DSE-301 Final Project

Nirnay Roy(16128)

Instructor: Dr. Parthibhan Srinivasan

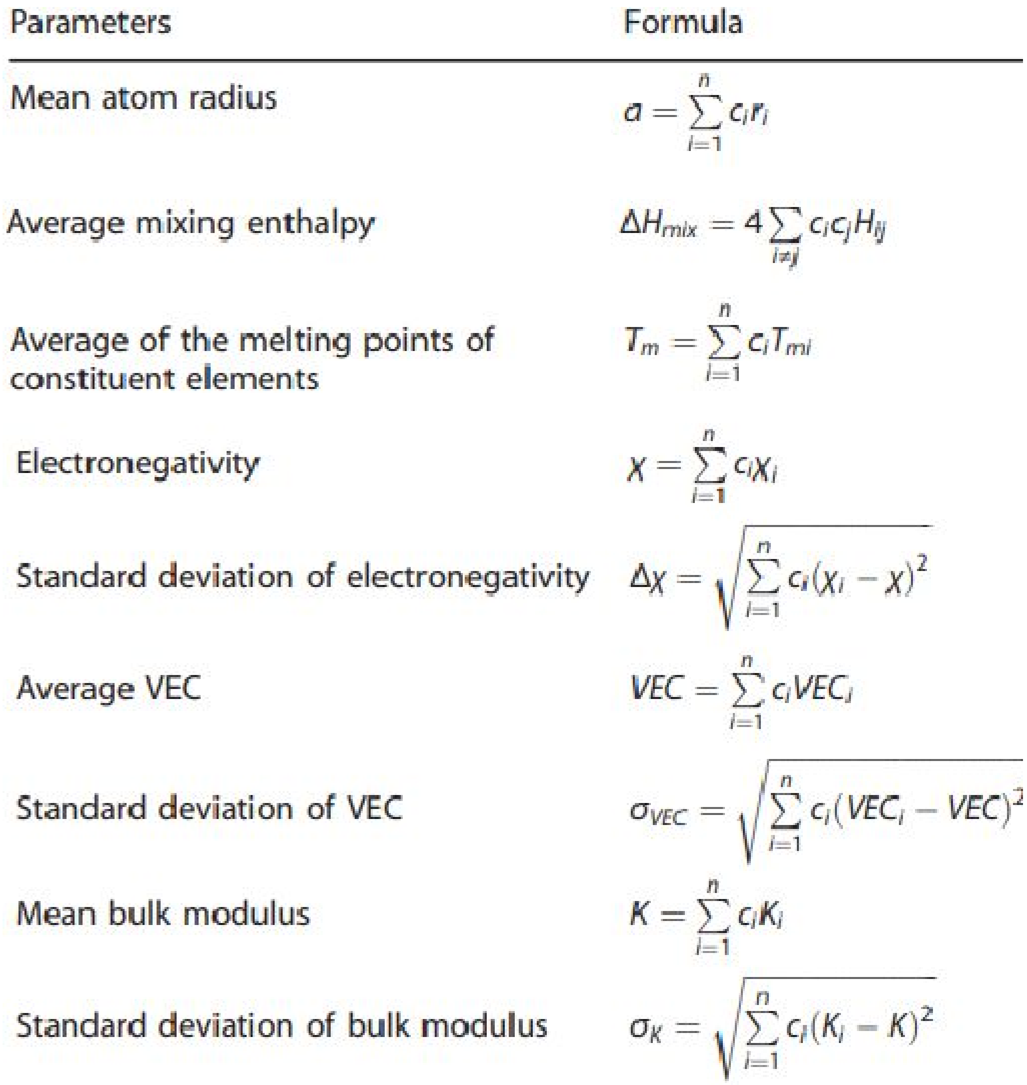
15 December 2020

Phase Prediction of High Entropy Alloys using Machine Learning

High Entropy Alloys(HEAs) are interesting because of useful properties like high hardness and ductility, high-temperature strength, antioxidant capacity and wear-resistance, which are absent from traditional metal alloys. These properties depend on the phase the HEAs take. HEAs can take 3 phases namely Intermetallic Compound(IM), Solid Solution(SS) and Amorphous Mixture(AM). Mixed states of IM+SS and IM+AM are also formed for some alloys. Phase prediction was done using calculations from DFT(Density Functional Theory) before. Recently, it was shown that machine learning can be used to predict phases of alloys using properties from the constituent metals like average Valence electron concentration(VEC), Pauling electronegativity, atomic size difference, mixing entropy and mixing enthalpy. In this project we analyse the data given by [Miracle and Senkov](https://www.sciencedirect.com/science/article/pii/S1359645416306759) in their review article of HEAs.

Preparing the dataset

The dataset for this analysis was forked from ZHOU-Ziqing/MLcode-for-HEAphase. The dataset has 603 HEAs and 9 features for each of them. The parameters with their corresponding formulae are given below.



