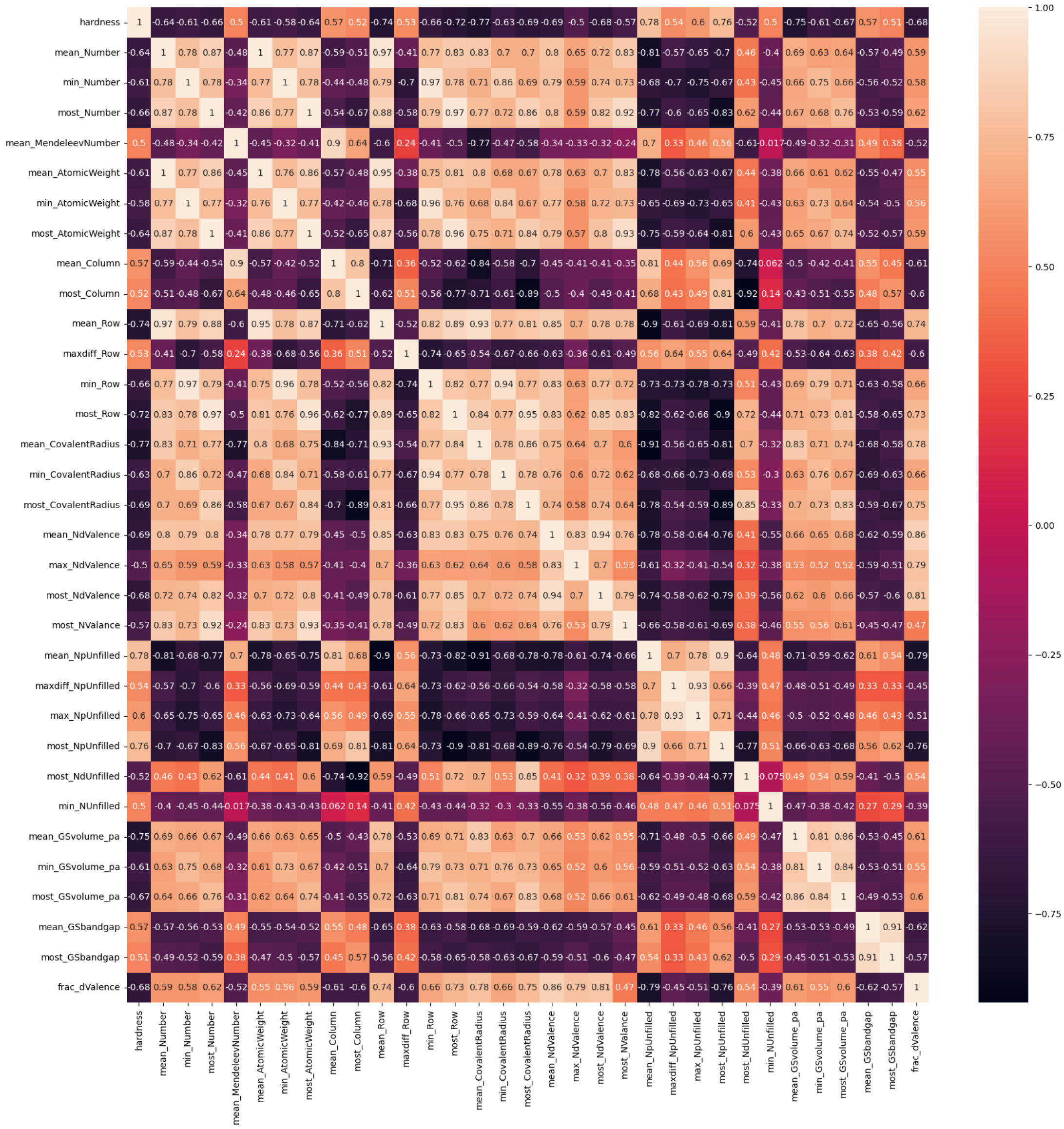


Figure 2.3: Correlation Heatmap.

