Journal of Engineering Technology and Applied Physics

An Overview of Python Libraries for Data Science

Ankush Joshi[1,*] and Haripriya Tiwari[2]
[1]*COER University Roorkee, Uttarakhand, India.*
[2]*Inspiration Public School, Haldwani, Uttarakhand, India.*
[*]*Corresponding author:* ankushjoshi1987@gmail.com, *ORCiD:* 0000-0003-0873-3340
https://doi.org/10.33093/jetap.2023.5.2.10
Manuscript Received: 20 March 2023, Accepted: 12 May 2023, Published: 15 September 2023

***Abstract* - In this era Python is the most popular as well as in -demand language for Data Science due to the number of libraries available for data processing, analysis and data visualization. The aim of this review paper is to give the overview of different available libraries. For this we grouped 48 different libraries in 3 different categories which are Data Collection, Data Analysis & Processing and Data Visualization. For comparison we use the GitHub community base (Stars, Forks and commits) as well as their properties and functionalities.**

Keywords— Python, Machine learning, Deep learning, Data science, Data analysis, Data visualization

I. INTRODUCTION

Data Science is the domain of study which deals with the very large amount of structured or unstructured data using various modern technical tools, statistics, artificial intelligence and machine learning techniques to produce and visualize meaningful information to help an organization to make business decisions. In the current era there are multiple forms of data which are in the form of text, image, numbers, audio and video. As a result, there is a requirement to handle this data and analyze it to produce some information by which any organization can make decisions easily which will be helpful for society as well as for the organization to grow well.

A complete Data Science process goes through some sub-processes which are Data Collection, Storage and Process, Data Analysis and the last one is Data Visualization. According to a survey conducted by Anaconda [1] out of 3104 respondents 63% of respondents always or frequently used Python, which makes it the most demanded language in 2021. Python provides a number of libraries which are freely available and anyone can use them for Data Science. The aim of this research paper is to provide an overview and pros & cons of these Python libraries.

According to the data science process we categorize these available libraries in three stages- Data Collection, Data Analysis and Processing, and Data Visualization.

II. OVERVIEW OF LIBRARIES

A. Data Collection

Data Collection plays a major role for any data science project. For scrapping and collecting the data python provides Beautiful Soup, Scrapy, Pattern, twitter-scraper, clevercsv, parse, camelotpy, MechanicalSoup, and many more data collection and extraction libraries which are presented in Table I and Fig. 1.

beautifulsoup4 [2], this python library is the most suitable and popular way to scrap any website and collect the data from its web pages.

textract [3] is used to extract the data from any type of file.

Scrapy [4] is the python framework which is used to scrap the data from multiple websites in a user required format.

Pattern [5] is used for web mining which provides the functionality to mine the data from Google, Twitter etc.

twitter-scraper [6], this provides the functionality to access the tweets from twitter. It retrieves the content with the help of requests package and parse that content with the help of beautifulsoup4.

clevercsv [7] provides the functionality to handle the unordered CSV files.

Parse is used for parsing a string according to the given format.

Published by MMU PRESS. URL: https://journals.mmupress.com/index.php/jetap/index

camelot-py [8] with the help of this library, we can extract a table from the file which is in pdf format. It accesses tables from only text based pdf document not from the scanned document.

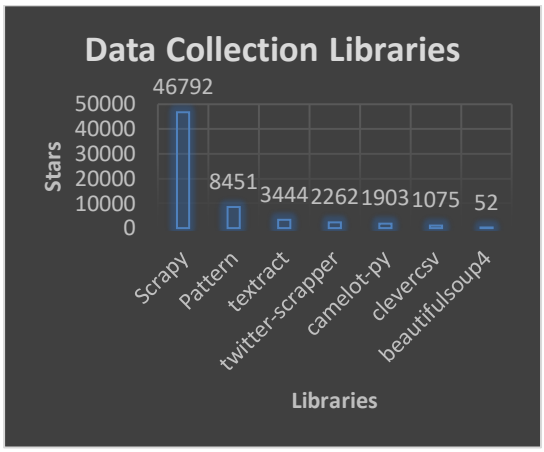

Fig. 1. Data collection libraries (Stars achieved data from GitHub).

Table I: Information of data collection libraries from GitHub.

Library	Stars	Forked	Commits
scrapy [4]	46792	9927	9730
pattern [5]	8451	1596	1434
textract [3]	3444	523	585
twitter-scrapper [6]	2262	574	262
camelot-py [8]	1903	347	599
clevercsv [7]	1075	60	800
beautifulsoup4 [2]	52	22	1

Note: Data collected till 12 April 2023.

