

Big Data Analytics with Pandas and Scipy Python Tools

Satish Premshankar Yadav¹, Adarsh S.K Singh²

¹Satish Premshankar Yadav, Department of Master of Computer Application, ASM IMCOST Thane(W),India

²Adarsh S.K Singh, Department of Master of Computer Application, ASM IMCOST Thane(W),India

Abstract - Big Data poses an inherent challenge in terms of its interpretation and analytics. Numerous software's are emerging to refine this data challenge and are bundled with plethora of tools for data correlation and analysis. This paper discusses Big Data, challenges and Use Cases of Big Data to highlight the requirement of Big Data Analytics. It also gives an insight into hottest emerging Python tools for giant Data Analytics alongside few of the examples. These libraries provide integrated, intuitive routines for performing common data manipulations and analysis on such data sets. It aims to be the foundation layer for the future of statistical computing in Python.

Key Words: Big Data, Big Data Analytics, Python, Data Science, Pandas, Matplotlib, Scipy

1. INTRODUCTION

Big Data is an extensive term for any collection of data sets so large or complex that it becomes difficult to process those using traditional data management techniques such as RDBMS (Relational Database Management Systems). RDBMS has been considered as a one-size- fits-all solution, but the requirements of handling Big Data are varied. Big Data has three V's characteristics, viz. Velocity, Volume and Variety. The data needs to be organized to transform the countless bits and bytes into actionable information. Sheer quantity of data would not be helpful unless we have ways to make sense out of it. Traditionally, programmers wrote code and statisticians did statistics. The programmers coded using a programming language, while statisticians drew interferences using specialized programs such as SPSS (Statistical Package for the Social Sciences) of IBM. Statisticians banked upon the national statistics or market research, whereas programmers managed large amounts of data in databases or log files. But now, Programmers, Statisticians, Scientists, Analysts and in fact everyone connected on net has access to a large amount of raw data. With this availability of Big Data from the cloud to virtually each individual changed the entire gambit.

2. Big Data

The term "Big Data" refers to any set of data that is so large or so complex that conventional applications are not adequate to process them. The term also refers to the tools and technologies used to handle "Big Data". Examples of Big Data include the amount of data shared in the internet every day, Amazon website accessed, Twitter feed, Facebook posts and mobile phone location data. In the recent years, data produced by learning environments have also started to get big enough raising the need for Big Data technologies, the tools to handle them and Big Data analytics to analyses that huge amount of data and make valuable decisions for individual or company.

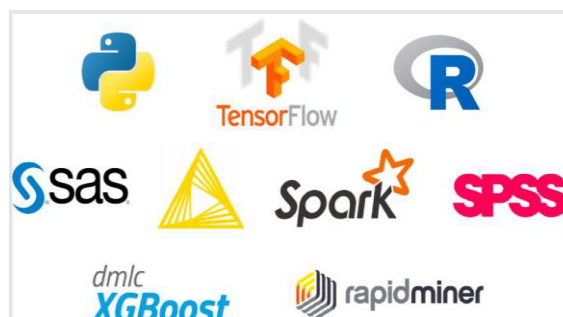


Fig -1: Tools available for data analysis

2.1 Challenges in Handling Big Data While handling Big Data, various challenges that need to be addressed include:-

- a) **Storage.** While the common capacity of hard disks nowadays is in the range of terabytes, the amount of data generated through internet every day is in the order of Exabyte. It is impossible for the traditional RDBMS tools to store and process such Big Data. Therefore, in order to overcome this challenge, databases should be such that they do not use traditional SQL based queries. Compression technology is used to compress the data at rest and in memory.
- b) **Analysis.** As data generated from several types of processes differ in structure and the size of the data is also huge, analysis of the data may consume a lot of time and resources. To overcome this, scaled out architectures are used to process the data in a distributed manner. Data are split into smaller pieces and processed in a vast number of computers available throughout the network and the processed data is aggregated.
- c) **Reporting.** Generally, the reports are based on statistical data and displaying the same in the form of numbers. Such reports are difficult to be interpreted by human beings if the amount of data involved is large. Hence, the reports should be represented in an easily comprehensible form by looking into them. These challenges can be easily overcome using Big Data technologies.

