

# R vs Matlab vs Python: A Comparison of Multiparadigm Programming Languages

Sara Durrani (i19-2054)  
National University of Computer and Emerging Sciences  
Islamabad, Pakistan  
I192054@nu.edu.pk

**Abstract**—There are a lot of multiparadigm programming languages. Each programming language's design and implementation comes with different variations of features. In this paper, I have provided a comparison of three multiparadigm programming languages i.e Python, Matlab and R on the basis of their features. I have selected five features for comparison which are research, online help, scalability, data loading and reusability. The comparison includes selection of datasets, their features presentation and model development. The languages recommendations are given on the basis of technical research analysis.

**Keywords**—R, Matlab and Python

## I. INTRODUCTION

The selected languages contain their own features and development support. This paper compares the effectiveness of R, python and Matlab using a professional research development environment. The python is a programming language that contains large set of standard libraries[2]. These libraries are structured for supporting general programming languages and the modules of OS, databases, threading, networking and many more. Further, Matlab was developed for commercial as well as research development modules. Matlab is introduced as programming language as well as standard library[3]. It mostly provides functionality for plotting and big data processing. It also contains the toolkits and the premium versions cost the developers extra for them. R is mostly introduced as open source language mostly used for statistical evaluations and analysis[1]. R pushes the big data analysis boundaries and provides a lot of standard libraries for data visualizations and data pre-

processing. Shiny R studio provides integrated development environment to develop models for predictions and recognitions. A vital role R plays as a support in data science decision making process.

## II. DATASET

For evaluating the selected dataset features I have selected two datasets for different tasks. The main tasks I decide to perform on those datasets is classification. The first dataset I pick from UCI repository is open source available with the name as Bank Note Authentication. Other dataset is from Kaggle that is a huge dataset of cat and dog images. Banknote Authentication dataset contains values that are extracted from images of forged and original banknotes. The images were of size 400\*400 pixels. The resolution achieved is 660 dpi through using grey scale images. The tool uses to extract these dataset feature is wavelength transform tool.

The basic information about banknote dataset is presented in the table as follows:-

Data Characteristics	Multivariate
No of instances	1372
No of variables	5
Task Affiliated	Classification
Source	University of Applied Sciences
Data	August 2012

Table 1: Bank Note Authentication Information

The features of Banknote Authentication datasets are given along with their datatypes as follows:

Variance	Continuous
Skewness	Continuous
Curtosis	Continuous
Entropy	Continuous
Class	Integer

Table 2: Features and Data Types

Assira is the second dataset that I chose for recognition task. Assira is known as Animal species Image Recognition for Restricting Access. This dataset is generated by partnership between petfinder.com and Microsoft. It is subset of 3 million images of cats and dogs for the recognition purposes which is manually annotated. The state of the art model is developed in 2013 and published in [3]. It is further divided into training and testing data that is already available on the kaggle. The images size reshape to 200\*200 pixels for preprocessing. The attributes of Cat Dog (Assire) dataset are given along with their datatypes as follows:

Data Type	Image
Task Associated	Classification
Dog Images	12,500
Cat Images	12,500

Table 3: Cat and Dog Dataset Information

These two datasets carry great importance for analysis of languages and classification tasks.

### III. IMPLEMENTATION DETAILS

After the selection of dataset the models for classification are developed. The basic model development structure followed for this paper is as follows:-

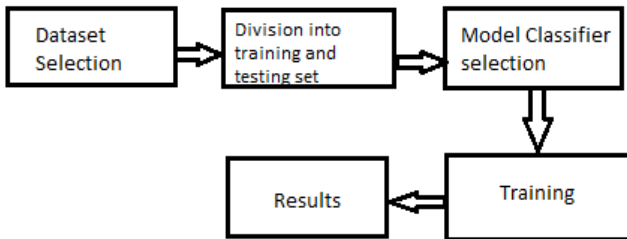


Figure 1: Block Diagram of Implementation Details

Hence, implementation is done with three different classifiers in these different language as R, Matlab and Python.

#### A. Language Features Analysis

This paper focuses on analyzing language features for the problem solved using R, MATLAB and python. The comparison between language feature design and implementation details is discussed in this section.

##### 1) Research/ Development point of View

Research remains a fundamental part of a technical project. Programming languages play a vital role in developing a required system. For a programmer or developer it matters a lot how much a language provides assistance in the task to be completed. While working with Assire dataset at the step of using built-in libraries for CNN dense layers, max pooling and 2D compression I tried to use language providing more support. Development point of view always contains the requirement of writing least lines of code. With python, it took least number of lines and a lot of built-in library support is available for this classification task. With R, it needs a whole bunch of libraries installation and their availability is required. The CNN model architecture also required a separate dense layer implementation that made work more hectic and tiresome. Using Matlab, the pertained model is used for recognition of the classification and recognition of test set of dataset. Further considering the second dataset of bank note authentication, development point of view recommends R due to its strong statistical analysis power. This dataset is more of values for different variable parameters. Python and Matlab are recommended as secondary recommendations due to the separate building of some KNN functions that are not build in available to tackle this kind of dataset. Hence, the research and development point of view varies with the type of dataset and the classifier selected.

