International Journal of Management, IT & Engineering
Vol. 8 Issue 8, August 2018,
ISSN: 2249-0558 Impact Factor: 7.119
Journal Homepage: http://www.ijmra.us, Email: editorijmie@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

Python vs. R Programming Language

Kalyan Sudhakar

Abstract

The research paper mainly concentrates on the comparison of R and Python programming language. More precisely, it outlines the similarities that the two shares with respect to predictive modeling, deep learning and the different library packages. It also goes a long way to contrast the same regarding different features. In fact, it will dwell on the named topics and relation to statistics including matrix multiplication at some point. The topics, therefore, provide detailed information about the two languages. The two languages have various keywords that assist in distinguishing them. They include null that is a special constant in describing the absence of a value. The assert keyword is a major keyword which is used for debugging purposes in the languages. Finally, the study will also reveal several examples and different areas of application of both R and Python.

Keywords: R, Python Data science, Numpy, Scipy, Pandas, Deep Learning

1. Introduction

Python, which is an object programming language, shares many similarities with PERL. It is popular among the programmers due to its conciseness and readability of its codes. Python is relatively flexible and supports easiness of learning since its statements can execute on various operating systems. The conventional operating systems that support this include UNIX, Macintosh, and other various functions from Windows. Guido Van Rassum is the person behind the creation of the Python language. The source code is open source that even users can easily modify.

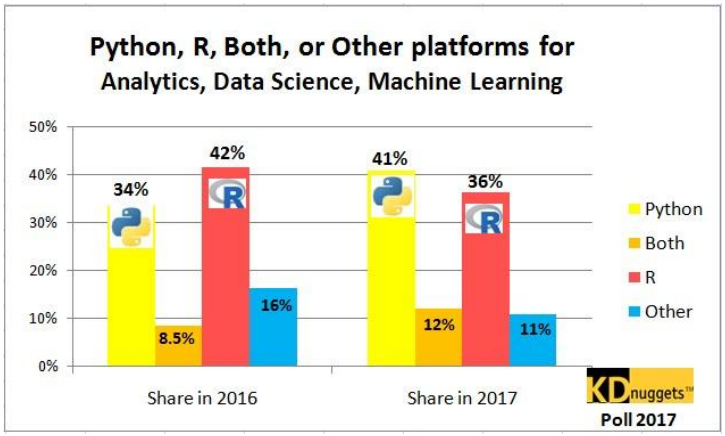
Indentation is one of the features that Python supports significantly. This makes the codes easier to read and comprehend. The language comes along with a dynamic data type, exceptional classes, and various interfaces that refer to available system calls and libraries. [3] Due to robustness, it can be extended using other languages including the C and C++. Python enjoys a lot of versatility when it comes to applications. For example, it has written popular applications and servers including Z object Publishing Environment, which is popular in many computing fields. Additionally, in design, it also has a share considering the scoreboard system that features for the Australia Cricket Ground. All these and many more applications illustrate that Python is an excellent programing language.

In case of R programming language, it is a product of Gnu project and equates to S language that is first came from Bell Laboratories in courtesy of Chamber John in conjunction with other stakeholders. R enjoys immense respect when it comes to graphical and statistical methods of programming. After using R, produces quality results and quality plots that entail many mathematical symbols and formulas. In fact, special care has been resolved over the defaults for small minor design selections in the case of graphics [5]. Under the terms and conditions of free software and the General Public License, R is available in the form of source code ready for installation. During compilation, any R program will often run on an array of operating systems including Windows, MacOS, and the like.

The working environment that R provides is a full stack integration of software facilities which are purposely for manipulating data, calculating and displaying graphically. Often, it will include the following in

R programming, in one way or another, plays a double role in data handling and storage. It acts as the best suite of operations and calculations especially in the case of multiplying matrices and aids significantly in the analysis of data. R programming also acts as a graphical facility for data analysis and displaying on either screens or hard copies. R programming is an effective programming language that involves the use of loops.

The environment in programming provides one with the knowledge of other languages. The advantages of R programming is that it is designed within a full computer language hence enables the user to execute additional functions. The language provides a platform where statistical techniques are implemented by using algorithms.