B. Data Analysis & Processing

After collecting the data, it should be analyzed and processed to produce the information as it is a very crucial part of the data science process. At this stage, firstly the major errors like duplicate data, NULL data etc. will be removed from the collected data. This process is known as data cleaning. After that the cleaned data will be organized in a structured manner and then this structured data will be processed to generate the meaningful information. For this complete process python provides us lots of libraries such as- numpy, scipy, pandas, scikit-learn, statsmodels, pycaret, tensorflow, opencv, keras, pytorch, Cython, pytables, h5py, mlpy, shogun, caffe,mrjob, dumbo, hadoopy, pydoop, PySpark etc. These data processing and analysis libraries are divided into two categories- Machine Learning Libraries (in Table II and presented in Fig. 2) and Deep Learning Libraries (in Table III and presented in Fig. 3).

Machine Learning Libraries:

numpy [9] and scipy [10] are the essential packages for data science projects. These are the mathematical packages which provide the functionalities to solve most of the mathematical expressions or problems.

pandas [11] is a Python package that provides flexible data structures to work on labeled as well as relational data easily.

scikit-learn [12] is a module for machine learning which is used for predictive data analysis.

statsmodels [13] is a python module that provides the functionality to conduct statistical tests.

pycaret [14] is an open source machine learning library which allows data scientists to perform complex ML tasks easily and quickly by writing a few lines of code.

tensorflow [15] is an open source ML platform which helps the developers to solve real world ML problems by preparing the data, building pre-trained or custom models and deploying them in device or cloud.

opencv [16] is an open source library for image processing and machine learning. With the help of this we can process images or videos to find any object as well as identify the handwriting etc.

keras [17] is the mostly used framework for deep learning applications. This high level, deep learning API is developed for implementing neural networks by Google.

h5py [18] is used to work on HDF5 binary files. It helps to store and manipulate huge amounts of data with the help of numpy. The most important thing of this is that the files created are in standard binary format which can be easily distributed to the developers, even those who work on MATLAB also.

mlpy [19] uses numpy/scipy and GNU scientific libraries to solve ML problems. It is used for supervised and unsupervised machine learning applications. It provides the features- least square regression, ridge regression, last angle regression, elastic net, PLS regression, SVR regression, LDA classification, SVM classification, DLDA classification, K-means and Hierarchical clustering, LCS etc.

shogun [20] is C++ based python package for ML applications. It is less popular than tensorFlow or SkLearn, because it provides much less technical documentation and support compare to SkLearn.

mrjob [21], python package for Hadoop streaming jobs. mrjob supports EMR, Google Cloud Dataproc as well as local Hadoop clusters. With the help of this we can write multi step jobs.

dumbo [22] is a python module for Hadoop streaming programs, especially used for MapReduce codes.

hadoopy [23] is used for Hadoop Streaming. It provides the facility to automate job parallelization, local execution of unmodified MapReduce jobs etc.

pydoop [24], it is used for Hadoop applications to write MapReduce applications and interact with HDFS.

Pyspark [25] allows us to write Spark applications using python as well as provides a PySpark shell for analyzing the data in a distributed environment.

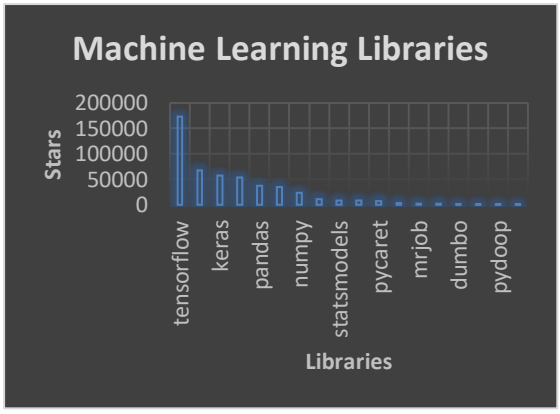

Fig. 2. Machine learning libraries (Stars achieved data from GitHub).

Table II: Information of machine learning libraries from GitHub.