2.2 Types of Big Data Analytics

Big Data Analytics should not be treated as a one- size-fits- all blanket technique. Best Data Analysts have the ability to identify the kind of analytics that can be leveraged for gaining benefits for their particular requirement. There are four dominant types of analytics available today:-

- a) **Descriptive Analytics.** Data Analytics is used to describe raw data and present it in a form that is easily interpretable by human beings. It analyses past events that allows organizations to learn from past behaviours, and help them influence future outcomes. Descriptive statistics are useful in showing things like total stock in inventory or average money spent per customer. Amazon has more than 1.5 billion items in its retail stores spread across 200 centres across the world and it uses Big Data to monitor, track and secure them. Amazon stores the product catalogue data in its S3 cloud that can write, read and delete objects up to 5 TB of data each. The catalogue stored in cloud receives more than 50 million updates a week and every 30 minutes all data received is reported back to the different warehouses and the website.
- b) **Diagnostic Analytics.** Diagnostic analytics focuses on examining the data/content to find answer to the question "Why did it happen?". It is characterized by techniques like drill-down, data discovery, data mining and correlations. It examines the database deeply to understand the causes of events and behaviours. One of the medical diagnostics company analysed database of millions of records to develop first non-intrusive test to predict coronary artery disease. This diagnostic based on the analysis of approximately 100 million samples was conducted to finally identify the 23 primary predictive genes for coronary artery disease.
- c) **Predictive Analytics.** Predictive analytics is useful in "Predicting" the next sequence of events likely to occur. This provides the organizations with actionable insights based on data. Moreover, it also provides estimates of the likelihood of a future outcome though, no statistical algorithm can "predict" the future with 100% guarantee. A very popular application of predictive analytics is to predict the credit score. These scores play an important role for the financial institutions to predict the probability of timely payment by the customers in future.
- d) **Prescriptive Analytics.** Prescriptive analytics is a relatively new field and it facilitates users to "prescribe" options of diverse possible actions to be implemented and also guide them towards a viable solution. It presents the quantified effect of future decisions in order to advise on all the possible outcomes before those decisions are actually made on ground. Not only does it predict what will happen, but it also tells the reason of why it will happen and thereby provide recommendations. For example, Netflix uses Big Data Analytics to prescribe favourite song/movie based on customers interests, behaviour, day and time analysis.

3. Big Data Analytics for Python

3.1 Advantages of Big Data Analytics for python

Python is the most popular language amongst Data Scientists for Data Analytics not only because of its ease in programming but also due to several other in-built features that Python offers. Few of the most important features are enumerated as under:-

- a) **Best Library for Big Data Analytics.** Before Python tools for Big Data, Data Science was something only statisticians, operations researchers, and applied mathematicians could do. Python libraries like Pydoop and Scipy are the best alternatives for data scientists who in the past used more scientific tools like Wolfram Alpha and Mathematics, Matplotlib, R to name a few.
- b) **Portable Library and Tools.** Just a couple of lines of code can help the user browse, summarize, and arrange data, just like Excel. That is of more significance if they use scratch pads like Jupyter, as they support creating graphs and tables with one button.
- c) **Extensibility.** Python code is extensible over multiple platforms. Many Machine Learning algorithms are written in Python. These include Google TensorFlow, Microsoft Cognitive ToolKit, Scikit-learn and Spark ML (Spark Machine Learning).
- d) **Diverse Data Structure Support.** It supports latest data structures like maps and sets in addition to the primitive types like integers and complex numbers. Python tools like Numpy supports matrices, lists, linear algebra and many more libraries required for Data Analytics and Machine Learning Algorithms.

3.2 Big Data Analytics for Python Tools

Python has an overwhelming number of packages that can be used in Big Data Analytics and Machine Learning. The Python machine learning ecosystem can be divided into three main types of packages as shown in fig.

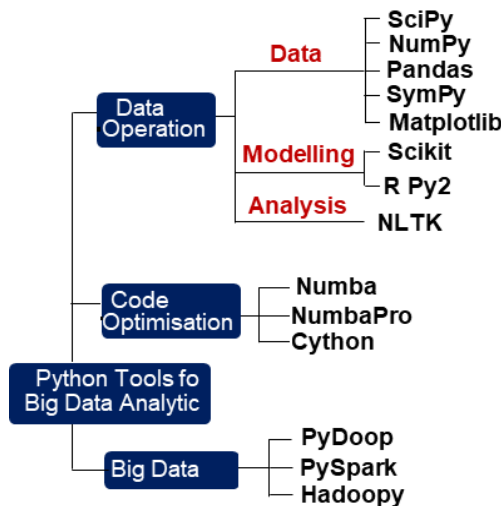


Fig -2: Popular Big Data Analytics for python Tools

4. Pandas

Pandas is a high-performance, very easy-to-use, data-wrangling package. It has a special data type called Data frames, a type of in-memory data table. Pandas provide diverse data structures and functions to aid in working with structured data in a fast, easy, and expressive way. It is one of the important libraries enabling Python to be a powerful and productive Data Analytics Tool. Pandas combine high performance array-computing features of NumPy with the flexible data handling features of spread sheets and RDBMS (such as SQL) . It has a sophisticated indexing functionality to make it easy to reshape, slice and dice,

perform aggregations, and also select subsets of data. It contains well defined data structures and manipulation tools to make data analysis fast and easy.