2. Research Methods

The data contained in the paper is subject to several techniques of data collections. The data is mainly obtained from the secondary sources. For the case of comparing R and Python programming, the data is heavily from online sources including journals and books that give the detailed information. Also, scraping websites forms the basis of secondary information. To make it more precise there are reference sources at the end of the paper to illustrate the publications used. These publications form the basis of sources of information about the programming languages. The inclusion to use the secondary source is that it provides detailed

information about the programming languages which other languages fail to provide. The reason to use this method is that it gives detailed information which is more accurate than other methods.

Advantages of Secondary Sources.

The method is time-saving precise information is available via engines, which makes the process very fast.

The method is accessible in collecting data within a short period of time. The online sources are more than enough to obtain a vast amount of information.

It helps to save money. It is quite less expensive than other ways of collecting data. Hence it is the most appropriate method.

Hence, the methods assist in getting the results of the two languages.

3. Results and Analysis

After collecting the data from the secondary sources, it helps in determining the results and analyzing it.

The results include the following.

Areas of Application

Areas of application of Python are very diverse. Python is used in many application domains. It offers many choices in web and internet development. It is used as frameworks such as Django and the pyramid. Also, in designing micro-frameworks such as the flask. Also used in advanced content management systems like Django content management system. Python's standard library supported many internet protocols like the hypertext markup language and extended markup language, JavaScript Object Notation, e-mail processing, it supports File transfer protocol, IMAP, and other internet protocols. Python has another package index which has more libraries. It requests a complex hypertext transfer protocol client library, and twisted Python forms a framework for asynchronous network programming. [2]

The second application of Python is in scientific and numeric computing. They include the software carpentry course which forms a foundation for educating about skills which assist coders to learn scientific computing and provides a platform where one can access the teaching materials.

Python is widely used in education in educating about programming both to the beginners and in other complex areas of Python as a language. Python books include the following Python programming and the practical programming. Another application of Python is in desktop graphical user interfaces. They include wxwidgets, Qt via pyqt. Also, it serves as a platform-specific toolkit are also available in Python they include the Microsoft foundation classes.

While the areas of application of R programming language includes the following, first it is used for statistical data analysis. Hence it enables in creating objects, functions, and packages Programs record the actions of analysis which enable you to update report, therefore, it can quickly try many ideas. Also, Google is making use of R programming for doing any form of statistics since the language is open source in its usage. R

and its libraries apply a diverse kind of graphical and the statistical parameters that consist of the linear and nonlinear modeling also the clustering and finally time sequence analysis [4].

The Core Libraries

Python and R programming have different core libraries. The various core libraries include the following NumPy, SciPy, and pandas that are different in the two languages. Additionally, they are as well different for the two languages.

The NumPy

Python NumPy is used as the library for Python language. Python also has large multidimensional arrays that are combined with the collections of very high-level mathematical functions that help to operate the arrays. NumPy uses CPython when referencing and executing Python codes as they consist of non-optimizing byte code interpreter.

Hence using NumPy in Python, you can compare it with R programming. Regarding the memory usage R uses a pass-by-value paradigm while the Python uses a pass-by-reference. The pass-by-value leads to code that is more intuitive while pass-by-reference helps to optimize the memory usage. When referring to the speed, pure Python is faster than the pure R programming when accessing the individual elements in the array. Also, NumPy Python is better in working with big data which requires on text processing and the shell -scripting.

The SciPy

Second, is the SciPy core library by considering SciPy Python lets you visualize using great graphics and has many numerical techniques build so that it can quickly analyze data and the theoretical idea that you may have. [4] It is an excellent front end to algorithms that are implemented in the C and Fortran. It serves as scientific niche so it is not very simple to install but it can be successfully installed on many versions of Windows, Linux and Mac OSX. SciPy offers tools for the data mining and the analysis that bolster Python's already -superlative learning usability.

The outstanding differences, which result from the SciPy, is that Python makes more sense hence it is widely used in the industries since it enables easier collaboration as a machine language. Python allows you to move on to projects in other fields when your machine data analysis project is done. Hence it is more preferred for production use since when the data analysis tasks require being integrated with web applications, hence you can use Python instead of integrating it with another language. For R it is a more significant data analysis tool, but it is limited regarding what to achieve beyond data analysis.

