ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 7, Issue 9, September 2020

Comparison Study of Python in Scientific Computing Using Mathematical Problems

[1] Naveen S.Sapare, [2] Sahana M.Beelagi
[1] Department of Engineering Mathematics, K.LE.Institute of Technology, Hubballi, 580027, Karnataka, India
[2] Department of Automation and Robotics, K.L.E Technological University, Hubballi, 580031, Karnataka,India

Abstract: By definition Scientific Computing is one branch of applied computer science and mathematicswhich is growing rapidly and has assortment of technical tools required to apply on a particular models of mathematics, computational models, and simulations developed to solve issues in many domains like science, engineering, and humanity problems. To form and implement a numerical model we need a support of high level language. Python having numerous rich libraries to integrate establishes platform which can be utilized for developing any mathematical models used in scientific related application. In this paper we are going to discuss such libraries which serve as core libraries in solving mathematical problems in scientific computing. And also needed essential math theoretical skills and programming that are used to implement and solve given numerical problem. As python and MATLAB are two major programming languages used in many scientific application. This paper reveals experimental study conducted on chosen mathematical problem and result on chosen problem of mathematics using python over its libraries and MATLAB on different configured hardware.

I. INTRODUCTION

In Scientific computing mathematics and computer science become the best way to solve problems recently. Different mathematical models like contained polynomials, polynomial interpolation, linear regression, linear algebra, linear operators, inverse problem are commonly used in different domains like image processing, machine learning, deep neural network and different streams of science. Scientific problems included problems from different streams of sciencejust not only from computer science**.** Python is a high level language used by scientists for numerous computations and mainly used to develop high performance applications by utilizing many scientific rich libraries to perform scientific computing. Python support for numerous rich computational libraries like NumPy, Matplotlib and scipy Which provide great documentation and visualization along with symbolic operators and a huge number of numerical methods for solving of many scientific and numeric applications effortlessly. In scientific computing majority of the techniques, and theories are originally developed in Mathematics and this set of mathematical theories and strategies are named as Numerical Analysis and constitutes importance for the efficient and major part of scientific computing. Huge numbers of the numerical techniques that had been created with the end goal of hand computation which incorporates the utilization of mini-computers for the genuine number-crunching which is tedious and cause many errors. Using high level programming language to solve these issues makes huge difference in terms of performance, flexibility, efficiency and time. The popularity of python probably stems from its relative ease of using huge ecosystem consisting of a huge number of libraries which are open-source. It also has additional frequent mathematical and numerical routines in pre-compiled, high-speed tasks in solving also in implementation of a large number of scientific algorithms. In our work presented in this article we focus on how python as a programming Language is used to solve much mathematical concepts.Our work contributes mainly focusing area of mathematics which are used currently in scientific computing. In this paper we show that python can be effectively used to implement and solve many scientific applications. Moreover, we provide here detailed analysis by taking one topic of mathematics that is regression analysis which is used to solve many applications in scientific analysis. We conducted our study deeper to know the strengths of using python over any other higher programming language under different hardware sources. We considered MATLAB to carry the comparative study and give reasons why actually to choose python which might be useful for the users.

A. Current Work

Many Python based frameworks supports today's scientific applications. In the area of Scientific Computing flexibility and efficiency plays a Vitol role. The speed of different languages in many scientific computing applications and in different domains like machine learning and deep learning are always compared.[1] In