For effective model training, I applied preprocessing techniques to scale both the input features and the target variable. I employed the Min-Max scaling technique to scale the input features, which transformed their values into a specific range. This scaling process ensures that features with different scales do not unduly influence the predictions made by the models. Additionally, I also scaled the target variable to align with the input feature scaling.

Finally, I split the dataset into training and testing sets using a standard train-test split approach. This division allowed me to train the models on a subset of the data and evaluate their performance on unseen data, providing an unbiased assessment.

By following these preprocessing steps, I appropriately prepared the dataset for training and evaluating my machine learning models.

2.3. Working Environment

I primarily used Google Colab connected to Google machines as my working environment. I used the open source libraries Scikit and Keras in Python to build my model.

3. MACHINE LEARNING MODELS

I used three different regression models to train on my dataset and they are Linear Regression, Decision Tree Regressor and Random Forest Regressor.

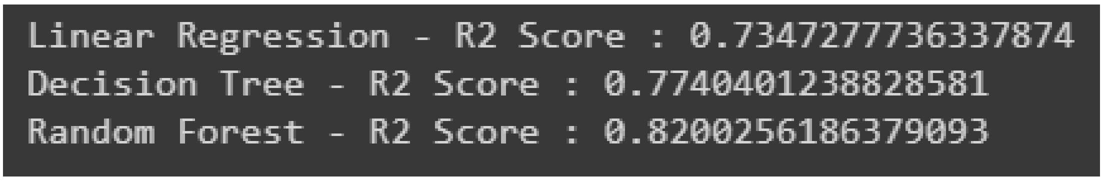
When it comes to Linear Regression, I achieved an R2 score of 0.73, indicating that approximately 73% of the variance in hardness can be explained by the selected features. It's a simple yet effective model that fits a straight line to the training data. While it performed reasonably well, I wanted to explore more powerful models.

The Decision Tree Regressor, on the other hand, produced an R2 score of 0.77.

However, the model that outperformed the others was the Random Forest Regressor, with an impressive R2 score of 0.82. This model combines multiple decision trees to make predictions, leveraging their collective knowledge.

These R2 scores offer valuable insights into the performance of each model and their ability to capture the relationships within the data. The Decision Tree and Random Forest models, with their higher R2 scores, demonstrate stronger predictive capabilities compared to Linear Regression.

To summarize, my machine learning models for hardness prediction displayed varying levels of performance. The Random Forest model stood out as the most successful, surpassing both Linear Regression and the Decision Tree model in terms of predictive accuracy.



```
Linear Regression - R2 Score : 0.7347277736337874
Decision Tree - R2 Score : 0.7740401238828581
Random Forest - R2 Score : 0.8200256186379093
```

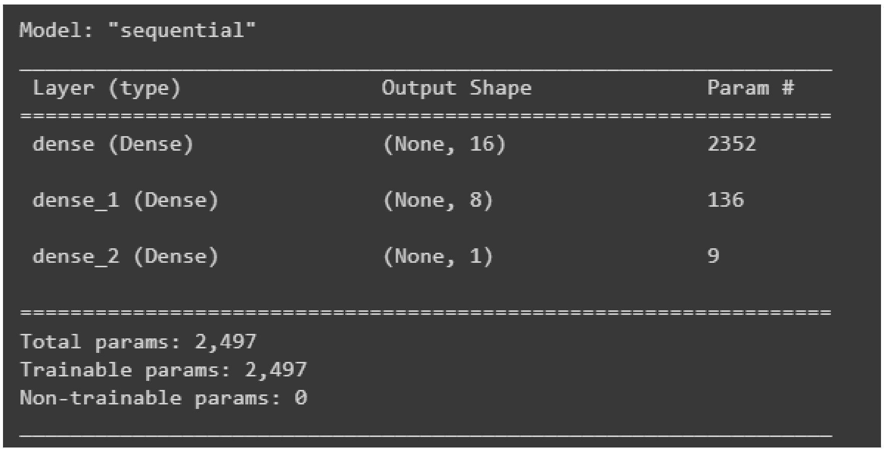
Figure 3.1: Regression Models Results

4. DEEP LEARNING MODEL

I also built a deep learning model to see if i can get better R2 score.