The Pandas

Considering pandas as a core library, it is entirely different in both Python and R Programming. Pandas in both languages are used to teach about Data ques. So, both contain data on NBA the primary difference between them is that is in the Python language you must import library packages like the pandas one to allow access to the standing data frames. The data frames are available in the two languages where there are 2D

arrays and each column can comprise different data types. Another difference is that both print the starting row of the data. However, in the case of Python, it is more object-oriented while R is more functional. Ahead is a method on the data frame object in both languages but R has a separate head function. Another difference occurs while finding the average of each statistic. In Python mean a method is what exposes the mean of the other ones by default while in R programming obtaining the mean of characters adjusts the results in NA mode. In making pairwise scatterplots, the matplotlib of the Python one serves as the central package for plotting whereas the seaborn is the layer that is often in use over matplotlib while R data science ecosystem consists of many smaller packages. Additionally, it also comprises other visualization packages. In Python, the visualization has only one mode of executing something while the R consists of many packages that enhance the various techniques of doing things.

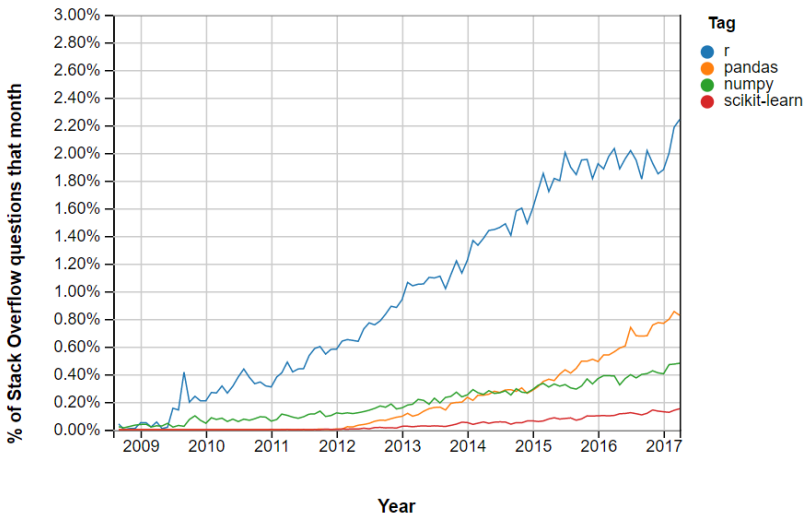
Another difference arises when comparing the clusters of the players by generating cluster plots. So that the cluster property must eliminate any of non-numeric columns with the lacking values. In R we achieve this by using a function in every column and eliminating it if it lacks some values or it is not numeric and then use the cluster package to look for K-means and obtain the clusters in the data and then we come up with a random seed using set. Seed so that we can get the results while in Python we use the Python machine learning package. Skit – learn is used fit a k means clustering model which gets our cluster models. The main difference is that in R we use to get- numeric – data and drop methods when removing numeric columns and the columns with the missing values. Both languages are used in plotting out the players by cluster with the main aim to get the patterns. In making scatter plots of the data in R the clusplot is used which is in the cluster library. To get PCA we use pccomp function, which is generated from R while in Python we use the PCA class, which is found in the Scikit-learn library and then use matplotlib to create the plot.

Both languages are used in splitting the data into training and the testing sets. In Python, the current version of pandas came up with a simple method which returns a certain part of the rows which are randomly selected from a source data frame which makes the code easier to understand while in R there are packages to make sampling simpler but are not much concise compared to using built-in sample function. The two languages assist in the univariate linear regression, which predicts the number of assists per player and the goals a player scores. The difference occurs that R language relies on the built-in and predict functions hence the predict will behave differently which depends on the kind of fitted model which is passed on it.