ISSN (Online) 2394-6849

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 7, Issue 9, September 2020

this paper author discusses about futures of python mainly considering scientific and technical applications and compares performance of python on deep learning frameworks on training a CNN while article[2]provides some study to compare the benchmarks of common linear algebra matrix computations using MATLAB and Scientific Python (SciPy).Here author explains the examination with the incorporated low-level languages, the profitability and effectiveness of the Python development condition has been contrasted and MATLAB and Octave utilizing a few benchmark models and [3] describes the development cycle using a simple running example that involves finding average of two parametric function. Reported on the feasibility of the python based environment for the development of scientific computing applications and includes example to compare the performance of the available algorithmic and executional futures and adaptability of scripting in Python applied to scientific applications have been altogether introduced in the text book [4]the adaptability of scripting in Python applied to scientific applications have been altogether introduced in the text book. Python as higher level language is best suited for scientific and engineering programming and it is not only useful but also easily modifiable.[5] author proposes how python tool is solving mathematical problems and vast usage of SymPy module is used to solve Derivatives, Integrals, Series Expansion, Limits, Trigonometric Simplification, Equation Systems, Vector and Matrices, Differential Equations.[6]and[7]secured the overall use of Python as an option to MATLAB for scientific computing. NumPy, SciPy, and Matplotlib are on the other hand utilized as MATLAB, so these three bundles give the option to MATLAB, and its condition is best suit for logical works. Python gives an option to replace MATLAB when its libraries best utilized. Here author highlights on code of python which is more readable, understandable then MATLAB code and utilization of Python as an alternative option to MATLAB while solving applications in scientific computing.

The paper is organized as follows.Section II overview of some python libraries and packages. Section III brings concepts of mathematics which are used majorly in scientific computing. Section IV explores our conducted experimental study mainly on knowing computational speed difference on different hardware machines between the python and MATLAB. Obtained CPU RESULTS and graphs are presented in Section V... Finally, Section VI presents the conclusion of the present research work.

II AN OVERVIEW OF PYTHON LIBRARIES & PACKAGES

Core Python Libraries: Python is one such general purpose language which is used widely from developing to deploying and been replaced by many programming languages because of many reasons. One of the important reasons behind replacing is, it contains amazingly large collection of libraries. Known to be beginner's level programming language because of its simplicity, python attracts many developers to create new libraries. The Python Standard Library contains syntax, token, and semantics of Python. We discuss such few libraries in this article which are mainly used in solving many numerical issues in various parts of scientific computing which form the core eco system of python for computing.

Domain	Library	fork	stars	commits	contributions
numeric Python	Numpy	4.7k	14.2k	23417	965
Mathematical Analysis	Scipy	3.3k	7.3k	23891	886
	Sympy	3.1k	7.1k	43892	887
	cvxpy	622	2.6k	2823	91
	FEniCS				7
Signal Processing	PyWavelets	291	919		4
Statistical Computing	pandas	10.5k	25.7k	23574	20021
	Stats models	2k	5.3k	13138	100
	Seaborn	1.3k	7.5k	2650	123
Mathematical And Text Analysis	NLTK	2.4k	9k	13906	307
wrapping multidimensional scientific data	xarray	558	1.7k	3217	237
Python binding for the R	Rpy2	9	15	3068	25
symbolic computing	sympy				
Image processing	scikit-image	1.6k	3.8k	12191	376
	Opencv	37.3	4.1k	28923	1193

ISSN (Online) 2394-6849

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 7, Issue 9, September 2020

		k			
machine learning	Scikit-learn	20k	41.4k	1802	25816
Quantum Computing	QuTiP	364	859	5632	77
	PyQuil	281	1k	983	80
3-D Visualization	Mayavi	223	740	3023	53
	napari	113	619	759	41
	Matplotlib	5.2k	11.8k	35411	947
Astronomy Processes	AstroPy	1.2k	2.5k	28878	361
	SunPy	377	509	13359	139
	SpacePy	27	59	2690	12
Cognitive Psychology	PsychoPy	551	897	11297	121
Bioin formatics	Biopython	1.2k	2.3k	14238	259
	Scikit-Bio	203	559	5588	59
	PyEnsembl	44	224	508	15
Bayesian Inference	PyStan	181	821	879	38
	PyMC3	1.6k	3.8k	12191	376
Simulation Modeling	PyDSTool	49	115	985	14
Multi-variate Analysis	PyChem				
Geographic Processing	Shapely	351	1.9k	1441	98
	GeoPandas	501	2.2k	1233	115
	Folium	1.7k	4.8k	1442	109

Table I

Above table shows useful information and detailed analysis of different python libraries and main use in different domains of science and computation. Table also lists information regarding usage of the same libraries using GITHUB platform. Information is based on commits and contributors for different Python libraries and the number of stars, and forks activity on the repository of particular library. Activity also shows the number of contributing authors and the number of commits in the July month 2020.

