

Reference guide: Python for machine learning

Previously, you learned more about the Python ecosystem for machine learning. You developed an understanding of the different Python file types that are available to approach a data analytical task, along with the various types of integrated development environments in which the coding takes place.

Here, you will learn more about different Python use cases, along with the advantages and disadvantages of using different types of files and development environments.

Coding languages for data professionals

As the field of data science has progressed, steadily tools have been released to facilitate the development of data-driven solutions for various problems. Not only have the tools become more sophisticated, but often easier to use as well.

Pieces of software such as Tableau and Looker have made performing data analysis much more simple and efficient. They can allow for quick yet comprehensive overviews of a dataset and are often used as a starting point before deeper analyses or developing models from the data.

This is where coding languages have a huge role in continuing to solve the task at hand. Not only can they perform much of the same preliminary analysis that is done in other pieces of software, but they also contain some very powerful functionality that can be used quite easily.

The two most popular coding languages for data science are R and Python. For the most part, both achieve all the same things, and deciding which one to use usually depends on personal preference or what the rest of your team is using.

The R language was designed for statistics from its inception. Much of the functionality required is baked into the language itself. Python, on the other hand, was and still is a general purpose language that gained popularity within the data science ecosystem.

In this certification, you have exclusively used the Python language to learn about data analytic techniques and how to use those techniques to solve problems you might encounter in the workplace. While you might come across a situation where you need to use R to perform an analysis, the same principles and concepts you have learned here apply no matter the software used.

Types of Python files

As you learned in a video, there are two general types of Python files—Python scripts and Python notebooks. Both types of files can run the same exact code; however, there are certain situations where one is preferable to the other.

Python scripts

Python scripts are arguably the more common type of file overall, but not necessarily in the world of data science. They are denoted with the file extension “.py,” and are used much more for larger projects, or for projects where it is not essential to see each part of code run separately. For code that is going to be deployed and put into production, Python scripts are much more common. They are much easier to debug than Notebooks, along with being much better for reproducing results. On top of that, they work much better with other pieces of software and infrastructure.

Python notebooks

Python notebooks are what you’ve been using throughout this entire program. While they technically can do everything that a script can do, they are mainly used for exploration, visualization, or presentation.

One of the main features of Python notebooks is being able to easily run different sections of code independently. Additionally, you are able to see the output of each section of code, rather than having it all come out at the end as is the case with scripts.

Notebooks also contain functionality to insert non-code elements into the notebook itself. If you want to insert text inside a Python script, you are limited to writing comments. But with

notebooks, you are able to add markdown text, images, and links to provide more context to the code.

Integrated development environments for Python

When doing any coding related task, an integrated development environment, or IDE, is where much of the work is actually done. It is a piece of software that gives a place to write, test, and run code. For any programming language, there are often many different IDEs that are available to use, varying slightly in functionality and included tools. Selecting one to use often comes down to what types of tools you need to create your program or even just personal preference.

In this certificate program so far, you've been using Jupyter Notebooks on the internet. Jupyter Notebooks is an IDE, however it only supports Python notebook files. It is possible to create your own instance of Jupyter Notebooks on your personal device, but the functionality is essentially the same as using a web-based instance. Other IDEs, such as Visual Studio Code and PyCharm, run locally on your device.

IDEs can have many included tools, but there are a few that are so common that you'll find them paired with almost every IDE. Code completion, file management, and debugging support come in very useful, streamlining your workflow and letting you solve problems in your code as they come up.

Key takeaways

- Coding languages are useful for approaching a data-driven problem. Python and R are two popular options for coding languages, each with their own advantages.
- The two main types of Python files are known as Python scripts and Python notebooks
 - Python scripts are better for production grade code, and are easier to debug and manage.
 - Python notebooks are better for exploratory analyses, presentations, or anything that needs to be human-facing. You are able to insert images, text, and links directly into the code.
- An integrated development environment (IDE) is a piece of software that gives a place to write, test, and run code. Every language has its own set of IDEs to choose from. IDEs

themselves can offer different benefits, such as code completion, file management, and debugging tools.