For my deep learning model I used 3 fully connected dense layers. First layer accepts 146 inputs and generates 16 outputs. The second layers takes these 16 outputs as input and generates 8 outputs. And the final layers takes those 8 numbers and produces a single number, which is the predicted value for the hardness factor. In total there were 2497 parameters to train

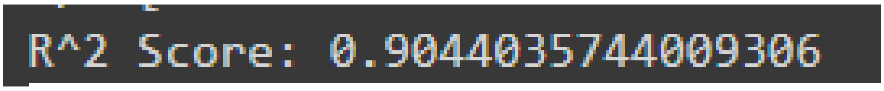
Here is the architecture of my model:



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 16)                2352

 dense_1 (Dense)             (None, 8)                 136

 dense_2 (Dense)             (None, 1)                 9

=================================================================
Total params: 2,497
Trainable params: 2,497
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.1: Neural Network Model Architecture.

I have trained this model in 400 epoch with the batch size 32, during the training validation loss has dropped from 100's to 12's. And it gave R2 score over 90%



```
R^2 Score: 0.9044035744009306
```

Figure 4.2: Neural Network Model Result.

To gain insights into the learned representations and the relationships between the selected features, I employed t-distributed Stochastic Neighbor Embedding (t-SNE). t-SNE is a popular technique for dimensionality reduction and visualization, particularly effective in revealing clusters and patterns within high-dimensional data. By applying t-SNE to the extracted features from the last layer of the deep learning model, I could visualize the data points in a lower-dimensional space, aiding in the understanding of their distribution and potential groupings.

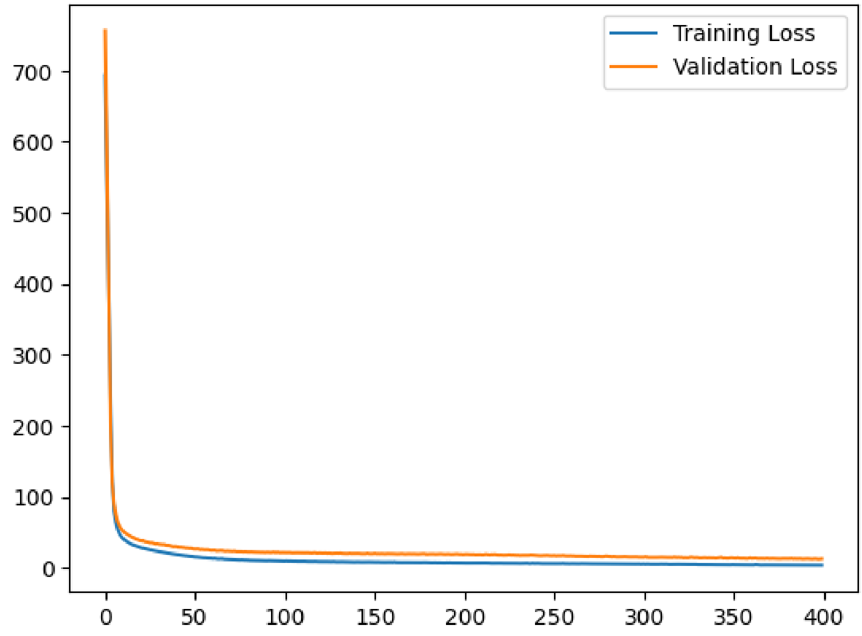


Figure 4.3: Training and Validation Loss Functions.

Here you can see some of the compositions on this 2d map. You can see that the compositions that have similar elements in it are close with each other.