B. Ecosystem of python for scientific computing

From the past few years' python is majorly used in all around the globe in many organizations to solve from simple to most complex application problems. Being very simple to implement and equipped with many rich libraries and supporting modules, python is used in the area of scientific computing to solve many different mathematical concepts. In this section we are going to discuss the main libraries which are core to solving all the mathematical related problems in computing any problem.

1. **Numpy**-Less complicated environment of Python makes development of scientific applications less complicated, and more efficient. Currently, NumPy and scipy are two being rich numerical libraries in this field most commonly and frequently used. Numpy used mainly to compute high-level mathematical functions. It forms core library for scientific computing. It conquers slow running algorithms on python by utilizing multidimensional arrays and functions that operate on arrays. Algorithms can also be expressed in terms of function in NumPy. Numpy also offers implementations on huge amount of data in solving modern mathematical applications.Numpy also comes with the implementation of functionalities which includes manipulation of logical shapes, general linear algebra, discrete Fourier transform, which are commonly part of computing any mathematical problem. It can also be used to perform variety of mathematical operations from matrix operations to finding sine and cosine, to solve linear equations, Fourier transform, shape manipulation.

2. **SciPy-** Commonly called as scientific python this SciPy library mainly utilized for systematic computation. Scientific python is appropriate for logical application improvement. SciPy combine utilizes the NumPy for progressively scientific figurings.SciPy additionally utilizes NumPy clusters as the essential information structure to actualize a stack and connected rundown application. Built on NumPy, SciPy is more focused on scientific programming which includes examples of solving linear algebra, numerical integration ,interpolation ,optimization, distributions and random number generation , signal processing. SciPy comes mainly with many types of sub packages which help to solve most common issues in scientific computing.
3. **Matplotlib-**Matplotlib is a plotting library built on NumPy mainly used for developing an interactive and visualizing over data. Matplotlib is suitable for creating figures for scientific applications and through coding all parts of the figure can be controlled. This is increases reproducibility helps to regenerate the figure with updated data or change its appearance. It is used heavily in scientific computing for visualizations of data. Many huge third party packages are extended and constructed on Matplotlib

ISSN (Online) 2394-6849

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 7, Issue 9, September 2020

which produces quality visualized pictures over data just by using less number of lines of code. Many options for visualizing data in python are also available. Most popular among them are pandas, Seaborn,ggplot,pygal,plotly.

4. Sympy - Symbolic algebra in Python
SymPyis library for arbitrary which incorporated with various scientific constants, for example, pi, e, oo for unendingness. To assess an articulation numerically we can utilize the capacity (or N). It holds a contention which determines the quantity of critical digits. It used to perform algebraic manipulations of expressions. Along with these it also to solve calculus, integrations, limits, linear algebra .In order to extend with custom functions SymPy can be embedded in other applications.
5.Stats models-Stats models are a python module that gives classes and capacities to the estimation of various measurable models, just as for leading factual tests, and measurable information investigation. It provides mainly classes and functions for the estimation of many different statistical models. A broad rundown of result insights are accessible for every estimator. The outcomes are tried against existing measurable bundles to guarantee that they are right. The bundle is discharged under the open source Modified BSD (3-provision) permit. The online documentation is facilitated at statsmodels.org.Build on the top of NumPy this module uses to solve many complex mathematical problems like Regression and Linear Models, Time Series Analysis, and many Statistical tests.

6.Sympy-Sympy is a package for symbolic solution in python that can be used to solve set of equations. Commonly called as symbolic python is having huge number of symbolic tools to manipulate which ranges from simple algebra to Gröbner bases.
These are some of the main core libraries used in solving mathematical concepts used any scientific applications. Some of the heavily and frequently used common concepts of mathematics are discussed in the following section.

