# Kobold programming challenge prompt

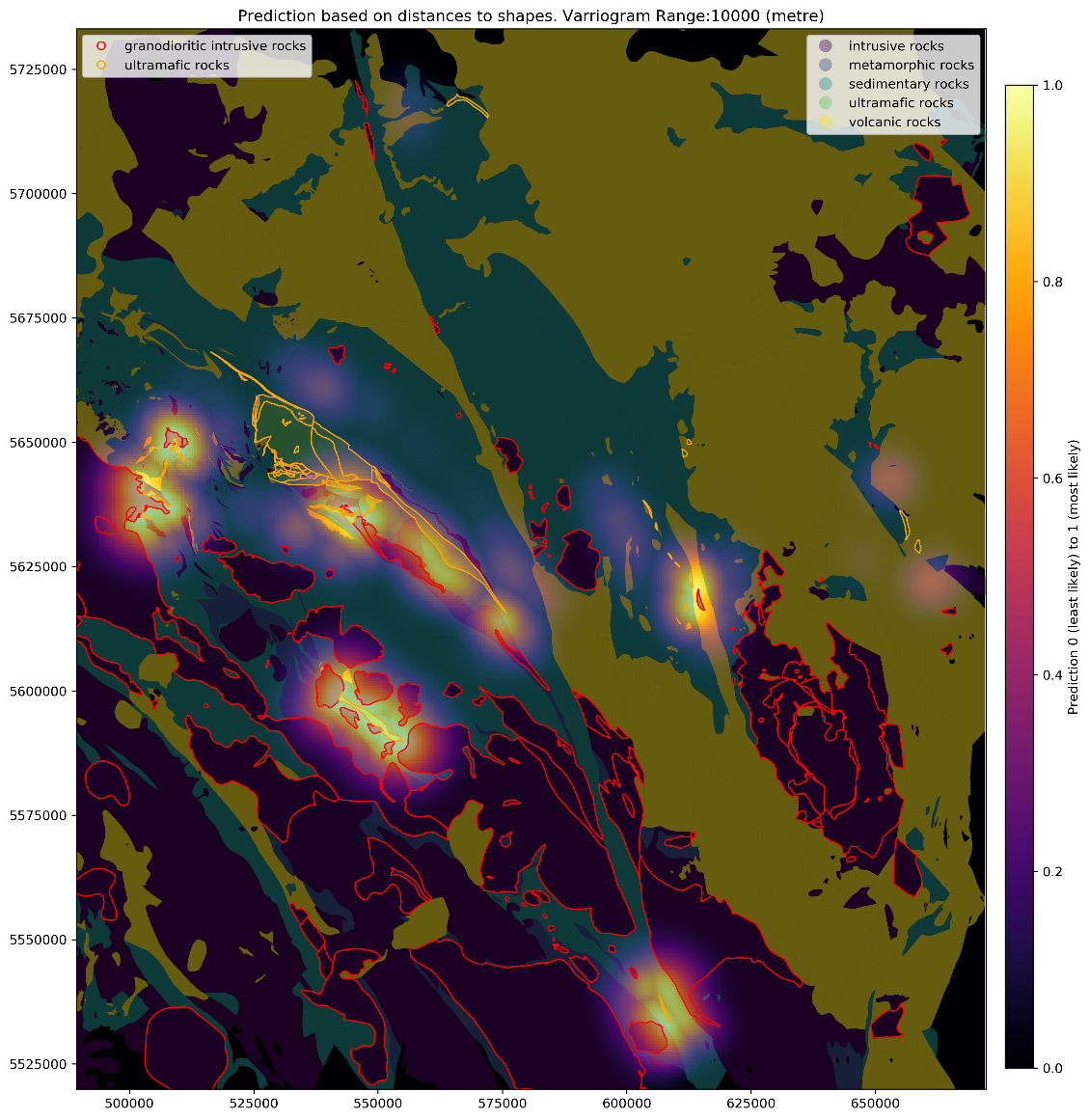
The objective was to create a dataset in the form of a heat map that represented the likelihood of finding a cobalt deposit at each point on the map. The heat map is stored in a raster format (included as range10km\_pred.tiff). Included is the methodology I used in writing the python package used to make the likelihood predictions, some thoughts and notes on that methodology and all information required to read and or reproduce the data.