The two languages are used to calculate the summary statistics for the model but they achieve it differently. In R language the built-in summary is used to get information on the model while in Python there is the application of the statistical model package which allows statistical methods to be applied in Python. In general, we achieve almost the same results but it is complicated to achieve the statistical analysis in Python. Another difference is some statistical methods that are found in R are not found in Python. Both languages assist in fitting a random forest model to ensure the data is linear. The main difference is that Scikit-learn consists of the unified interface, which is compatible with many algorithms in Python while in R it consists of very many smaller packages which consist of single algorithms which are always with inconsistent ways to

access them. Hence it results in a diversity of the algorithms but it is used well. Also, the two languages can be applied to calculate errors in Python the Scikit learn library consist of a variety of error metrics which can be used while in R it consists of smaller libraries which are used to calculate MSE.

The panda core library in both languages assists in downloading a webpage. Using Python, the request package makes downloading of the webpage easier which consist of API for very request types. While in R, RCurl it usually provides a very direct way to make requests. After downloading the web page there is the need to extract the player box scores which the two languages assist by doing it differently. In R the code is more complicated compared to the Python code since in R there is no convenient way to use the regular expressions while selecting the items hence the need to use another parsing to acquire team names from the hypertext markup language R also discourages the use of for loops and instead apply functions along the vectors. Another difference is that rvest is the widely used web scrapping package in R to extract the data we need while in Python we use Beautiful Soup as the web scraping package. It allows us to loop through the tags and then come up with a list of lists in a direct way.



Predictive Modeling

In the case of both R and Python, the Predictive modeling is used widely for various prediction purposes. Among the many forms of models available in data science, predictive modeling is a straightforward way of creating models that can predict future happenings and behaviors. It compares both languages whereas in R it mainly dwells on statistics. In prediction in areas like weather forecasting, a statistical model is as a result used. With this predictive attribute, the main role of predictive modeling in both R and Python is to come up with models that identify a future occurrence from a given set of data.

The relationship between this form of modeling and the two languages is explicit with factors that compare and contrast them respectively. In relation to R that is often described as a statistical language, predictive modeling favors it more compared to Python. Therefore, implementing is quite easy when working with R. As for the latter, it may not be fully efficient to implement it. In fact, when one is building the

predictive model with Python it may be quite laborious considering importation and installation of packages will have to occur for every stage. Precisely, Pandas that is imported as Pd is one of the library packages that are often in use in this regard.

Numpy is also another equally significant package that equates to Pandas and it comes in as np during importation. For predictive modeling, during analysis and data manipulation all these packages provide an ideal environment for tabling some of the high performing data structures. When building the model in reference to Python the steps are straightforward but they also assume competence and prior knowledge of proper knowledge of the language syntax and execution. First, everything begins with installation and setting up of the required library package and in this case, the Pandas (pd) which has to be imported. To carry out this phase efficiently the use of packages in high-level languages should be conversant to the modeler. Besides this, the whole idea will additionally entail working with different categories of data, indexing them, and more importantly to handle any missing data. The build-up of the model continues sequentially to other descending phases that come along with various accompaniers.

On the same note, the next step in creating the model pertains to analogy and modeling of available data. Afterward, the completed analysis is subject to an organization where similar sets are grouped together. More importantly, when summing up the model it is highly recommended to give all the results in the form of plots or any other suitable visualization means. One fact to underline all along is that the final predictive models in Python are must make use of Scikit-learn and the pandas [2]. By the end of this modeling in Python, it should be able to predict different future behaviors of data such as sequences. Specifically, on the other hand of R programming building, a predictive model is at least simplified considering that some steps like importing packages do not feature. What makes this possible is the fact that R is a statistic-centric language that has its own statistical full package of libraries that illustrate some of the conventional data algorithms.

However, when building a predictive model in R, this should never curtail from installing new packages bearing in mind the flexibility that allows installation of other new ones that assist in the analysis of data that can take any form. More precisely, when building the model it is possible to import any dataset to the R environment with the aid of read.csv, which is a preloaded function of the language. Now, considering that R is a prominent language when coming up with a predictive model, there are distinct ways to go about the whole idea of coming up with it. The debut build-up begins with R installation that is an open source and is available for different operating systems including Windows, Linux, and OSX. Before starting, there is an unavoidable need of having a data set that covers the area of interest like hospital patients and such. Suppose that the data is readily available then the due task in this step will be to load in the environment.