III. Mathematical concepts which are used in scientific computing:

Linear equations: In most of the applications in scientific computing and applied mathematics, commonly used mathematical components are the system of linear algebraic equations. These equations contains the set of variables like, 8x+3y-2z=9, -4x+7y-5z=15, 3x+4y-12z=35**.**The system of linear equations (1) can be solved by Numpy module which usnp.ling.solve () function to solve set of equations for defined variables x, y, z

```
import numpy as np

A = np.array([[8, 3, -2], [-4, 7, 5], [3, 4, -12]])
b = np.array([9, 15, 35])
x = np.linalg.solve(A, b)
x
```

```
array([-0.58226371,  3.22870478, -1.98599767])
```

The system of equation (1) can also be solved by using Sympy module

```
from sympy import *
x, y,z = symbols(['x', 'y','z'])
system = [
    Eq(8*x + 3*y-2*z, 9),
    Eq(-4*x + 7*y+5*z, 15),
    Eq(3*x+4*y-12*z,35)
]
soln = solve(system, [x, y,z])
print(soln)
```

```
{x: -499/857, y: 2767/857, z: -1702/857}
```

Linear regression: The process of finding the best fit straight line to the observed data is called linear regression and the line is called line of regression. Example, to relate the relationship between the population and use of available resources and the relation between weight and heights of students are the examples model of linear regression. The general form of linear regression equation for the above example is given by: $y = b_1x_1 + c$, where x_1 is independent (predictor) variable and y is dependent (response) variable. It's one of the important, simple and commonly used models. Its advantage lies in its simplicity in interpreting results. Till now, in simple linear regression, a single variable which is independent is used to model the dependent variable Y. But In many applications, there is more than one element that affects the response.

ISSN (Online) 2394-6849

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 7, Issue 9, September 2020

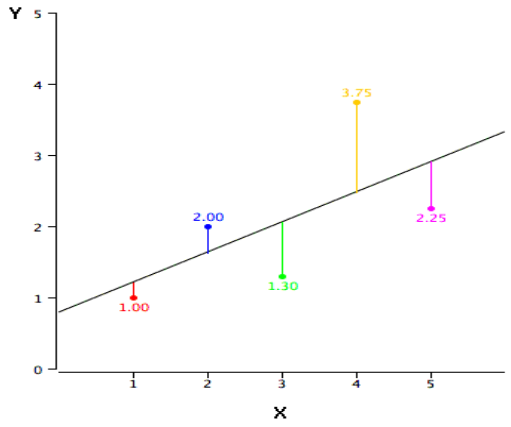

Figure 2

The predictions are represented by black line the points are showing the actual data, and the vertical lines on black lines represent prediction errors. In figure 2, represented by black line(Diagonal line) is the line of regression and it consists the values of Y for each corresponding value of X. In figure 2 the red color point is very near to the regression line and its prediction of error is least. And color point which is shown in yellow is more distant from the line of regression and its prediction of error is more.

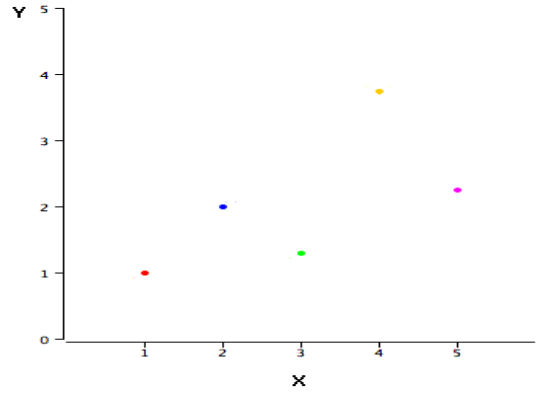

Figure 1

A scatter points are the points of data. Predictions are represented by the black line, the points are the particular data, and the vertical lines to the black line represent prediction errors. Therefore "best-fitting line." is the line that minimizes the sum of the squared prediction errors. That is the criterion that we use to find the line in Figure2.