Library	Stars	Forked	Commits
tensorflow [15]	173238	88109	146348
opencv [16]	67968	54872	32778
keras [17]	57875	19314	8024
scikit-learn [12]	53764	24171	29857
pandas [11]	37908	16058	32113
pyspark [25]	35460	26830	
numpy [9]	23181	7979	31987
scipy [19]	11058	4672	29853
statsmodels [13]	8370	2683	14950
cython [26]	7837	1376	17571
pycaret [14]	7166	1605	4720
shogun [20]	2939	1052	17588
mrjob [21]	2599	609	8622
h5py [18]	1855	496	4207
dumbo [22]	1041	150	389
hadoopy [23]	243	59	490
pydoop [24]	233	58	2666
mlpy [19]	9	5	42

Note: Data collected till 12 April 2023.

Deep Learning Libraries:

pytorch [27] is a framework for deep learning applications written in Python developed by Facebook. It is commonly used in image processing and language processing applications.

pytables [28] is used to handle a very large amount of data easily and efficiently. The most important feature of PyTable is memory optimization.

caffe [29], it provides the functionalities for deep learning projects where Expression, Speed, Modularity matter.

Theano [30] allows users to define, optimize and evaluate mathematical expressions and provide low-level API for deep learning applications.

mxnet [31] developed by Amazon, allows us to mix the flavors of two different deep learning applications to maximize efficiency and productivity.

Microsoft cntk [32] helps the library users to develop a distributed deep learning application.

tflearn [33], it is a modular and transparent deep learning library which is easy to use and provides fast prototyping and complete transparency over Tenserflow.

nolearn [34] provides a high-level interface for building and training deep neural networks. It includes inbuilt CNN and RNN for data processing.

elephas [35] is an extension of keras to run distributed deep learning applications.

Lasagne [36] allows us to build and train neural networks in Theano and also support CNN and LSTM.

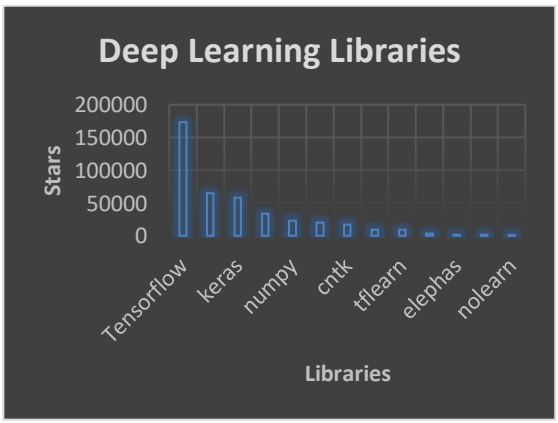

Fig. 3. Deep learning libraries (Stars achieved data from GitHub).

Table III: Information of deep learning libraries from GitHub.

Library	Stars	Forked	Commits
tensorflow [15]	173238	88109	146348
pytorch [27]	65153	17980	58758
keras [17]	57875	19314	8024
caffe [29]	33241	18978	4156
numpy [9]	23181	7980	31987
mxnet [31]	20358	6871	11896
cntk [32]	17343	4382	16117
theano [30]	9711	2508	28132
tflearn [33]	9590	2412	613
lasagne [36]	3842	974	1161
elephas [35]	1564	312	509
pytables [28]	1199	246	4410
nolearn [34]	950	269	396

Note: Data collected till 12 April 2023.

C. Data Visualization

How to represent the analysis output or represent the analyzed information, plays an important role for any Data Scientist. We may represent the information or processed and analyzed data in the form of tables, charts or graphs. To visualize the data in different forms python provides us with a number of libraries or tools such as seaborn, matplotlib, plotly, bokeh, plotnine, altair, pygal, geoplotlib, folium, and Gleam etc, in Table IV and presented in Fig. 4.

matplotlib [37] is the library which is mostly used for static, animated and interactive data visualization in high quality.

seaborn [38] is a statistical data visualization library based on matplotlib. It performs semantic mapping and statistical aggregation internally to produce an effective graph.

plotly [39], it is an open source browser based data visualization library for python which provides approx 30 chart types.

bokeh [40] is an open source library to visualize the productive data in the form of charts, dashboards which is flexible, shareable and interactive on modern web browsers.

plotnine [41] is the python implementation of the Grammar of Graphics inspired by the ggplot2 library of R. It is used to draw the complex plot from the dataframe.

altair [42], it is a python API based on traitlets for data visualization. Here we require the data which is in data frames (row-column) and the data is mapped by the group-by data transformation.

pygal [43], python module to create interactive graphs and charts in the form of SVG format, png format etc.

geoplotlib [44], it is used to visualize the geographical data. It launches a window to show the openstreetmap and plots the point and allows us to zoom, map projection and download the map tiles with openGL.