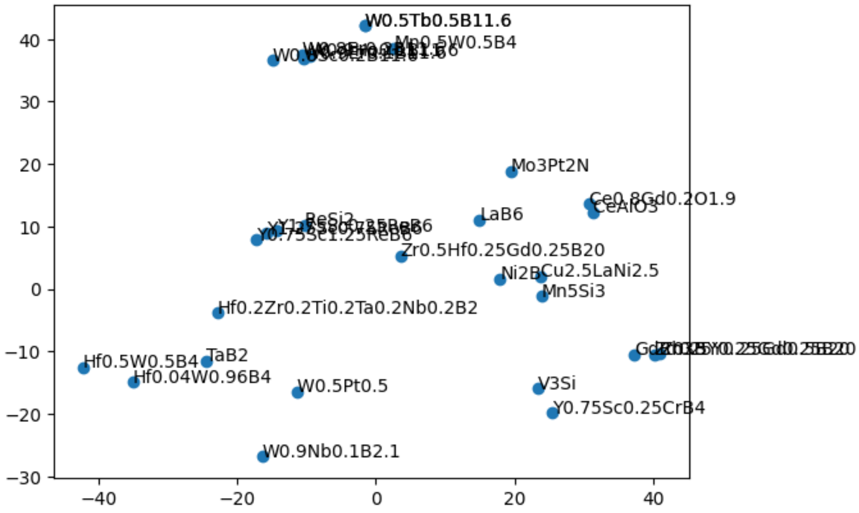


Figure 4.4: Dimensionality Reduction.

Moreover, to investigate the interrelationships among the extracted features, I computed a correlation matrix. By extracting the activations from the last layer of the deep learning model for a subset of eight features, I constructed a matrix that quantifies the pairwise correlations between these features. Visualizing this correlation matrix in the form of a heatmap offered valuable insights into the dependencies and interactions between the features, contributing to the interpretation of the model's predictions.

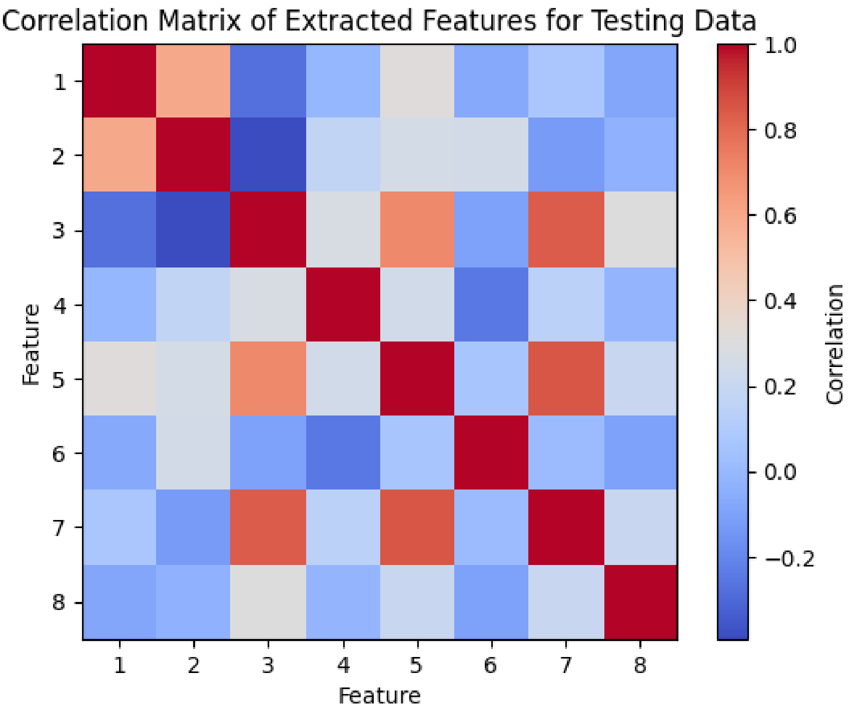


Figure 4.5: Extracted Features Correlation Matrix.

In addition to these visualization techniques, I also employed clustering algorithms to further explore patterns and groupings within the data. By applying clustering algorithms such as K-means or hierarchical clustering to the extracted features, I aimed to identify distinct clusters or subgroups of samples that share similar characteristics.

To select the number of clusters, I have used the ”Elbow Method”. The Elbow Method is a technique used to determine the optimal number of clusters in a dataset when performing clustering analysis, such as K-means clustering. It is named after the shape of the plot resembling an elbow.

By looking at this graph, I chose the number 3 as my cluster number. Because after that point SSE doesn't decrease that much.