Multiple regressions:

It is an extension of linear regression. **Multiple regression**usually define as the relation between dependent and independent variable. A dependent variable is expressed as the function of many independent variables with their coefficients with one constant term. Multiple regressions have more than two independent variables and it is called as multiple regressions. Multiple regressions is used estimate the value of dependent variable using the known values of independent variable and it is used to test specific assumption about whether to and to what extent certain independent variables explain the variations in a dependent variable of interest. Regression on being most powerful statistical method that recognize the relation between two or more qualitative variables.Ex.The price of a house is depends on the attractness of the location, the number of rooms, the number of bathrooms, built year, the size of constructed area and a number of other factors and this explained example takes multiple regression equation in the general form as $y = b_1x_1 + b_2x_2 + \ldots + b_nx_n + c$.Here, b_i's (i=1, 2…n) are the regression coefficients, which represent the value at which the dependable variable changes when the independent variable changes. While it is possible to estimate the parameters of more complex linear models with many methods, the computations become very difficult very quickly.

The routes by which one can fit the Regression lines numerically:

Analysis of fitting linear line observed set of points. It mainly has three steps

(1) Analysis of the correlation and directionality of the observed data,

(2) Calculating the given model nothing but fitting of a line,

(3) Assessing the viability and profitability of the model.

Constrained optimization:

Many engineering design and decision making problems have an objective of optimizing a function and simultaneously have a requirement for satisfying some constraints arising due to space, strength, or stability considerations. So, constrained optimization refers to the process of optimizing an objective function with respect to some variables in the presence of constraint of those variables.

Selection of methods for solving this particular type of constrained nonlinear optimization problems are listed here. In the parlance of mathematical optimization, there are two routes by which one can find the optimum (Numerically):

1. Using Direct Search methods: Here only the function values are used at a given point to find the optimum. It works by comparing the function values in a neighborhood of a point and subsequently moving in the

ISSN (Online) 2394-6849

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 7, Issue 9, September 2020

direction that leads to a decrease in the function value (for minimization problems). Direct Search methods are typically used when the function is discontinuous, and hence the derivate is not available at that point.
2. Use of methods based on gradient: Here the first and second-order derivatives are used to locate the optima. These methods take the gradient information into account and thus have the advantage of converging faster to the optima.

Polynomial Interpolation:
Polynomial interpolation is a method where we estimate values between data pointsknown. The interpolation has vast application not only in statistics, but also in science, business, or whenever there is a need to find the values that lays within two given data points.• Polynomial interpolation. In classical times, such methods were used to construct tables with values of complicated functions, like the logarithm, trigonometric functions, and more recently statistical density functions. These lookup tables, as they are now called, had a wide range of practical applications, from computations in celestial mechanics to nautical navigation. There are several methods to interpolate the data. Some of the methods are as follows Newton's Gregory, Newton's divided difference formula, etc. In python scipy.interpolate is a great way for function creation centered on fixed points of data using scipy library. Simple representation of python is as shown below

```
import numpy as np
from scipy import interpolate
import matplotlib.pyplot as plt
a = np.linspace(0, 8, 12)
b = np.cos(a**2/3+4)
print(a,b)
plt.plot(a,b,'r')
plt.show()
```

Extrapolation: It is an extension of interpolation for estimation beyond the range of values of the independent variable.A number of methods available for both interpolation and extrapolation.
The version 0.17.0 0f scipy, there is a option scipy.interpolate.intrp1d which allows extrapolation. Modifying the code in the following way gives:

```
import numpy as np
from scipy import interpolate

x = np.arange(0,10)
y = np.exp(-x/3.0)
f = interpolate.interp1d(x, y, fill_value='extrapolate')

a=f(9)
print(a)
b=f(11)
print(b)
```

Some of the scientific computing applications which are built on using mathematical concepts are: In the Design of linear-phase FIR filters using many problems in health care domain are being solved by operational research method. The topology optimization is used in micromechanical resonator design. Regression analysis has majorly used in developing many applications like optimize business processes and prediction models.It has also used when one need to find some uncovered patterns in data and transforming data into useful information.

