Webology (ISSN: 1735-188X)
Volume 16, Number 1, 2019

Effectiveness Of Python Libraries In Machine Learning: A Review

Harbhajan Singh[1], Vijay Dhir[2]

[1]Research Scholar, PG Dept. of Computer Science, Khalsa College, Amritsar, 143002

[2]Director, R&D, Sant Baba Bhag Singh University, Jalandhar, 144030

Abstract
Python is a scriptable and interpreted language that is used for developing solutions to real-world issues as well as to grasp conceptual ideas for learners. It was created by Guido van Rossum. The key characteristics of Python includes simple to learn, freeware, open source and a high level programming language. It is also portable, loosely typed, procedural as well as object-oriented and embedded. Additionally, Python has a sizable collection of comprehensive standard libraries that are expandable. In this paper, we will explore the usefulness and performance of python libraries especially used for machine learning. This paper begins with the brief introduction of machine learning and Python programming language. In the next part, review of those python libraries which are particularly designed to perform machine learning tasks, are performed thoroughly followed by the conclusion of this paper.

Keywords: Python, machine learning, big data, Python libraries.

Introduction

Machine Learning: - Artificial intelligence includes machine learning, which is used to educate computers how to learn from their past experiences and make judgements when necessary. Making computer programmes in advance without human interaction is the basic goal of ML[1]. Learning any machine requires familiarity with patterns, predictions, input, and prior experience [13]. All of these are utilised to create a machine that can decide by itself (a person is not involved in the decision-making process) and provide the desired results. A machine learning model takes raw data as input, interprets it, and then makes predictions about the output based on that understanding.

Some of machine learning based technologies:

- Search engines learn to deliver us the best results.
- Anti-spam software learns to filter our email communications.
- Software that learns to detect frauds secures credit card transactions.
- Digital cameras pick up on facial recognition, while smartphone personal assistant apps pick up on voice instructions.

Python Programming Language: - The use of Python in the area of data science has reached unprecedented levels, especially in the area of freely available tools and libraries. The data science community has chosen Python as their preferred programming language, with R coming in second[3]. The Python's success is likely due to its enormous ecosystem, which has a variety of libraries for every area of data science and machine learning, as well as its relative simplicity of use (even for non-computer scientists).Python is an OOP language with highly sophisticated capabilities that may be used to create and implement different machine learning methods. Each phase of the machine learning paradigm has a specialised library package available in Python[5].

Review of Python Libraries

1. Pandas: - Pandas is a Python library used for working with data sets. It has functions for analysing, cleaning, exploring, and manipulating data. It was created in 2008 by Wes McKinney and is used for data analysis in Python. Following are the key features of Pandas library:-

- Pandas is an open-source library in Python that is made mainly for working with relational or labelled data. It provides various data structures and operations for manipulating numerical data and time series. The data produced by Pandas are often used as input for plotting functions of Matplotlib, statistical analysis in SciPy, and machine learning algorithms.
- Easy handling of missing data (represented as NaN) in floating point as well as non-floating point data.
- Columns can be inserted and deleted from DataFrame and higher dimensional objects.
- Powerful group by functionality for performing split-apply-combine operations on data sets[6].
- Data Visualization.

The two primary data structures of pandas, Series (1-dimensional) and Data Frame (2-dimensional), handle the majority of typical use cases in finance, statistics, social science, and many areas of engineering. The term "Pandas" refers to an open-source library for manipulating high-performance data in Python[7].

Installing Pandas

Use the following command to install the Pandas:

pip install pandas

Importing Pandas

After the pandas have been installed into the system, you need to import the library. This module is generally imported as follows:

import pandas as pd

Here, pd is referred to as an alias to the Pandas. However, it is not necessary to import the library using the alias, it just helps in writing less amount code every time a method or property is called.

Example:- Reading CSV file

```
import pandas as pd
df = pd.read_csv('c:\\bank.csv')

df.head()
print(df.to_string())
```

```
    age;"job";"marital";"education";"default";"balance";"housing";"loan";"contact";"day";"month";"duration";"campaign";"pdays";"previous";"poutcom
e";"y"
0                                   30;"unemployed";"married";"primary";"no";1787;"no";"no";"cellular";19;"oct";79;1;-1;0;"unknown";"n
o"
1                                33;"services";"married";"secondary";"no";4789;"yes";"yes";"cellular";11;"may";220;1;339;4;"failure";"n
```

2. NumPy: - NumPy **(**Numerical Python**)** is an open source Python library that is used in almost every field of Science and Engineering. The NumPy [2] API is used extensively in Pandas, SciPy, Matplotlib, Scikit-learn, Scikit-image and most other data science and scientific python packages. The NumPy library contains multidimensional arrays and matrix data structures. It provides **ndarray**, a homogeneous n-dimensional array object, with methods to efficiently operate on it. NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices. It supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.

