**David Montes de Oca Zapiain**

ME PhD 2019 Georgia Institute of Technology

**Citrine Data Challenge Submission**

1. **Introduction**

The objective of this work is to leverage data-science and data-analytics to predict the stable binary compounds an element pair will produce when mixed; a renowned thermodynamics problem.

In order to limit the effect of other phenomena (i.e. heat transfer, time, etc.) and to focus on the effect the different features each element pair have on generating a stable binary compound the following assumptions were used:

* The system is a closed system with negligible heat transfer (adiabatic)
* The kinetic and potential energy of the system are neglected since it is assumed that the results provided are in steady state.
* The results of each component of the stability vector are assumed to be independent of each other.

The data provided has 2572 pairs of elements with 98 features per pair of elements. The objective is to leverage this data to produce a robust model capable of predicting the stable binary compounds that each element pair will produce. The stable compounds are a discretization of the 1D binary phase diagram at 10% intervals, where a 1 indicates a stable compound at that concentration of elements and a 0 indicates that no stable compound has been produced. It is also important to notice that the first and last column of the stability vector are always stable (all have a 1) since these columns represent a 100% compound of each of the elements in the pair, which are inherently thermodynamically stable. Therefore, there is no need to build a model for these components of the stability vector.

1. **Methods**

At first, one can think of using a multi-class classifier to predict all the components of this vector. However, this approach will no doubt be futile because the number of classes that this problem has is considerable for the amount of training data available. Therefore, the amount of training data is simply not sufficient. To solidify this point, think about all the possible combinations of places where a 9x1 vector can have of 0’s or 1’s and also recall that more than one component can be stable for a pair of elements.

A more intelligent way to address this problem is to leverage insights that the data provides in order to simplify the complexity of the problem. Having said that, a first logical step is to investigate the columns of the stability vector that characterize the binary compounds produced by each element pair and determine the number of stable compounds each pair produces. An efficient way to do this is by plotting a histogram of the number of stable compounds each element pair produces as shown in Figure 1.



**Figure 1. Number of Stable Compounds per element pair.**

From Figure 1 it can be seen that the number of element pairs that produce no stable compounds at all is roughly half of all the element pairs. A model that determines whether the input element pair will procure at least one stable compound is a good first step to address this this complex problem since it reduces a multi-class problem to a binary classification problem. In other words, this first model will leverage the features of each of the element pairs to discern which ones of them yield a stable compound.

In order to build a robust and accurate binary classification model, I will train it on the features that have highest influence/impact on the output. These relevant features will be identified by pre-processing the data and leveraging Pearson Correlation. Pearson correlation will be leveraged since it provides quantitative information about the linear dependency of each individual input to the desired output of the model.

The input data was normalized to ensure that their standard deviation is 1 and their mean is 0. Subsequently, the Pearson Correlation coefficients were calculated and used as a measure to identify the most relevant features. The input data was split into a training set (85% of the data) and a validation/hold-out set (15% of the data) in order to prevent building a model that fits the training data in an excellent fashion but fails to generalize to a test set. Moreover, to further prevent the model from overfitting to the training data the model leveraged a Cross Validation.

The performance of this binary classifier will be assessed with the help of a confusion matrix, the area under the ROC curve, and the precision and recall of the associated model. Remember that the objective of our model is to predict which elements produce a stable compound. Therefore, it is highly important for our model to capture as accurately as possible all the element pairs that will produce a stable compound. This performance trait is observed on models that have high recall (lowest number of false positives) and as such high importance will be placed on this metric.