IV .MEASUREMENT OF COMPUTATIONAL SPEEDS IN SCIENTIFIC COMPUTING:

Many mathematical concepts and methods used to quantify data analysis used in scientific applications. one among is regression analysis which has many applications in engineering and others, like data designing, weather forecast predication, climate research, astrophysics, etc. We focus our study on comparing computational speed between two different higher level languages python and MATLAB. Reason for choosing MATLAB with comparing with python as MATLAB stands as multi paradigm numerical computing environment which is enclosed with standard library and used to perform various mathematical related applications.
A) MATLAB:
MATLAB and python both programming languages are widely used to solve wide range of applications in real time. Both languages provide quick and easy ways to code for many scientists who are not so familiar with programming/coding techniques.AsMATLAB stands for numerical computing environment and easy to learn. MATLAB is built on closed ecosystem which does not require external libraries. All usable commands are available in core installation or toolbox.
B) Python
Unlike MATLAB, python requires external libraries to be imported to perform different type of tasks oroperations

ISSN (Online) 2394-6849

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 7, Issue 9, September 2020

depending on the type of domain as we listed in our table[reference table number to be given here].This can be done just using simple import command which brings flexibility in installing any third party packages or libraries. Different modules are discussed in section II of this paper. In our experimental study while implementing regression analysis we used numpy,pandas and sklearn on python.

Here we used Jupyter notebook to run Python.Jupyter is an open-source and program based instrument that incorporates many libraries.AJupyter notebook can work either locally or on the cloud. Each archive is made out of different cells, where every cell contains content language or markdown code, and the yield is implanted in the record. Common place yields incorporate content, tables, diagrams, and illustrations. Utilizing this innovation makes simpler to share and repeat logical works since the examinations and results are introduced in an independent way.

Our focus of study contains computing the runtime comparison on these two programming language on regression analysis. Here we perform regression analysis on chosen real time data set. The data represents marks of 2364 students over a provided set of sixquestions of seven marks each. The problem is solved in two programming languages MATLAB and python. The timing analysis is performed by using methods provided by both languages. In MATLAB we used tic()/tok() to compute the elapsed time where as in python importing time() module which provides functions to perform various time related operations.

To make our study accurate we put maximum care while coding the given problem in both the languages. The code we used here uses same set of exact data and python and MATLAB code almost kept same both keeping performance in mind and also to justify our study on these two languages.

V.CPU RESULTS:

Set of CPU results were obtained from different configured set of machines. Specification of the three test environment are used in the preliminary performance evaluation: Workstation, mainstream systems 1 and 2 are detailed in below table 2

Test Environment	CPU Clock	Hard Disk	RAM
Mainstream Machine1	3.00 GHz	750GB	16GB
Mainstream Machine2	1.70GHz	320GB	6 GB
Workstation Dell Precision 7920 Rack	1.70 GHz * 6	1TB SATA solid state drive	128GB

Table II: Test Environment Specification

Below table briefs out software versions of MATLAB and Python and its libraries used in all the mainstream machines and workstation.

Language	Version	Free available
MATLAB	2020a	No
Python	3.7	Yes
NumPy	1.17.4	Yes
SciPy	1.3.2	Yes
sklearn	0.20.1.	Yes

TABLE III: Versions of the Software used

Each of the tests was run for specific number of times over on complex data. Uncertainty bars were generated using the elapsed time of each of the recorded times. Here we have chosen different set of systems configuration to know more about the computational speed in depth. How much exactly difference in elapsed time while performing regression analysis over same set of data on each of these systems.

Details of operations performed:

We performed regression analysis using python and MATLAB both on same set of data on different set of configured machines as described above. Here we performed finding coefficient using multiple regression analysis operation on same set of data and noted down the running time and average running time is calculated. The first set of operation of regression is performed on both the modeled machines using same method of data loading. The operation is coded in python and repeated for 25 times for finding the coefficients using regression.

ISSN (Online) 2394-6849

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 7, Issue 9, September 2020

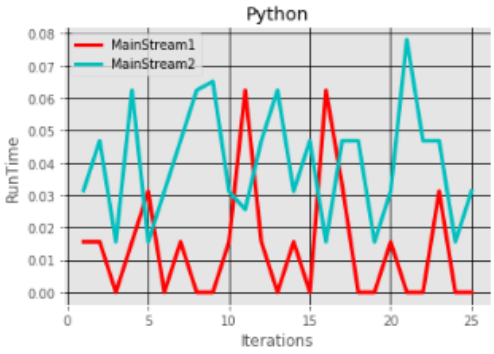

Figure 1

Fig.1 shows Runtime of Python for regression analysis using both mainstream 1 and mainstream 2 system configurations. From the figure it is noted that average performance of mainstream machine 1 is almost 4x times the mainstream machine 2.

