**Data Science Challenge**

The idea behind this challenge is to use data to solve a canonical thermodynamics problem: given a pair of elements, predict the stable binary compounds that form on mixing. Within the attached .zip, we've provided roughly 5000 element pairs as training data. Each of the pure elemental compounds have been expanded into features for you using a naive application of the magpie feature set (<https://bitbucket.org/wolverton/magpie>). Feel free to prune, extend, or otherwise manipulate this feature set in pursuit of a predictive model!

The training labels we've provided are a discretization of the 1D binary phase diagram at 10% intervals. For example, the label for OsTi ([1.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,1.0]) translates into the following stable compounds:  Os, Os{0.5}Ti{0.5} or OsTi, and Ti.

Your task is to build a machine learning model in Python to predict the full stability vector.  Please explain your approach, including your choice of algorithm, data-preprocessing, and model accuracy evaluation.  Please write a framework for hyper-parameter tuning from scratch (not using sci-kit learn’s built in methods). Use this as an opportunity to showcase your coding ability!

Please also estimate the performance of your model on the test set (both precision and recall) and explain your methodology in the write up.  Include a discussion of when your model may be more or less accurate.

For evaluation, please send us predictions for the test set we've held out (simply fill in the last column with your predictions in the same format as the training data), along with your code and a brief writeup explaining the results (what worked/didn't work and why?).  Please package your submission in a zip file before uploading (it is easier for us to retrieve all of your files this way).

Again, please feel free to reach out anytime with questions/clarifications. I hope you'll give the challenge a crack - we're really looking forward to reviewing a solution from you!

We will evaluate your submission based on:

* Code quality, including readability and extensibility
* Explanation of approach in write-up
* Demonstrated statistical intuition
* Model performance (as evaluated via the predictions on the test set)
* Accuracy and methodology of the model performance estimate for precision and recall