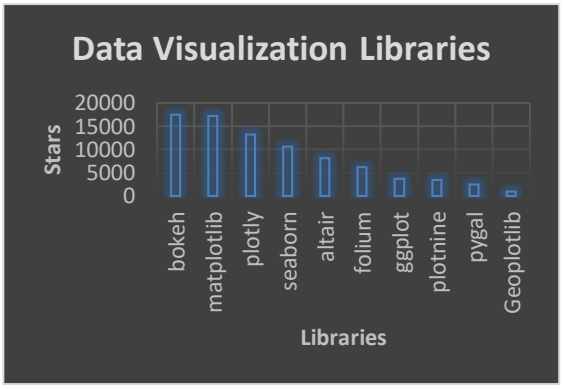

Fig. 4. Data visualization libraries (Stars achieved data from GitHub).

Table IV: Information of data visualization libraries from GitHub.

Library	Stars	Forked	Commits
bokeh [40]	17461	4093	20078
matplotlib [37]	17195	6832	46919
plotly [39]	13246	2337	5849
seaborn [38]	10582	1719	3130
altair [42]	8181	724	3460
folium [45]	6174	2188	1663
ggplot [46]	3645	585	256
plotnine [41]	3423	196	1929
pygal [43]	2532	407	1018
geoplotlib [44]	985	174	159

Note: Data collected till 14 February 2023.

Table V compares various data visualization libraries which are available in Python. This comparison is based on the plot/chart types available in these libraries. Kindly note, if in the table below any entry is marked as N, it doesn't mean that this type of chart or plot is not created through this library. If we want, we can access the functionality of that plot type by writing a few lines of code.

Table V: Data visualization libraries comparison. [37, 38, 39, 40, 41, 42, 43, 46]

Supported Plot	matplotlib	seaborn	plotly	bokeh	ggplot	plotnine	altair	pygal
Bar Graph	Y	Y	Y	Y	Y	Y	Y	Y
Line Graph	Y	Y	Y	Y	Y	Y	Y	Y
Dual Axis Chart	Y	N	Y	N	N	N	Y	N
Area Chart	Y	N	Y	Y	N	Y	Y	N
Stacked Bar Graph	Y	Y	Y	Y	N	N	Y	Y
Mekko / MaosaicChart	Y	Y	N	N	N	N	N	N
Pie Chart	Y	N	Y	Y	N	Y	Y	Y
Scatter Plot Chart	Y	Y	Y	Y	Y	Y	Y	Y
Bubble Chart	N	N	Y	N	N	N	Y	N
Waterfall Chart	N	N	Y	N	N	N	Y	N
Funnel Chart	Y	N	Y	N	N	N	Y	Y
Bullet Chart	Y	N	Y	N	N	N	Y	N
Heat Map	Y	Y	Y	Y	N	N	Y	N
Pair Plot	N	Y	N	N	N	N	N	N
Regression Plot	N	Y	Y	N	Y	N	Y	N
3D	Y	Y	Y	N	N	N	Y	N
Joint Plot	N	Y	N	N	N	N	N	N
Tree Map	Y	N	Y	N	N	N	N	Y
Venn Diagram	Y	N	N	N	N	N	N	N
Radar/ Spider/ Polar Chart	Y	N	Y	N	N	N	N	Y
Lollipop plot	Y	N	N	Y	N	N	N	N
Histogram	Y	Y	Y	Y	Y	Y	Y	Y
Violin	Y	Y	Y	N	N	N	Y	N
Box Plot	Y	Y	Y	Y	N	Y	Y	Y
Hexbin plot	Y	N	Y	N	N	N	N	N
Swarm plot	N	Y	N	N	N	N	N	N
Count plot	N	Y	N	N	N	N	N	N
Strip plot	N	Y	Y	N	N	N	N	N
Boxen Plot	N	Y	N	N	N	N	N	N
Point plot	N	Y	N	N	N	N	N	Y
Dist/ distribution plot	N	Y	Y	N	N	N	N	N

Note: Y means direct function or method is available. N means direct function or method is not available.