##### 2) Online help

Analyzing the languages with respect to online help also provide a vast knowledge for selection of a programming knowledge. For my tasks of developing classification models for two different datasets I have found all kind of online help from specific language

development communities. By this experience I have found that python has largest development community available on the web. Matlab has second largest development community for helping in resolving technical errors. I have faced a lot of problems and difficulty finding, posting and resolving R language errors. As, the community of R is small and is in developing mode. R's community is expanding but as compared to other old and stable programming languages it lacks stability. For online help I used these platforms frequently:

- Stack overflow
- Github issues
- Maths Works
- Shiny R Ask

Moreover, the online tools are also available for all programming languages. IDEs are also available for R, Python and Matlab. Matlab's IDE takes a lot of time to install and functionality is not completely open source as versions vary a sudden change in functionality happen.

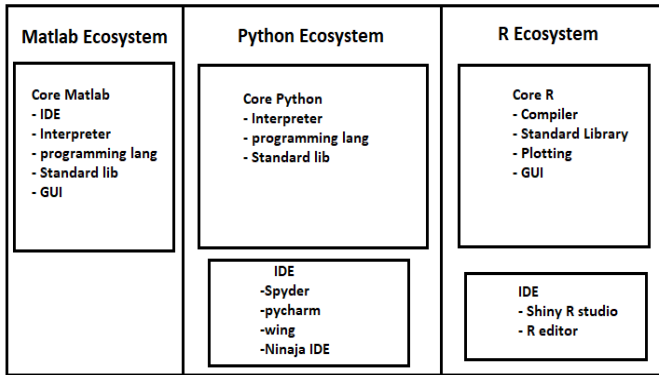


Figure 2: IDE's Comparison

### 3) Computation Complexity

Computation complexity of a programming language matters very much. The training of model in my selected methodology gives a lot of idea about computation complexity. The R has most high computational complexity handling power for classification task. After that python handles very high complex ratios. With respect to the models developed and classifiers selected the computational complexity is give in the table below:

SVM	KNN	CNN
$O(\max(n,d) \min(n,d)^2)$	$O(ndk)$	$O(n^5)$

Table 4: Computation Complexity

These computation complexities are picked from the literature review which are already proved. The paper [4] shows the SVM complete computation complexity along with calculated accuracy. Similarly, KNN and CNN's computational complexity is calculated in [5] and [6] respectively.

### 4) Data Loading

Data has the importance of gold for tasks like classification, prediction, recognition and analysis. Data loading for any one of the tasks requires to be smooth and easy. Python uses pandas library to read and load a data file depending on the type of data. For reading image data, Keras's framework's image library is available that accepts the folder's path. For R, read function is available to load a file having type csv. The data is acceptable in case of space or comma separated data in all languages. Among all, R provides very smooth, readable and presentable data visualizations. Along with that, matlab uses library of dlmread to read data and I use it to read csv of Banknote authentication dataset. As a recommendation, R is very suitable for data loading and representations.

### 5) Reuseability/ Maintainability

This feature of programming languages is important as maintenance is required at every level of software development cycle. All three of the languages I have worked with have this feature but python's importing structure and recalling of functions is very efficient. Along with that, the function definition is easy, smooth and accessible. R and Matlab are also functional languages and support functions but their declarations of functions are difficult to handle and remember. The maintenance of developed model is also easy with python as just replacing of data is enough but with R and Matlab I felt difficulty to handle data passing to the model developed due to the slow running speed and more time consuming scripts.

Hence, the overall language features availability and analysis can be observed through this table as given:

Features	Python	R	Matlab
Development point of View	✓		
Online help	✓		✓
Computation Complexity		✓	
Data Loading		✓	✓
Reusability	✓		

Table 5: Features Comparison

#### IV. RESULTS

The overall comparison of languages and accuracy results with appropriate classifiers is shown in the table as:

Language	R	Mat lab	Python
version	R-4.0.0	8.5 R2015a	3.7.7
Classifier	Assira: CNN Banknote: SVM	Assira: CNN Banknote: KNN	Assira: CNN Banknote: KNeighbors Classifier

Table 6: Comparison of languages with appropriate classifiers

##### A. Assire dataset results

For implementation in all languages the model parameters are kept same so as to maintain the input to the model. Following table represents the parameters selected for Assire dataset for model development:

Activation Layers	Relu, sigmoid
Target size	64*64
Batch size	32
Class mode	binary
labels	1-dog, 0-cat
Epochs	3

Table 7: Model Parameters

These parameters were passed to the model and as a result the loss and accuracy of model is measured. The accuracies attained for this dataset are as below:

Accuracy & Loss Values		
R	Python	Matlab
67.55%	70.09%	75%
0.6046	0.5661	0.5872

Table 8: Accuracy & loss Values

The following graphs show comparison of accuracy and loss during epochs with R, matlab and python.