Parameters	Formula
Mean atom radius	$a = \sum_{i=1}^{n} c_i r_i$
Average mixing enthalpy	$\Delta H_{mix} = 4 \sum_{i \neq j} c_i c_j H_{ij}$
Average of the melting points of constituent elements	$T_m = \sum_{i=1}^{n} c_i T_{mi}$
Electronegativity	$\chi = \sum_{i=1}^{n} c_i \chi_i$
Standard deviation of electronegativity	$\Delta \chi = \sqrt{\sum_{i=1}^{n} c_i (\chi_i - \chi)^2}$
Average VEC	$VEC = \sum_{i=1}^{n} c_i VEC_i$
Standard deviation of VEC	$\sigma_{VEC} = \sqrt{\sum_{i=1}^{n} c_i (VEC_i - VEC)^2}$
Mean bulk modulus	$K = \sum_{i=1}^{n} c_i K_i$
Standard deviation of bulk modulus	$\sigma_K = \sqrt{\sum_{i=1}^{n} c_i (K_i - K)^2}$

-Two Alloys had Nan and infinite values. We drop them from our data

Anomaly Detection

We use Anomaly Detection to find Outlier Alloys. Such Alloys may be of specific interest due to their rare properties. We use Univariate and Multivariate Anomaly Detection here.

Interquartile Range(Univariate)

Here we consider Alloys with features less or more than 1.5 times the Interquartile range of that particular feature. Interquartile range is the amount of spread in the middle 50% of a dataset.

Parameter	No. of outliers	Outlier Alloys
Mean Atomic Radius(a)	18	'AlB12', 'B4Co', 'AlB2', 'B6Co2Nb2', 'Sr36Al24Co20Y20', 'GdTbDyTmLu', 'Ca6.23Mg3.78Sn7', 'SrCaYbMgZn0.5Cu0.5', 'SrCaYbLi0.55Mg0.45Zn', 'Sr40Al20Co20Y20', 'SrCaYbMgZn', 'YGdTbDyLu', 'HoDyYGdTb', 'Sr46Al14Co20Y21', 'Ca65Mg15Zn20', 'GdHoLaTbY', 'Al3Ca8'
Average Melting Temperature(Tm)	3	'B2CoW2', 'MoNbTaVW', 'NbMoTaW', 'HfW2'
Average Mixing Enthalpy(Hmix)	4	'SmFe6Ti6N', 'Zr17Ta16Ti19Nb22Si26', 'AlMoNbSiTaTiVZr'
Electronegativity(E)	27	'Au46Ag5Cu29Si20', 'Au52Pd2.3Cu29.2Si16.5', 'AuCu', 'Pd95Si5', 'AuSb2', 'AgPt3', 'Au2Bi', 'Au9Sn', 'Au3In2', 'Al4Li9', 'AuSn', 'Au2Pb', 'AuPb2', 'Bi3Pb7', 'AuPb3', 'GdTbDyTmLu', 'SrCaYbMgZn0.5Cu0.5', 'SrCaYbLi0.55Mg0.45Zn', 'SrCaYbMgZn', 'YGdTbDyLu', 'HoDyYGdTb', 'Ca65Mg15Zn20', 'GdHoLaTbY', 'Al3Ca8'
sd of Electronegativity(dE)	2	'CuGeLi2', 'HfW2'
Average VEC	0	
sd of VEC	1	'Cd5Li4Mg'
Bulk modulus	8	'AlB12', 'B4Co', 'B6Co2Nb2', 'B2NiTa', 'B2Mo2Ni', 'B2CoMo2', 'B2CoW2', 'AlRe2'
sd of Bulk Modulus	10	'AlB2', 'HfB2', 'B4Fe4Nd', 'B3FeNd2', 'B2Co3Zr', 'BCaNi4', 'AuSb2', 'AlRe2', 'HfW2', 'Au2Bi', 'Au3In2'

On the basis of the above table, we can interpret the following:

- The Mean Atomic Radius of an alloy takes on extreme values when the alloy either has Boron or Aluminium, else there are large atoms in the constituent metals.
- The Average Mixing Enthalpy takes large values in alloys of 4 or more metals.
- Electronegativity values are extreme for 27 alloys depending on the constituents.
- The bulk modulus takes extreme values in alloys containing Aluminium, Boron, Silver and Tungsten.

Local Outlier Factor (Multivariate)

The LOF calculates the outlier score based on local outlier factor. An anomaly score is computed by the distance of each instance to its cluster center multiplied by the instances belonging to its cluster.

Using LOF we find 9 alloys which are outliers for all features. These alloys may be of further interest due to their rare feature distribution. We find 9 alloys to be outliers namely AlB12, B4Co, B2CoW2, AlRe2, Al4Li9, Cd5Li4Mg, CdIn3Na2, SrCaYbLi0.55Mg0.45Zn.

We observe that most outlier alloys contain Al, B, Mg and Cd.