Similarly we executed the operation on MATLAB on both the defined modeled machines mainstream 1 and mainstream 2.The run time of MATLAB on these two machines over the same data is observed and noted for the same set of 25 epchos iterations.

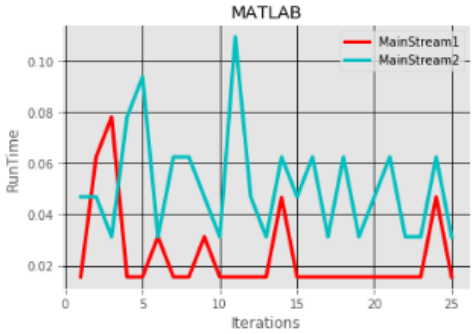

Figure 2

Fig.2.Runtime vs Number of iterations for operation regression using MATLAB.Its observed that Mainstream machine 1 works on and average 2x times faster than the mainstream machine 2 for MATLAB.

Next set of operations were performed on to study the speed of both the languages in each of the machines. In this study we tried to observe the how much exactly variations in run time takes place and to look down and compare any of the programming language

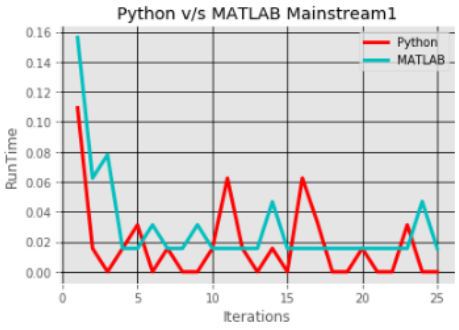

Figure 3

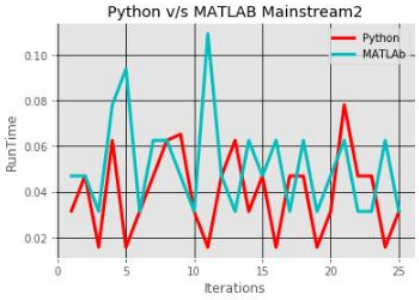

Figure 4

The Figure 3 and Figure 4 both represent Runtime vs Python and MATLAB on both the mainstream machines. The results shows that both the programs outperform almost equal. The difference between average elapsed time remains little varied and our study shows python used over its libraries numpy,pandas and sklearn can outperform MATLAB.

We continued our study still more in deeper and performed the operation on above defined workstation for the configuration defined in table I. The set of results obtained by performing regression analysis using the same set of data and coding but running it on workstation.

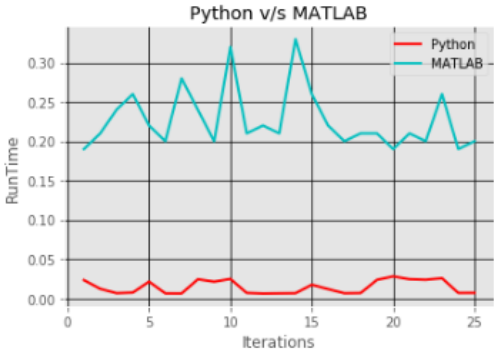

Figure 5

Fig 5: Runtime vs. number of Iterations performing for regression analysis using MATLAB and Python on the

ISSN (Online) 2394-6849

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 7, Issue 9, September 2020

same set of chosen data. Each of this operation was repeated 25 times for on same set of data both in MATLAB and PYTHON. The results of this operation can be seen in Fig.5.The python used over its libraries NumPy, pandas, sklearn.Table III shows the average running time in seconds on all three models of the machines. We can see here that there is not much difference between run time of python and MATLAB in mainstream model machine I and II but for the same set of operation we can see noticeable difference while running on workstation. Python when used over its libraries performs faster than the MATLAB.