III. CONCLUSION

From above mentioned Tables I, II, III, IV and V, we conclude that data collection of Scrapy and Pattern are the most suitable libraries available in python but sometimes it depends on the requirement of the user or the platform from which we have to collect the data. But according to Github data we have now proposed that the textract and Scrapy are the most useful libraries. When we come to data processing or analysis phase pandas, numpy and scipy are the most demanding libraries having strong communities. But when we have to work on machine learning tensorflow is the most suitable. Whereas scikit-learn is useful for predictive analysis. For deep learning again tensorflow is most popular but keras, pytorch and cafee are very useful libraries. When we come to data visualization phase the one and only one library comes to mind which is matplotlib. We can represent our data in most of the plot types directly due to their methods availability in this library. Some plotting methods/functions which are not present in this, their functionality can be implemented by writing a few lines of code. We have proposed that bokeh and plotly also are popular among data science community for data representation. We suggest geoplotlib [44] and folium [45] for representing maps or geographical data.

ACKNOWLEDGEMENT

Funding

This review paper is not funded by any organization directly or indirectly.

Author Contribution

Ankush Joshi and Haripriya Tiwari conceived and designed the study. Ankush Joshi conducted the literature search and data analysis. Both the authors wrote and reviewed the manuscript.

Conflict of Interest

The authors declare no conflicts of interest related to this work.

Data Availability

The data sources used in this review paper are publicly available and can be accessed through the following repositories: GitHub, https://github.com/.

REFERENCES

[1] Anaconda, 2021 State of Data Science, https://www.anaconda.com/state-of-data-science-2021, [Accessed on 14 February 2023].