Preparing training and testing sets

After observing that many alloys of 4 or more metals show significant differences in their properties as compared most alloys of lesser no. of constituents. Thus, we want alloys of all no. of metals to be equally distributed in our training, validation and test sets. To achieve this, we separate alloys into binary, ternary, quaternary and higher order, shuffle them and distribute them to the three datasets.

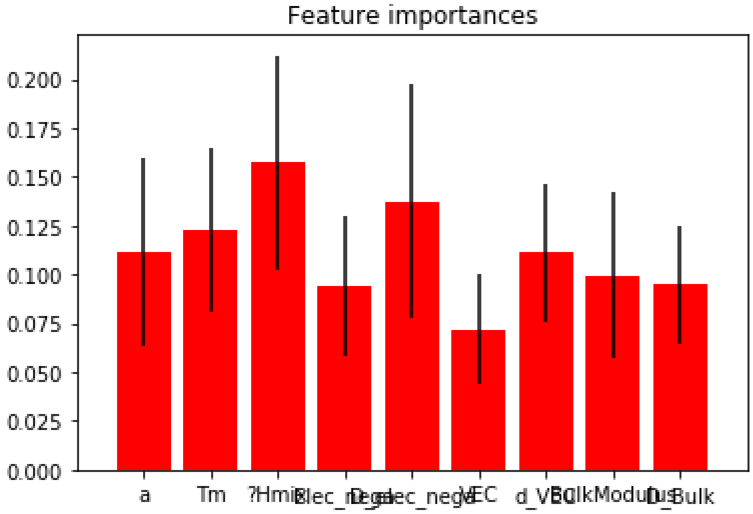
Scaling our Data

We use scikit learn Standard Scalar for scaling our datasets

Classification using various classifiers

Logistic Regression

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable



The classifier gives us the importance of each feature. According to this classifier, Enthalpy of mixing is the most important feature and VEC is the least important feature.

K-nearest Neighbor (KNN)

This algorithm classifies data points based on clustering of nearest neighbors by Euclidean Distance or any other measure for distance between two data points.

Support Vector Classifier

Support Vector Machines classify data points by learning a hyperplane distinguishing the data points.

Gausian Naive Bayes

Naive Bayes Algorithm calculates probability for each class, and computes the conditional probability of each input for its corresponding class. The Gaussian Naive Bayes assumes that the data points are distributed normally.

Decision Tree Classifiers

These classifiers create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

SGD classifier¶

These classifiers optimize using the Stochastic Gradient Descent rule.

The accuracies for all the models above are summarized below.

Model	**Accuracy**
Logistic Regression	72.2%
K-nearest Neighbor (KNN)	76.6%
Gaussian Naive Bayes (KNN)	60%
Support Vector Classifier	75.5%
Decision Tree Classifiers	78.8%
SGD Classifier	60.0%

Now we move on to ensemble methods for better accuracy

Ensemble methods

Bagging Classifier

A Bagging classifier is an ensemble meta-estimator that fits base classifiers each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction. Such a meta-estimator can typically be used as a way to reduce the variance of a black-box estimator (e.g., a decision tree), by introducing randomization into its construction procedure and then making an ensemble out of it.

Random Forests Classifier

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

```
In [305]:
```

AdaBoost Classifier

An AdaBoost classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.

Gradient Boosting Classifier

GB builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage n_classes_ regression trees fit on the negative gradient of the binomial or multinomial deviance loss function.

Voting Classifier

The idea behind the VotingClassifier is to combine conceptually different machine learning classifiers and use a majority vote or the average predicted probabilities (soft vote) to predict the class labels. Such a classifier can be useful for a set of equally well performing model in order to balance out their individual weaknesses.

Stacked Model Classification

It uses a meta-learning algorithm to learn how to best combine the predictions from two or more base machine learning algorithms.

The accuracies for all the models above are summarized below.

Model	**Accuracy**
Bagging Classifier	74.4%
Random Forests Classifier	83.3%
AdaBoost Classifer	42.2%
Gradient Boosing Classifier	80.0%
Voting Classifier	82.2%
Stacked Model Classification	80.0%

We observe that these ensemble methods give better accuracies ovarall, AdaBoost Classifier being an exception.

Deep Learning Method

We use a 3-layer deep dense Neural Network with RelU activation on the hidden layer and Sigmoid on the output layer. All the hidden layers had a dropout rate of 0.5. Categorical cross entropy loss was used in the Adam optimizer.

We observe that the Neural Network performs badly on this data. This may be because the dataset is too small to generalize.

Test Accuracy

We use our best performing model(Random Forests Classifier with Extra trees) to predict on the test set to get final test accuracy.

the final test accuracy was 86.8%

We also try training this model after removing all outliers found by the LOF

We observe a 3% decrease in classification accuracy. The causes for this are yet to be investigated.

We observe a 3% decrease in classification accuracy. The causes for this are yet to be investigated.

References¶

- Machine-learning phase prediction of high-entropy alloys
- Machine learning guided appraisal and exploration of phase design for high entropy alloys
- Scikit-learn Documentation