## Problem

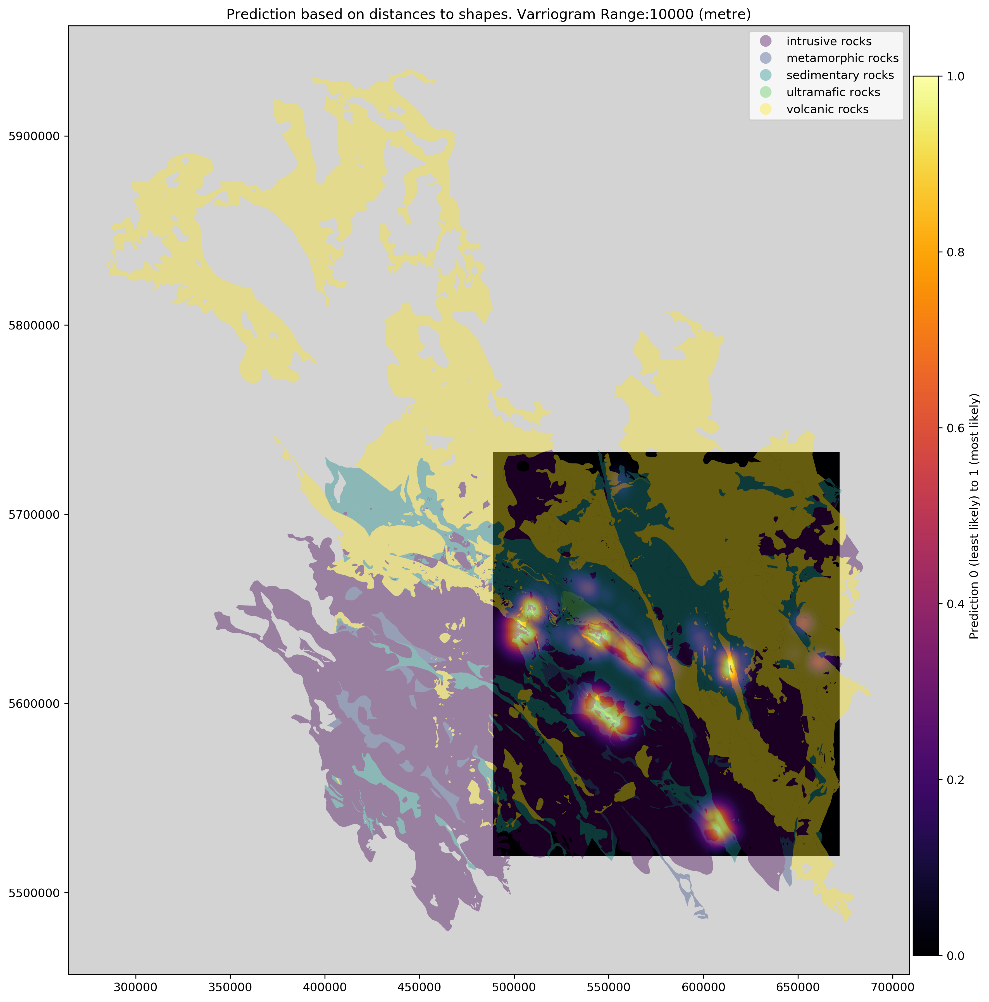
Generate a heat map representing the likelihood of finding a cobalt deposit. The prediction is based on a weighting schema given the distance to specific bedrock units (granodiorite and serpentinite/ultramafic rocks) and a prediction range. The prediction should fall smoothly towards zero at some distance away (i.e. the range).

## Solution

To solve the problem, I developed a python package and prediction weighting schema to predict the likelihood of finding a cobalt package. The resulting heatmap is saved as a raster file (range10km\_pred.tiff) that can be loaded and viewed using the included package or the rasterio package. Below is the resulting heat map zoomed into the project boundaries.



Here is the heat map superimposed on the larger shapefile boundary to get a better sense of the location of the prospective areas.



## Python Package

The package developed for this challenge (prospectPredictor) handles the prep work and prediction based on GIS shapefiles. The package resides at <https://github.com/tyleracorn/prospectPredictor> and is included in a zip file. Included in the package is an example.ipynb that demos using the package and creating the heat map (I’ll also include a pdf of the notebook in the email with this document). Since the package relies heavily on other package such as geopandas, an environment.yml file is included to show the environment I used when developing the package.