Test Environment	Average Performance Time(In Seconds)	
	Python	MATLAB
Mainstream 1	0.011325	0.029356
Mainstream 2	0.039493	0.051276
Workstation	0.014446	0.2272

TABLE IV: Average Speed of Python and MATLAB for Testing Environment used

Many emerging young scientists and research scholars face many challenges while doing research. One common problem is lack of available resources needed. Which is may be due to limited access people, resources or denied access. While working on python using Jupiter notebook "Memory Error" – that all too familiar dreaded message while executing many programs having large and complex data set. We continued our study focusing on this and we repeated same operation on this platform. To describe about Google Colab, its online browser-based ,a free online cloud platform based on Jupyter notebook environment that allows to conduct and share complex experiment[8].It overcomes the restriction of poor computational power due to hardware limitations on our machines and provides same resources in any of the computer irrespective of any configuration. The configurations we tested for mainstream machine 1 and 2 as listed in table II may not be sufficient or even may take longer time to perform the operation when data becomes more and more complex. In this regard when resources are limited one can adopt Google Colab which operates at Intel xeon processor two cores with 2.30 GHz and 13 GB RAM and givesout 12 Hours of execution time continuously.

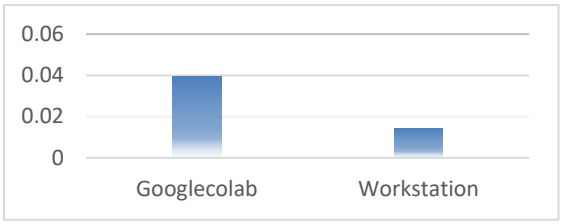

Figure 6

We continued our study and carried our set of operation on Google Colab and results obtained. From it we drawn a comparison between the average elapse times as shown in the figure 6.From the above figure it shows workstation performance nearly 3times lesser the Google Colab.This Google Colab surely act as survivor for solving many applications where researcher is economically week or having restricted access to the resources.

VI.CONCLUSION

Python being most friendly and easy coding language can make scientists adoptable to use it to perform various mathematical operations in computing many scientific applications. Many of the scientists may have not received formal computer science training and may not be good in programming. Python known as beginner's languageand having huge variety of libraries can be more helpful for scientists to perform different type of operations on scientific applications. It also provides significant speed in comparison with MATLAB. While one has to consider all the factors while deciding /choosing programming language, python clearly out performs for the above defined operation. Our focus of study here limited to choosing only one set of operation that is on different set of configured machines. Python also can be run on many online cloud platforms with dedicated hardware and resources unlike MATLAB which can help many scientists and research scholars to eliminate many barriers of availability of resources. This study shows how python provides significant speed up when used over its high performing libraries and can provide well support environment to solve scientific applications.

REFERENCES

[1] A. Nagpal and G. Gabrani, "Python for Data Analytics , Scientific and Technical Applications," 1999.

[2] J. Unpingco, "Some Comparative Benchmarks for Linear Algebra Computations in MATLAB and Scientific Python," pp. 503–505, 2008, doi: 10.1109/DoD.HPCMP.UGC.2008.49.

[3] R. Chudoba, R. Rypl, and M. Vorechovsky, "Using Python for scientific computing : Efficient and flexible evaluation of the statistical characteristics of functions with multivariate random inputs Using Python for scientific computing : efficient and flexible evaluation of the statistical characteristics of functions with multivariate random inputs," no. February, 2013, doi: 10.1016/j.cpc.2012.08.021.

[4] T. Oliphant and C. Analytics, "Python for Scientific Computing," no. December, 2014, doi: 10.1109/MCSE.2007.58.

[5] N. Ari and N. M. Mscs, "Symbolic python," pp. 1–8, 2014.

[6] A. Sheela, "Combination of NumPy , SciPy and Matplotlib / Pylab - a good alternative methodology to MATLAB - A Comparative analysis."

[7] O. Important *et al.*, "MATLAB vs Python : Why and How to Make the Switch MATLAB vs Python : Comparing Features and Philosophy," pp. 1–54, 2020.

[8] https://www.analyticsvidhya.com/blog/2020/03/google-colab-machine-learning-deep-learning/