1. **Results**

Having defined a method for identifying the most relevant features to train the model, a formalism to avoid overfitting and the performance metric by which the classifier will be evaluated it was proceeded to try three different simple classifiers: Decision Tree, K-nearest neighbors and support vector machines. The calibrated classifiers were then tested on the hold out set and obtained in average around .85 AUC and the model with the highest recall (SVM) yielded .92. These results might deem satisfactory but recall that we are only using simple classifiers. Therefore, in order to procure a model with better performance ensemble learning was leveraged. Ensemble learning was performed using a Random Forest and the model yielded an AUC of .9 and a recall of .94 on the hold-out test. As a final attempt to obtain better performing bagging and boosting was performed but the model with the best performance was still the Random Forest. It is important to mention that an optimal version of all the different classifiers evaluated was obtained by tuning their hyper parameters to an ideal value. The ideal values for the hyper parameter were found by performing a grid search over the parameters deemed to have most importance for each classifier and for a wide array of values. The detailed results of all the classifiers mentioned previously are on the Evaluating\_best\_model\_for\_stable\_element.ipynb jupyter notebook.

The results obtained by the best performing model on the hold-out test mean that a successful (accurate and precise) model has been defined that allows one to define, with quite a certain degree of certainty, which element pairs will yield a stable compound. Once these elements have been successfully identified it is needed to determine which components of the stability vector will yield a stable binary compound.

In order to achieve this task, we can leverage the assumption that the stability of each component of the stability vector is independent of each other. This assumption enables one to decompose this task into nine different binary classification problems. The previously results show that the employed framework is effective for calibrating a robust and accurate binary classifier. Therefore, we can leverage it to procure the binary classifiers for each component of the stability vector. It is important to mention that the data used as input of the model was further pre-processed for these binary classifiers. The additional step of pre-processing ensured that the model was only trained on the elements that yielded a stable compound for that component. In other words, if the element pair Be-Ge yielded a stable compound for component one, all the element pairs that have Be and Ge will be considered for training for that component. This step was added to increase the performance of the binary classifier. The results of the best binary classifier models obtained for each component of the stability vector are the following:

**Table 1. Results of the classifiers used to predict the components of Stability Vector**

|  |  |  |
| --- | --- | --- |
| **Component** | **AUC** | **Recall** |
| 1 | .82 | .85 |
| 2 | .76 | .69 |
| 3 | .8 | .87 |
| 4 | .78 | .77 |
| 5 | .8 | .9 |
| 6 | .79 | .67 |
| 7 | .79 | .83 |
| 8 | .8 | .78 |
| 9 | .87 | .88 |

The detailed results of the best performing models for each component are on the attached jupyter notebooks. The Validation\_Pipeline.ipynb jupyter notebook puts together all these classifiers and provides the results to the provided test set (test\_data.csv) and writes it to a new .csv (test\_data\_predictions.csv).

1. **Discussion**

The fact that each binary classifier was trained using cross-validation reduces significantly the risk for overfitting. It is also important to remember that the reported results were obtained from a hold-out set on which the model was never trained. This measure enables one to provide a suitable estimate of how the model will perform on a completely new dataset. The first model, the one that determines whether an element pair will yield at least one stable compound, has really good performance metrics on the hold-out test and as such we can expect for it to perform suitably on the test set. An important observation of the performance of this model on the test data set (test\_data.csv) worth mentioning is that this model successfully identified that the elements that did not yield any stable compound regardless of the element they were paired were the noble gases (Kr, Ar, He, Xe, Ne). The fact that the developed model is able to identify that these elements are not capable of producing any stable binary compounds is indeed remarkable, and attests to the power of the framework leveraged to train this model. Most importantly, it demonstrates that the developed model is capable of providing good predictions overall and that most importantly match with the physics/theory of the problem. Having said that, the results in Table 1 show that there are still opportunities for further enhancements, specifically in component 2 and in component 6. Where unfortunately the performance on the validation dataset will not be as suitable as in the other components.

The performance of all the models can be certainly improved by incorporating more relevant features to the model. A great guidance on the features to be incorporated can be procured from the physics of the problem. For example, the governing principle in the mixing of two elements is the Gibb’s free energy and the amount of it generated when two elements are combined. Therefore, an approach I will take will be to supplement the current training data with simulations results of the Gibbs free energy formed for each component of the stability vector. This enriched data-set will enable the developed framework to incorporate more physics into the model and it will most likely procure better results.