Big Data and Analysis with Python

With the advent of social media and its widespread acceptance came the unprecedented need for data management. Now billions of gigabytes of data are produced every day and much of it is generated by the end-users. Organizations recognized the huge potential in harnessing this data using predictive and machine learning algorithms to generate insights. But before tackling that challenge, came the challenge of efficiently and systematically storing and handling this data in a way that made it available for quick access.

Big data is the management of large sets of data, both structured and unstructured. Today, this large amount of data is stored in the form of data warehouses and data lakes, both on servers and in the cloud. The main characteristics that are commonly identified for use of Big Data tools are Volume, Variability and Velocity.

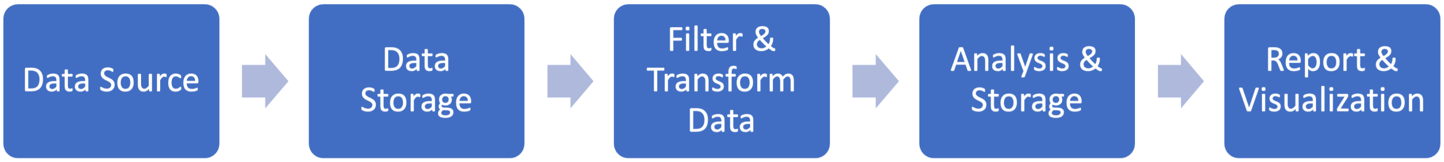
Volume is the size of data under question and, if large enough, may require different handling to traditional data storage and management.

Variability or veracity refers to the inconsistency that may be present in this data. In huge data repositories, it is difficult to intervene manually on every wrong entry and thus enough scope for variability must be defined and established while handling such data.

Velocity is the speed of handling this data. With data sources such as social media which are continually active, there is a need for constant updates as well as robust storage. When there is a need to be processed, it should also not create a bottleneck where data retrieval takes longer. As such, velocity plays a very important role in Big Data.

This is the ability to handle a large amount of heterogeneous data with ease of access and speedy processing. The next step in this process is when this data is analyzed and is broadly called data analysis. The final step is publishing this data in form of reports, visualizations and web pages, as per the requirement.

The whole pipeline can be summarized as below:



Data Source → Data Storage → Filter & Transform Data → Analysis & Storage → Report & Visualization

Here are several reasons why Python has found a place in the domain of Big Data:

Ease of use: Ease of use is a prerequisite for any large-scale and commonly used technology and language. Python helps setting up and running infrastructure with just a few lines of code.

Licensing structure and open-source nature: The open source paradigm has picked up immensely in recent years. Python provides many very well developed open-source libraries and frameworks, even for large scale applications. Some organizations prefer this, as it saves on cost, as well as providing easier licensing.

Active community: The Python community today is vast and very supportive. This helps with the swift resolution of issues a user may face, as well as the development of new features when required.

Libraries: Possibly the strongest reason for the acceptance of Python is the host of libraries that provide direct support for Big Data. In addition, there are many packages that also aid in bridging the gap between Python and other languages and tools enabling swift deployment of services.

High compatibility with Hadoop and Spark: Hadoop and its Hadoop distributed file system is arguably one of the best solutions for large-scale storage. The support available in Python has also helped in wider acceptance of Python. The same can be said about Spark as Python has supportive libraries such as PySpark and host of API libraries that facilitate its usage.

High processing speed: Python has support for prototyping and with its Object-oriented methodology, processing in Python is much better in comparison to other languages. With its increase in speed, Python is also able to provide adequate stability in its usage.
Portability and scalability: Broadly as mentioned before, Python's support for cross-language platforms and operations, its ease of extensibility, various libraries, support for frameworks and API overall, makes it easy to scale and flexible.
Python tools and libraries: Most of the libraries in Python that are used for Big Data are widely common and is associate with Data Sciences and Machine Learning. Big Data includes wide-scale usage and acceptance of libraries such as: Numpy, Pandas, Scikit-learn and Scipy. To name just a few.
Additionally, here are a few more libraries that are more specific to a Big Data domain such as:

RedShift and S3: Amazon services are used with their cloud services. S3 is a storage service and RedShift is a data warehousing service.
BigQuery: Developed by Google, BigQuery is a Cloud service library that is useful with RESTful APIs.

PySpark: This is an open-source framework used for large scale data processing and works with resilient distributed datasets.
Kafka: This is a publish-subscribe messaging system that receives logs in the form of packages and is stored in partitioned spaces.
Pydoop: Pydoop provides an interface between Hadoop and Python and support for handling its Hadoop distributed file systems.
Mark as completed

👍 Like 👎 Dislike ⚐ Report an issue