[2] BeautifulSoup official documentation, https://www.crummy.com/software/BeautifulSoup/bs4/doc/, [Accessed on 14 February 2023].
[3] Textract official documentation, https://textract.readthedocs.io/en/stable/, [Accessed on 14 February 2023].
[4] Scrapy official documentation, https://scrapy.org/companies/, [Accessed on 14 February 2023].
[5] T. D. Smedt and W. Daelemans, "Pattern for Python," *J. Machine Learning Res.*, vol. 13, pp. 2031–2035, 2012.
[6] Taspinar/twitterscraper, https://github.com/taspinar/twitterscraper, [Accessed on 14 February 2023].
[7] G. J. J. van den Burg, A. Nazábal, and C. Sutton, "Wrangling Messy CSV Files by Detecting Row and Type Patterns," *Data Min. Knowl.*, Disc 33, pp. 1799–1820, 2019.
[8] Camelot official documentation, https://camelot-py.readthedocs.io/en/master/, [Accessed on 14 February 2023].
[9] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. Fernández del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke and T. E. Oliphant, "Array Programming with NumPy," *Nature*, vol. 585, pp. 357–362, 2020.
[10] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature*, vol. 17(3), pp. 261-272, 2020.
[11] Pandas (Computer Software), https://github.com/pandas-dev/pandas, [Accessed on 14 February 2023].
[12] F. Pedregosa, G. Varoquaux ,A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrotand and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *J. Machine Learning Res.*, vol. 12, pp. 2825-2830, 2011.
[13] S. Seabold and Josef Perktold. "Statsmodels: Econometric and Statistical Modeling with Python." *in Proc. 9th Python in Sc. Conf.*, Texas, pp. 92-96, 2010.
[14] Pycaret: An Open Source, Low Code Machine Learning Library in Python, Moez Ali, https://pycaret.org/, [Accessed on 14 February 2023].
[15] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu and X. Zheng, "TensorFlow, Large-scale Machine Learning on Heterogeneous Systems," *Comp. Sci.*, 2015. arXiv:1603.04467.
[16] OpenCV Introduction, https://docs.opencv.org/4.x/d1/dfb/intro.html, [Accessed on 14 February 2023].
[17] Official Documentation, https://keras.io/, [Accessed on 14 February 2023].
[18] Andrew Collette, Official Documentation h5py, https://pypi.org/project/h5py/, [Accessed on 14 February 2023].
[19] D. Albanese, R. Visintainer, S. Merler, S. Riccadonna, G. Jurman and C. Furlanello. "mlpy: Machine Learning Python," *Comp. Sci.*, 2012. arXiv:1202.6548.
[20] S. Sonnenburg, H. Strathmann, S. Lisitsyn, V. Gal, F. J. I. García, W. Lin, S. De, C. Zhang, frx, tklein23, E. Andreev, J. Behr, sploving, P. Mazumdar, C. Widmer, P. D. Zora, G. D. Toni, S. Mahindre, A. Kislay, K. Hughes, R. Votyakov, khalednasr, S. Sharma, A. Novik, A. Panda, E. Anagnostopoulos, L. Pang, A. Binder, serialhex and B. Esser, "shogun-toolbox/shogun: Shogun 6.1.0," Zenodo, 2017. https://doi.org/10.5281/zenodo.1067840.
[21] Development Team, Official Documentation mrjob, https://mrjob.readthedocs.io/en/latest/index.html, [Accessed on 14 February 2023].
[22] Dumbo, https://klbostee.github.io/dumbo/, [Accessed on 14 February 2023].
[23] B. A. White, Hadoopy: Python Wrapper for Hadoop using Cython, https://hadoopy.readthedocs.io/en/latest/, [Accessed on 14 February 2023].
[24] S. Leo and G. Zanetti, "Pydoop: A Python MapReduce and HDFS API for Hadoop," *in Proc 19th ACM Int. Symp. High Perform. Distrib. Comput.*, Illinois, pp. 819-825, 2010.
[25] PySpark Documentation, https://spark.apache.org/docs/latest/api/python/#:~:text=PySpark%20is%20an%20interface%20for,data%20in%20a%20distributed%20environment, [Accessed on 14 February 2023].
[26] Cython, https://github.com/cython/cython, [Accessed on 14 february 2023].
[27] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," *in 33rd Conf. Neural Inform. Process. Sys.*, Canada, pp. 1-12, 2019.
[28] PyTables Hierarchical Datasets in Python, https://www.pytables.org, [Accessed on 14 February 2023].
[29] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama and T. Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding", *Comp. Sci.*, 2014. arXiv:1408.5093.
[30] Theano 1.0.5, https://pypi.org/project/Theano/, [Accessed on 10 April 2023].
[31] Mxnet Features, https://mxnet.apache.org/versions/1.9.1/features, [Accessed on 10 April 2023].
[32] The Microsoft Cognitive Toolkit, https://learn.microsoft.com/en-in/cognitive-toolkit/, [Accessed on 10 April 2023].
[33] TFLearn: Deep Learning Library Featuring A Higher-level API for TensorFlow, http://tflearn.org/, [Accessed on 10 April 2023]
[34] Daniel Nouri, Nolearn: Sckit-learn Compatible Neural Network Library, 2014. https://github.com/dnouri/nolearn.
[35] Danielenricocahall, "Elephas: Distributed Deep Learning with Keras & Spark," https://github.com/danielenricocahall/elephas#elephas-distributed-deep-learning-with-keras--spark.
[36] S. Dieleman, J. Schlüter; C. Raffel, E. Olson, S. K. Sønderby, D. Nouri, D. Maturana, M. Thoma, Eric Battenberg Baidu Research, J. Kelly, J. De Fauw, M. Heilman, diogo149, B. McFee, H. Weideman, takacsg84, peterderivaz, Jon, instagibbs, Dr. K. Rasul, CongLiu, Britefury and J. Degrave, Lasagne: First Release, 2015, http://dx.doi.org/10.5281/zenodo.27878
[37] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," *Comput. Sci. & Eng.*, vol. 9, no. 3, pp. 90-95, 2007.
[38] M. L. Waskom, "Seaborn: Statistical Data Visualization," *J. Open Source Soft.*, vol. 6(60), 3021, pp. 1-4, 2021.
[39] Plotly Open Source Graphing Library for Python, https://plotly.com/python/, [Accessed on 14 February 2023].
[40] Bokeh Documentation, https://docs.bokeh.org/en/latest/, [Accessed on 14 February 2023].
[41] A Grammar of Graphics for Python, https://plotnine.readthedocs.io/en/stable/, [Accessed on 14 February 2023].
[42] J. VanderPlas, B. E. Granger, J. Heer, D. Moritz, K. Wongsuphasawat, A. Satyanarayan, E. Lees, I. Timofeev, B. Welsh and S. Sievert, "Altair: Interactive Statistical Visualizations for Python," *J. Open Source Soft.*, vol. 3(32), 1057, pp. 1-2, 2018.
[43] Pygal: Beautiful Python Charting, https://www.pygal.org/en/stable/, [Accessed on 14 February 2023].
[44] Geoplotlib, https://github.com/andrea-cuttone/geoplotlib#readme, [Accessed on 14 February 2023].
[45] Folium, https://github.com/python-visualization/folium, [Accessed on 14 February 2023].
[46] Ggplot, https://github.com/yhat/ggpy, [Accessed on 14 February 2023].