### Package structure and methodology

The package is structured into 3 classes that are designed to be chained together. Each class is meant to handle the following steps of the workflow

* Explore and prep the input shapefile
* Create a raster template
* Predict likelihood for each location in the raster

#### PrepShapes()

This class handles exploring and prepping the shapefile and determining a project boundary. It can read in a single shapefile (using geopandas) and you can use it to select the polygons of interest. In the provided example I selected as my project shapes “granodioritic intrusive rocks”, and “ultramafic rocks”

#### RasterTemplate()

This class uses the previously initialized PrepShapes() class to help you setup the raster template used by the predictor to prediction prospectivity at a specified location

#### PredictByDistance()

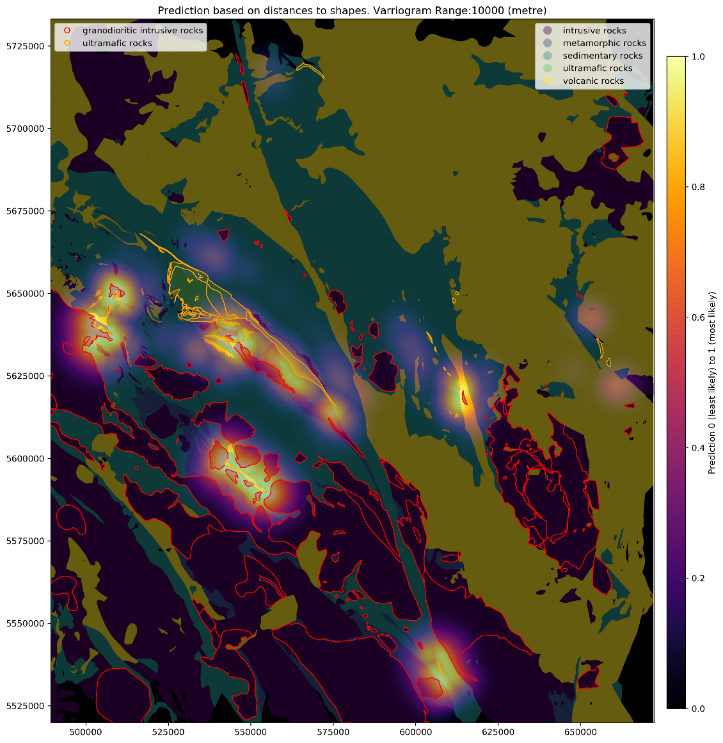
The final class is the predictor class that predicts the likelihood based on distance to shapes of interest and the weighting schema architect. Only one prediction architect was included for this project.

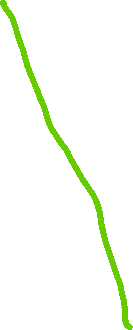
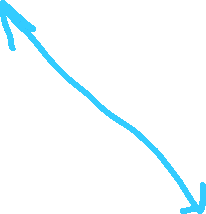
## Prediction Weighting Schema

For the prediction I decided to use an omni-directional variogram weighting schema using the variogram model below.

In the above variogram model dist1 = distance to 1st bedrock unit, and dist2 = distance to 2nd bedrock unit. This ensured that the predictions approached zero when a location reached the range of the variogram. There are some issues / notes to think about with this weighting schema.

### Weighting schema issues:

1. The weighting schema I used is an omnidirectional schema so there isn’t any directional continuity captured in the prediction. This is will likely increase the uncertainty of the predictions since we are dealing with geological features that do surely have directional continuity associated with them. This is apparent in the heat map plot below. Green line shows a possible fault. Blue line shows orient of likely major direction of continuity.



A more realistic weighting schema would take into account the major and minor directions of continuity in the spatial relationship of the bedrock units.

1. Uncertainty is not captured. The weighting schema makes is making a estimation without taking into account the distribution of possible predictions at each location. We would need to incorporate a more stochastic prediction approach to also have access to the uncertainty of the predictions at each location

The prediction raster/heat map included in this report uses the following template:

CRS: EPSG:26910  
Cell Width, Height: (500, 500)  
Raster Dimensions: (427, 366)  
Raster DataType: float32   
Raster Boundary: minx (489327), miny (5519925), maxx (672302), maxy (5733138)  
prediction values: range from 0 (least likely) to 1 (most likely)