Soon after having the data set loaded there is the most hectic part of predictive modeling using R. It is all about preparation and harmonization of the same data. What happens is that meaningless columns and rows are washed out and the whole process is made possible by setting oblivious values to null. Conversion of the data from one form to another is also part of the preparation stage. With the use of the numeric function, any form of data that is inconvertible will be set to as Not Applicable (NA). If all this is on point then with the aid

of the complete.cases function one can now check for an empty entry. It is no doubt that the predictive model should now be working after all this. Checking the predictive powers is what helps in determining the functionality of the model. To do this effectively some known values are checked against the expected values. If the two correspond, then it is a fully functional model. Merging up predictive modeling for Python and R there is a lot to contrast and compare between the two.

When doing a predictive modeling project R is undoubtedly better off as compared to the Python as seen all along in the case of build-up. In fact, data scientists will tell that when handling data sets, Python tackles smaller amounts as opposed to R that keeps on evolving and allows handling of large data sets that it stores on the computer RAM. Nevertheless, this fact could also be an apparent shortcoming that tends to limit R language to in-memory computations and more so when building the predictive models it could run on any server and give correct analysis. Besides this, in the case of computing both Python and R can connect successfully to Hadoop and function in a parallel mode. During coding of the predictive model, Python has an upper hand considering that its codes are elementary to interpret and write. More often than not, they are equal to literature notes and as a result, the preferences when doing predictive modeling are ideally many. As for the R part, the codes are seemingly tedious, and most people are never satisfied when building the model on its environment.

Various packages for use in Python and R also do compare and contrast the two significantly. Considering them for each case, it is explicit that the preloaded packages in R allow any data scientist to use the functions that are defined in the packages. On the other hand, when coming up with the predictive model on Python the idea is different since this is only attainable after installation of the packages. One similarity between them is that the two are well established when it comes to visualization of data. Independent and dependent variables are widely in use for both cases in prediction. What happens is that both the two have exclusive functions that identify the relationships between these variables in a useful manner. However, the entirety pertaining to the success of the idea in R and Python is subject to a lot of trial and failure by the various statisticians who use the model to predict some aspects of future data change.

The requisite necessities between the two languages also form a basis of similarity where something like a package cuts through R and Python in predictive modeling. The difference in respect to this is that in one that is R the libraries are readily available unlike in Python where they ought to be installed. Another similarity is that when coming up with the model a prior feasibility study is necessary for both. Its main objective is to understand the provided data sets, desired output and the various steps to follow in building the model. Data preparation before loading to the working environment is also inevitable when doing predictive modeling in both R and Python.

Loading in of data on any of the environment is done via the use of commands. The step is a quite crucial one where all aspects of the data set have to be captured accordingly without omission whatsoever. Moreover, it will also be interesting to sieve and harmonize the data to make sure that it is clean for the next stage. A simple way to go about this is harmonization. Exploration usually takes place in both as the subsequent step

where the representation of the analysis takes place graphically usually a plot. Eventually, creation and validation of the model, which is perhaps among the final stages, is what marks the authenticity of the prediction. If it is working convincingly in both cases, it is good practice to keep on experimenting and improving the model. Yes, that is all about the predictive model in R and Python.

Deep learning

Data science helps understand deep learning in details. In one way or another deep learning is a complement of machine learning. It works under the principle of reading concepts and has many applications including detection. In respect to R and Python, it differentiates as well as equates. The correlation between each of the language and deep learning is a precise one. In Python, it makes use of powerful libraries like the Keras one. To run the library, it will often start with installation using the PIP and install command. Setting that aside and having a look at the same concept of data learning in R will make you better than never. Surprisingly, in R, it also makes use of the powerful Keras library but additionally, it also makes use of the R interface. The overall concept is intuitive, but the implementation is what remains tedious. The striking fact about the idea of using Keras as a network API is only to understand that it is written in Python.