And I listed all the compositions with their corresponding cluster assignments.

Table 4.1: Material Composition and Clusters

Cluster 0	Cluster 1	Cluster 2
Cr2B	W0.9Ta0.1B2.1	Zr0.25Hf0.5Y0.25B20
TiMn2	Hf0.25Zr0.25Ti0.25Ta0.25B2	Y0.75Zr0.25B12
La3Te4	Hf0.04W0.96B4	NbC
NdMn2Si2	Zr0.5Ta0.5N	Y0.75Zr0.25B12
Hf0.1Pt0.9	Os0.5Ru0.5B2	Sc0.25Zr0.75B12
ThC	YRe0.95Cr0.05B4	W0.8Er0.2B11.6
Re0.9Si0.1	W0.93Ta0.02Cr0.05B4	MoB
SmMn2Si2	Y0.5Sc1.5ReB6	Zr0.5Hf0.25Gd0.25B20
Hf3Ge	Zr0.2W0.8B4	TaC
Cr11Ge19	Ti0.2Ta0.2Cr0.2Mo0.2W0.2B2	Y0.25Zr0.75B12
Cd3Gd	Y0.05Sc1.95ReB6	W0.8Gd0.2B11.6
Y0.1Ce0.8Sm0.1O1.9	Y2Re0.5Cr0.5B6	Zr0.25Hf0.25Y0.5B20
Ru0.5Pt0.5	Ti0.08W0.92B4	DyB12
V6Ga5	Cr2.4W0.6B4	Zr0.25Hf0.5Gd0.25B20
Nb5Ge3	YRe0.75Cr0.25B4	BaHfO3
V0.25Re0.75	Cr0.05W0.95B4	Yb0.2Zr0.8O1.92
TaCr2	Y1.75Sc0.25ReB6	W0.8Sc0.2B11.6
Ce0.8Nd0.2O1.9	Re0.6W0.4B2	CrB
Ru0.5Ir0.5Al	Mo0.03W0.97B4	Y0.75Sc1.25ReB6
Re2Si	Hf0.2Zr0.2Ti0.2Mo0.2W0.2B2	Y0.5Hf0.25Gd0.25B20
V1.16Re2.84	Zr0.08W0.92B4	CeGaO3
Y5Si4	Ti0.3W0.7B4	Sc0.5Y0.5B12
HfW2	YReB4	VC
NbCr2	YRe0.5Cr0.5B4	Zr0.5Gd0.5B12
Cr4Al9	Mn0.3W0.7B4	ZrO2
CrGe	W0.5Ta0.5B2.1	TiB2
TiGe2	Os0.3Ru0.7B2	W0.8Dy0.2B11.6
NdC2	WB4	Y0.25Sc0.75CrB4
CeRh3B	OsB2	Zr0.5Hf0.5B12
V3Ge	Ta0.25W0.75B	Zr0.1W0.9B4
SrVO3	YRe0.75Cr0.25B4	Ti0.2Zr0.2Nb0.2Ta0.2Mo0.2C
Si	WB2.1	Ta0.01W0.99B
NbB	WB2.1	Y2Re0.75Cr0.25B6
V0.8Re0.2	Hf0.5W0.5B4	Hf0.2Zr0.2Ti0.2Ta0.2W0.2B2
Fe0.85Pt0.15	YReB4	YRe0.5Cr0.5B4
Nb3Sn0.5Ge0.5	W0.5Nb0.5B2.1	W2B
NbB	Y0.25Sc1.75ReB6	BN
Fe0.9Ru0.1	Ti0.08W0.92B4	Y1.75Sc0.25ReB6
Re0.25Pt0.75	YRe0.25Cr0.75B4	WB2.1
ZrMo2	Y0.25Sc1.75ReB6	Hf0.25Zr0.25Ti0.25Ta0.25B2
InBi0.024Sb0.976	W0.5Ta0.5B	Hf0.2Ta0.2Ti0.2W0.2Zr0.2C5
Cr0.43Re0.43Co0.14	WB	Y2Re0.95Cr0.05B6
InN	Cr0.5W0.5B4	W0.9Nb0.1B2.1
EuB6	Hf0.2Mo0.2Ta0.2Nb0.2Ti0.2B2	Ti0.1W0.9B4
UCo5	Cr0.3W0.7B4	Hf0.3W0.7B4
Rh0.5Pt0.5	YRe0.75Cr0.25B4	Ta0.04W0.96B4
ZnTe	Y0.5Sc1.5ReB6	YReB4
FeMn	Hf0.25Zr0.25Ti0.25Ta0.25B2	CrB
NbB	YRe0.5Cr0.5B4	TiB2