4.1 Pandas Data Structures

Pandas have two types of data structures, Series and Data Frames. Series is a one dimensional data structure (“a one dimensional array”) to store values and it has a unique index for every value it holds. A Data Frame is similar to a table or a spread sheet and it contains an ordered collection of columns. Each column can have a different value type (numeric, string, Boolean, etc.). The Data Frame provides an auto row and column index for the values.

```
In [28]: from pandas import Series, DataFrame
In [29]: import numpy
In [31]: s1=Series([5,6,3,7,8,-34,76])
In [32]: s1
Out[32]:
0      5
1      6
2      3
3      7
4      8
5     -34
6      76
dtype: int64

In [34]: data={'student_roll':[1,2,3,4,5,6],
              'student_name':['shakti','anurag','vivek','acb','afds','bvc'],
              'studetn_addr':['street1','s2','s3','s4','s5','s6']}
In [35]: frame=DataFrame(data)
In [36]: frame
Out[36]:
   student_roll  student_name  studetn_addr
0              1          shakti    street1
1              2          anurag         s2
2              3          vivek         s3
3              4             acb         s4
4              5             afds         s5
5              6             bvc         s6
```

Fig -3: Series & Data Frames in pandas

4.2 Filters

For effective data analysis, our tools should provide filter facility to sort data with to respect one or more fields. Panda provides this facility very effectively, in this we can take out data equivalent to some field or print data having a field common in all the records. Let’s say, we want to see a list of only the users who belong to country “country 7”. This filtering of data at run-time is very valuable in Big Data Analytics for sorting the values amongst billions of data sets available.

```
In [18]: article_str[article_str.country == 'country_7']
Out[18]:
   my_datetime  event  country  user_id  source  topic
0  2018-01-01 00:01:01  read  country_7  2458151261  SEO  North America
1  2018-01-01 00:03:20  read  country_7  2458151262  SEO  South America
2  2018-01-01 00:04:01  read  country_7  2458151263  AdWords  Africa
3  2018-01-01 00:04:02  read  country_7  2458151264  AdWords  Europe
8  2018-01-01 00:07:21  read  country_7  2458151269  AdWords  North America
11 2018-01-01 00:08:57  read  country_7  2458151272  SEO  Australia
14 2018-01-01 00:11:06  read  country_7  2458151275  Reddit  Africa
15 2018-01-01 00:11:22  read  country_7  2458151276  SEO  North America
21 2018-01-01 00:15:50  read  country_7  2458151282  Reddit  Africa
24 2018-01-01 00:17:58  read  country_7  2458151285  Reddit  Africa
29 2018-01-01 00:23:51  read  country_7  2458151290  Reddit  Asia
31 2018-01-01 00:26:28  read  country_7  2458151292  Reddit  Australia
32 2018-01-01 00:26:48  read  country_7  2458151293  Reddit  South America
33 2018-01-01 00:27:10  read  country_7  2458151294  Reddit  Australia
37 2018-01-01 00:31:24  read  country_7  2458151298  AdWords  Europe
```

Fig -4: Filtered Data

Data Aggregation, Grouping & Formatting: Pandas can very easily perform various data aggregation functions like min, max, count, sum etc. A Data Analyst requires detailed segmentation to display output or analyse data. For this, grouping of data according to specific field(s) is required. In Fig, grouping of data has been done as per animal name and the mean of water Requirement per animal is calculated with just a single line of code. Pandas have various formatting methods like merge, sort and reset index, fill etc. Data available to Data Analytics from various sources is generally in smaller tables. Therefore, they need to pull out data from two or more different tables. The solution for that is provided by Formatting functions like merge.

```
In [32]: jungle.groupby('animal').mean()
Out[32]:
   uniq_id  water_need
animal
elephant  1002.0    550.000000
kangaroo  1021.0    416.666667
lion      1017.5    477.500000
tiger     1006.0    310.000000
zebra     1012.0    184.285714

In [33]: jungle.groupby('animal').mean()[['water_need']]
Out[33]:
   water_need
animal
elephant  550.000000
kangaroo  416.666667
lion      477.500000
tiger     310.000000
zebra     184.285714
```

Fig -5: Grouping & Mean

5. SciPy

SciPy is pronounced as „Sigh Pie“. It is a Python based open-source software for mathematics, science, and engineering. It provides a vast variety of modules for linear algebra, interpolation & integration, special functions for Fast Fourier Transform, signal processing and image transformation. SciPy is basically a library that integrates basic packages used in scientific computing such as NumPy, Matplotlib, Pandas and SymPy. It is a collection of packages used to solve a large variety of standard problem domains in scientific computing. It can be utilized for standard continuous and discrete probability distributions (density functions, samplers, continuous distribution functions), various statistical tests and more descriptive statistics.

