DIY – Implementing Customized Python Workflows for Flexible and Effective Mineral Resource Estimation

L. Bertossi[1], *D. Carvalho*[2]

1. Snr. Resource Geologist, Glencore, Brisbane, 4123. Email: laerciobertossi@glencore.com.au
2. Principal Geologist, GeologicAI, Brisbane, 4123. Email: dcarvalho@geologicai.com

Keywords: Python, Programming, Automation.

ABSTRACT

Traditional and conventional commercial mineral resource estimation (MRE) software rarely offer the flexibility needed to create modern customized workflows and address specific requirements of unique deposits and operations. These software packages are typically designed with a set of predefined tools and functionalities (sometimes with scripting capabilities) but often lack the required adaptability when dealing with more complex deposits, unconventional settings, different data sources and types. When scripting is available, most of these packages offer specific programming languages, making it challenging for users to learn and integrate different platforms and tools.

This paper advocates for a 'Do It Yourself' (DIY) approach to generate MRE workflows using Python, a general-purpose programming language that has become one of the most popular worldwide. Python workflows are transparent, easy to learn, self-documenting, capable of processing large datasets and integrating many machine learning routines. Such workflows improve reproducibility, validation and revision of results, independent of the commercial software used for modelling.

Python's open-source nature allows for greater flexibility, integration and control over the entire MRE workflow. By integrating various free-to-use libraries and capable commercial geostatistical/resource modelling modules, it is possible to create dynamic and interactive routines that incorporate all key steps: data validation, exploratory data analysis, variography, block modelling, and grade estimation. In a scripted environment, many iterations are possible, and validation becomes more visual, featuring a variety of customized plots, statistics, grade-thickness relationships, 3D views, grade-tonnage curves and swath plots in a streamlined fashion.

This flexibility is highly effective for large-scale MRE projects, as it automates repetitive tasks and reduces time spent on manual processes. This allows practitioners to focus on critical geological interpretation, refining models, and testing estimation scenarios, leading to more accurate and realistic MRE.

This work explores insights into the advantages of Python over other programming languages, practical examples, and a selection of open-source and commercial Python-based tools for MRE and machine learning, aiming to inspire geologists and modellers to innovate and approach challenges creatively.