The many programming languages of predictive analysis

Introducing Python

In this activity, you will read a short overview of available programming languages and an introduction to Python (the programming language of this course).

Coding is key

Throughout the course, you will learn to implement various concepts related to predictive analysis. To get a good grasp on the whole modelling process, you have to get your hands dirty as this is the most effective way to take in what you have learned. Working with data is also not a straightforward task. Despite the fact that the data in this course is usually provided, collecting and cleaning it is often the lengthiest part of the whole process. You will only run into all the particularities of a dataset once you actually start using it, which is impossible without computers, and computers rather like their instructions in code. The ability to code is truly a key skill, and often the most sought-after attribute of a business analyst.

About coding languages

Coding is a prerequisite of the course after all, however, we will quickly refresh the basics of programming in Python in the next activity. There exist a variety of languages of which Java, C, C++, Python, R, Scala, MATLAB, SQL and SAS are probably the most prevalent ones. They all have their benefits and downsides and are used for different purposes. SQL, for example, is mostly used for querying and data manipulations. C provides a language written closely to the computer's hardware that forms the underpinning for a lot of other languages such as Java. The latter is especially strong in object-orientation and is heavily used in software engineering (most notably the Android operating system) and Python, R, MATLAB, SAS, and Scala are ubiquitous in data science.

Code (2016) by StockSnap on Pixabay, licensed under CC0 (modified image)

Many libraries that we will use (more on this later), are also available for a number of languages, as they have interfaces in Python, R and so on, but have the main procedures translated to C to ensure performance. C, Java and Scala (although this is somewhat more complicated) are compiled before execution, while Python and R are interpreted on the spot. This means we call the program to perform certain operations, rather than creating a program (which is done by compilation). Python and R are probably the most important for predictive analytics and overall can be used in the same fashion. They are both supported by Jupyter (JUlia PYThon R, but Scala is also supported) notebooks, which will be used throughout the course. They are an environment to create a particular set of code snippets that are great for data visualisation and analysis.

The case for Python

The reason for using Python in this course lies mainly in its recent popularity and the slightly lower initial learning curve. Python was created to be easy to understand and use, with a lot of syntax that is concise yet powerful. Python 2 and 3 are the two versions of Python, but unfortunately are not completely compatible. We will be using Python 3, although you should be able to understand most of Python 2 as well. Python supports object-oriented, procedural, functional and other programming paradigms, which makes it highly versatile. Also, it uses dynamic types, which means objects we create can change from a string in one location, to an integer in another. This makes debugging quite a pain sometimes. A good piece of advice: always try to be careful with reusing variables.

All in all, it is important to become fluent in writing Python, as it will help you a great deal throughout the MicroMasters. You will use it for many module exercises, projects and other assessments.

Learn About Verified Certificates