a) Discrete Fourier Transform (DFT)

```
In [28]: %matplotlib inline
from matplotlib import pyplot as plt
import numpy as np

#Frequency in terms of Hertz
fre = 5
#Sample rate
fre_samp = 50
t = np.linspace(0, 2, 2 * fre_samp, endpoint = False )
a = np.sin(fre * 2 * np.pi * t)
figure, axis = plt.subplots()
axis.plot(t, a)
axis.set_xlabel ('Time (s)')
axis.set_ylabel ('Signal amplitude')
plt.show()
```

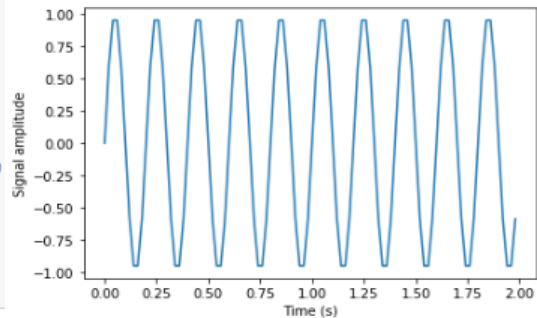


Fig -6: Depiction of Wave using DFT

b) Image Processing

```
In [29]: from scipy import misc
from matplotlib import pyplot as plt
import numpy as np
#get face image of panda from misc package
panda = misc.face()
#plot or show image of face
plt.imshow( panda )
plt.show()

In [30]: flip_down = np.flipud(misc.face())
plt.imshow(flip_down)
plt.show()
```



Fig -7: Easy Image Processing

c) Interpolation

```
In [2]: import numpy as np
from scipy import interpolate
import matplotlib.pyplot as plt
x=np.linspace(0,4,12)
y=np.cos(x**2/3+4)
print (x,y)

[0.          0.36363636  0.72727273  1.09090909  1.45454545  1.81818182
 2.18181818  2.54545455  2.90909091  3.27272727  3.63636364  4.          ] [-0.65364362 -0.61966189 -0.51077021 -0.31047698 -0.0071547
 6  0.37976236
 0.76715099  0.99239518  0.85886263  0.27994201 -0.52586509 -0.99582185]
```

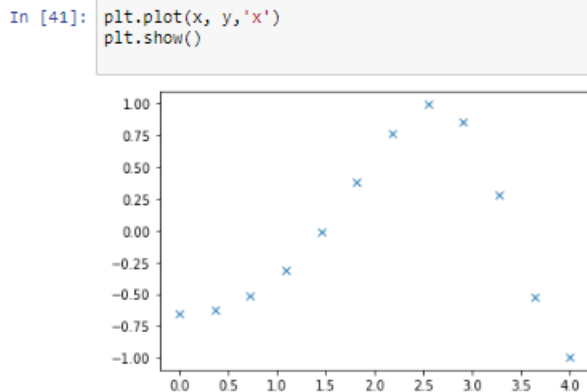


Fig -8: Graphical Interpolation of Arrays

CONCLUSION

Big Data Analytics is a task that can be achieved only by obtaining structured data by well-defined software tools and programming languages that offer high computational and scientific features. Python tools provide a solid foundation upon which a very powerful data analysis ecosystem can be established. The features offered by various Python tools are apt for Data Analytics and at the same time they are memory efficient and fast. Tools like SciPy and Pandas are very well established in terms of their extensible modules and are widely used by the Data Analytics. Immense numbers of tools based on Python are still evolving to assist in Data Analytics. Python tools are a compelling choice for Big Data Analytics applications due to the easy to use data structures that cohere with the rest of the scientific Python stack.

REFERENCES

- 1) W. McKinney, pandas: a python data analysis library,
- 2) W. McKinney, Data Structures for Statistical Computing in Python Proceedings of the 9th Python in Science Conference,
- 3) Davy Cielen, Arno Meysma, Mohd Ali – “Introducing Data Science, Big Data, Machine Learning and more using Python tools”.
- 4) Vivekananth P, Leo John Baptist A – “An Analysis of Big Data Analytics Techniques” Volume 5, Issue-5, October-2015 International Journal of Engineering and Management Research.
- 5) M. Knox, P. Gerard-Marchant, scikits.timeseries: python time series analysis,
- 6) Wes McKinney – “Python for Data Analysis”.
- 7) Wikipedia – “<https://en.wikipedia.org/wiki/SciPy>”.