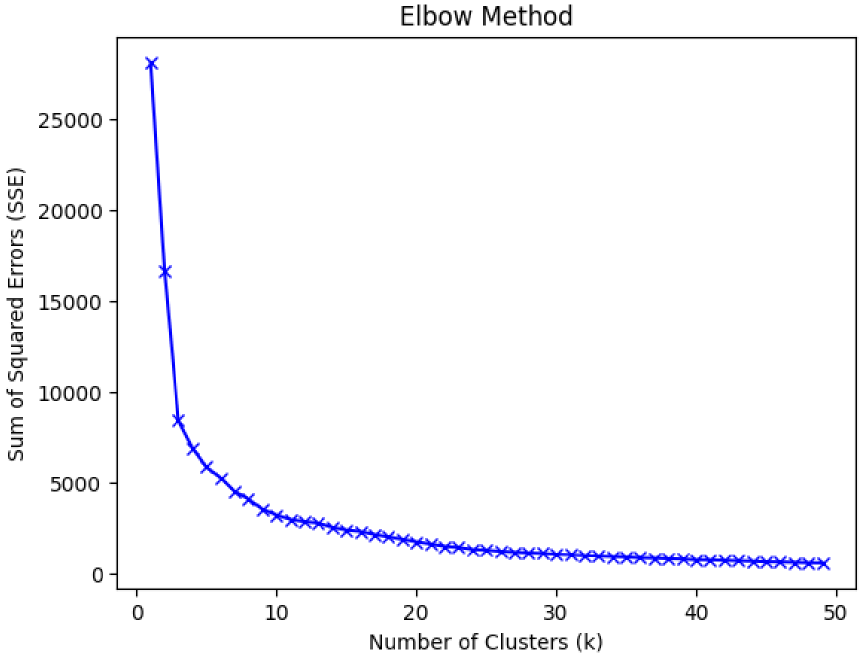


Figure 4.6: Elbow Method.

5. CONCLUSIONS

In conclusion, I embarked on a data-driven exploration to predict the hardness of compositions using machine learning and deep learning techniques. The dataset was thoroughly analyzed, and relevant input features were identified based on correlation analysis. Preprocessing steps, such as feature scaling and train-test splitting, were employed to ensure the dataset's suitability for model training and evaluation.

Multiple machine learning models, including Linear Regression, Decision Tree, and Random Forest, were trained and evaluated. The R2 scores obtained indicated that the Random Forest model outperformed the other models, demonstrating its effectiveness in predicting hardness.

Furthermore, a deep learning model, specifically a Multilayer Perceptron, was developed and trained on the dataset. The model achieved satisfactory results, showcasing the potential of deep learning for hardness prediction.

To gain insights into the relationships among the input features, a correlation matrix was generated by extracting the last layer of the deep learning model. This visualization provided valuable information about the interplay between features and helped identify influential factors affecting hardness.

Additionally, a dimensionality reduction technique called t-SNE was utilized to visualize the compositions in a 2D map. However, the scatter plot of the clustered compositions revealed that the clustering algorithm did not effectively separate compositions with distinct characteristics.

5.1. Evaluation of success criteria

1. I have collected over 1000 labeled data to train my models

2. The system was able to achieve over 90% success in regression.

3. Estimating the hardness factor of a composition takes less than 2 milliseconds

I can proudly say that I fulfilled all the success criteria I aimed for.

APPENDICES

- https://www.sciencedirect.com/science/article/abs/pii/S0167577X21015962
- https://ieeexplore.ieee.org/abstract/document/6310529