Since both R and Python capitalize on Keras, it is essential to have a better understanding of this exclusive library. Implementation of any in-depth learning program on Keras is subject to a systematic process with several steps. Loading of data is the foremost step that is succeeded by the definition of a suitable model. Compilation of the model occurs later together with its fitting. Evaluation and of everything and merging is perhaps the final step. Arguably, since the two forms of data learning are interconnected, then there are no striking differences between the two. Despite this, there may be some few, but the concept has dominant similarities in both cases. At first, it occurs that the idea of deep learning can serve a wide range of applications in both Python and R categories. Some of the most prominent results from this learning include applications in areas like speech recognition, self-driven cars, and shape detection.

The idea of having all concepts pertaining to deep learning in place makes it seem explicit. However, despite its relation to languages like R and Python programming, there are many facts that relate it to machine learning. Overall, it is no doubt that the whole idea of deep learning works well with large sets of data. In assumption, if there are some smaller sets, then its algorithm may as well not be able to perform the role exhaustively. Therefore, in the non-professional language, the performance of deep learning seemingly increases with an escalation in the scale of data. To make the relationship between Python, R and deep learning clearer it is right to argue that it also favors R since it is a statistical language with a capability of performing computation on large data sets to come up with equations like those of regression.

In the case of hardware dependency, data learning is also not an exception. Contrary to other forms, it is highly dependent on high-end machines. Purposely, its algorithms perform large computations of matrix multiplication [1]. Given that, the operations will often require integration of the GPU to trigger better results. GPU is for primarily serving the purpose and thus explaining the need for the GPU in the data learning algorithms. Just like its definition so is the role. For any given set deep leaning will go a long way in ensuring

that it extracts and learns high-level features from the provided sample. In fact, this is a plus when it comes to different forms of learning including machine and traditional ones. What deep learning does in this regard is that it aids significantly in reducing the chances of coming up with a new extractor for every problem. For example, it could learn different parts of a human face and in turn, come up with a high-level representation of the same.

There are ideally many examples in relation to R and Python just like any other high-level programming language. They both can feature in many areas including predictive modeling as discussed above. More precisely, a language like R can work well in areas that relate to statistics and analytics. Currently, in many varsities around the world, both R and Python are parts of the academic curriculum for students taking statistics and computer-related programs respectively. It is possible to come up with both simple and complex projects using any of the two. In R, it is specifically possible to code equations including a regression one whereas Python can achieve any output that equates to a high-level language. The robustness of Python is similar to those of dynamic languages like Java. The two have different syntax and modes of executions with that of R named as the most challenging one. Taking into consideration the concept of deep learning and predictive modeling R and Python can work collectively in many areas requiring critical handling and manipulation of data including tabulating of marks being one of the simplest areas of application.

4. Conclusion

In conclusion, the comparison of R and Python is explicit with numerous striking factors that tend to separate them. However, though they have distinct differences and similarities, many more features provide an intersection of the two. The requisite knowledge in using any of the two is deliberately minimal in that it favors both beginners and first-hand programmers. Furthermore, the ability to incorporate library packages and evaluate needs like prediction explains why Python and R are rich in relations. However, any of the two has distinct applications, syntax and different modes of executions. Notably, both R and Python can also work collectively in a parallel mode to achieve a common goal.

References

[1] Al-Rfou R, Alain G, Almahairi A,AngermuellerC, Bahdanau D, Ballas N, et al.Theano: A Python framework for fast computation of mathematical expressions. arXiv. 2016 May; 1605.02688v1:473.

[2] Diamond S and Boyd S. CVXPY: A Python-embedded modeling language for convex optimization.J Mach Learn Res. 2016 Apr; 17: 83.

[3] Haraldsson SO, Woodward JR, Brownlee AE,Cairns D. Exploring fitness and edit distance of mutated Python programs.In: McDermott J, Castelli M, Sekanina L, Haasdijk E, García-Sánchez P, editor. Genetic Programming: 20th European Conference; April 19-21, 2017; Amsterdam. Cham: Springer International Publishing;2017. pp. 19-34. European Conference on Genetic Programming. Springer, Cham, 2017.

[4] Hunter JD. Matplotlib: A 2D graphics environment. Computing in Science &Engineering. 2007 June; 9(3):93.

[5] Johansson R. Installation.In: Johansson R, editor. Numerical Python. Berkeley, CA:Apress; 2015. p. 476.