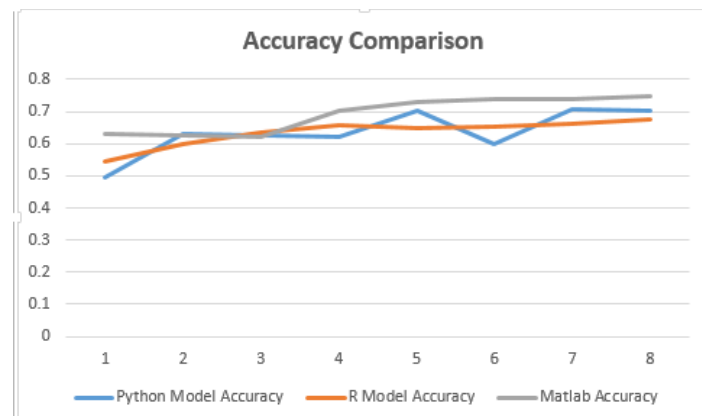


Figure 3: Accuracy Comparison

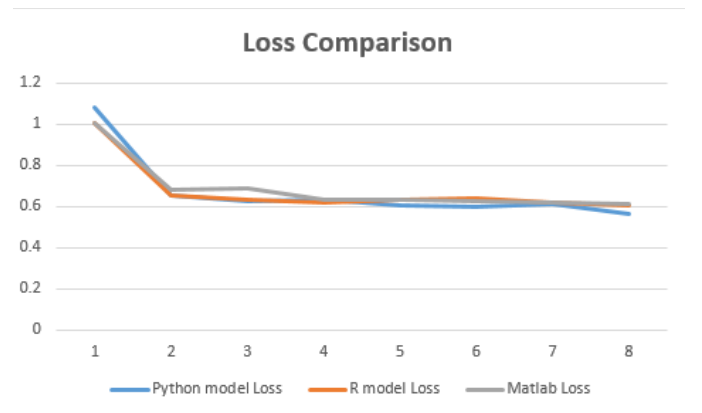


Figure 4: Loss Comparison

On the basis of following values of accuracy, the Matlab's model attained highest accuracy and after that python and R's accuracy is attained overall as given in figure 4.

##### B. Bank Note Authentication Dataset

The parameters to attain good accuracy and results are selected according to classifier used. For this

dataset three different classifiers are used and their results are compared in this section.

Accuracy Values		
R	Python	Matlab
92.7%	99.27%	99.8%

Table 9: Accuracy Values

The following graphs show comparison of accuracy with R, Matlab and python.

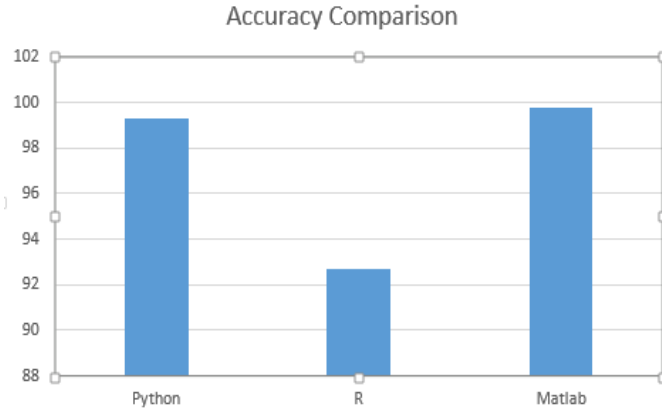


Figure 5: Accuracy Comparison

On the basis of following values of accuracy, the Matlab's model attained highest accuracy and after that Python and R's accuracy is attained overall as given in figure 5.

## V. CONCLUSION

In this paper, I have proposed a comparison of multiparadigm programming languages such as R, Matlab and python by selecting two different datasets. All model's analysis is done using five selected features of programming languages. Moreover, the results are also given and compared according to the accuracies and loss of models. This whole comparison helps in choosing programming language for a specific dataset and its development. All selected features of these three languages are briefly discussed. In future these languages are aimed to be compared with more features.

## VI. ACKNOWLEDGMENT

This paper is written under the supervision of Professor Dr. Labiba Fahad for the subject of Theory of programming languages. I want to thank my

parents and faculty members to support me in every phase of

learning to complete such researches and development of projects.

## VII. REFERENCES

- [1] Morandat, Floréal, Brandon Hill, Leo Osvald, and Jan Vitek. "Evaluating the design of the R language." In *European Conference on Object-Oriented Programming*, pp. 104-131. Springer, Berlin, Heidelberg, 2012.
- [2] Van Rossum, Guido. "Python Programming Language." In *USENIX annual technical conference*, vol. 41, p. 36. 2007.
- [3] Gilat, Amos. MATLAB: An introduction with Applications. *John Wiley & Sons*, 2009.
- [4] Chapelle, Olivier. "Training a support vector machine in the primal." *Neural computation* 19, no. 5 (2007): 1155-1178.v
- [5] Hajebi, Kiana, Yasin Abbasi-Yadkori, Hossein Shahbazi, and Hong Zhang. "Fast approximate nearest-neighbor search with k-nearest neighbor graph." In *Twenty-Second International Joint Conference on Artificial Intelligence*. 2011.
- [6] He, Kaiming, and Jian Sun. "Convolutional neural networks at constrained time cost." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5353-5360. 2015.