NumPy has the following features:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

NumPy stands for numeric python which is a python package for the computation and processing of the multidimensional and single dimensional array elements. Travis Oliphant created NumPy package in 2005. It is an extension module of Python which is mostly written in C. It provides various functions which are capable of performing the numeric computations with a high speed. NumPy provides various powerful data structures, implementing multi-dimensional arrays and matrices.

Installing NumPy

To install NumPy, we strongly recommend using a scientific Python distribution. If you already have Python, you can install NumPy with the following command:

conda install numpy

or

pip install numpy

Example: - Reshaping 1D array into 2D using Numpy

```
import numpy as np

array1 = np.array([1, 3, 5, 7, 2, 4, 6, 8])

# reshape a 1D array into a 2D array
# with 2 rows and 4 columns
result = np.reshape(array1, (2, 4))
print(result)
```

```
[[1 3 5 7]
 [2 4 6 8]]
```

3. **Scikit-Learn:** - Scikit-Learn, also known as sklearn is a python library to implement machine learning models and statistical modelling. Through scikit-learn, we can implement various machine learning models for regression, classification, clustering, and statistical tools for analyzing these models [4]. It also provides functionality for dimensionality reduction, feature selection, feature extraction, ensemble techniques, and inbuilt datasets.
This library is built upon NumPy, SciPy, and Matplotlib. Scikit-learn comes with several inbuilt datasets. These datasets are easy to understand and we can directly implement ML models on them. These datasets are good for beginners.

Scikit-learn is an open-source python library that is used for pre-processing, cross-validation, and visualization algorithms.

Important features of Scikit-learn:

- Simple and efficient tools for data mining and data analysis. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means, etc.
- Accessible to everybody and reusable in various contexts.
- Built on the top of NumPy, SciPy, and matplotlib.
- Open source, commercially usable – BSD license.

Installation

If you have already installed NumPy and Scipy, then use pip command to install it as:

pip install -U scikit-learn

Example: - Loading Datasets

```
In [1]: import pandas as pd
        from sklearn import datasets

        df = datasets.load_boston()
        type(df)

Out[1]: sklearn.utils.Bunch

In [2]: boston = pd.DataFrame(data=df.data, columns=df.feature_names)
        boston['target'] = df.target

        boston.head()
```

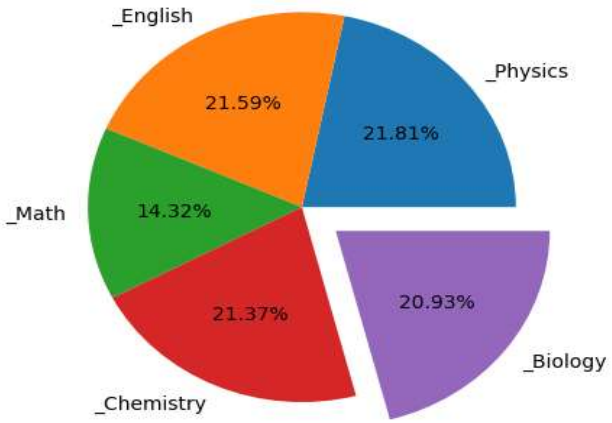
Out[2]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	target
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33	36.2

4. MatplotLib: - Matplotlib is a data visualization and graphical plotting library used for Python. A Python matplotlib is a structured script in which few lines of code are required in most instances to generate a visual data plot. Matplotlib is an amazing visualization library in Python for 2D plots of arrays. It is a multi-platform data visualization library built on NumPy arrays[8]. It was introduced by John Hunter in the year 2002. It has a module named pyplot which makes things easy for plotting by providing feature to control line styles, font properties, formatting axes etc. It supports a very wide variety of graphs and plots namely - histogram, bar charts, power spectra, error charts etc. It is used along with NumPy to provide an environment that is an effective open source alternative for MatLab. It can also be used with graphics toolkits like PyQt and wxPython[9]. It is an exceptionally fast at a variety of operations. In addition, it can export visualizations to all popular image formats, including PDF, SVG, JPG, PNG, BMP, and GIF.

Example

```
import matplotlib.pyplot as plt
_Marks = [99,98,65,97,95]
_Subjects = ["_Physics","_English","_Math","_Chemistry","_Biology"]
plt.axis("equal")
plt.pie(_Marks,labels=_Subjects,explode=[0,0,0,0,0.2],autopct="%1.2f%%")
plt.show()
```



5. Seaborn :- Python Seaborn library is a popular data visualization library that is commonly used for data science and machine learning tasks. It is built on the top of matplotlib data visualization library and can perform exploratory analysis. Its plotting functions operate on data frames and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots. Its dataset-oriented, declarative API lets you focus on what the different elements of your plots mean, rather than on the details of how to draw them[10].

Plots are basically used for visualizing the relationship between variables. Those variables can be either completely numerical or a category like a group, class, or division. Seaborn divides the plot into the below categories – Relational plots: This plot is used to understand the relation between two variables[15].

Categorical plots: This plot deals with categorical variables and how they can be visualized.

Distribution plots: This plot is used for examining univariate and bivariate distributions

Regression plots: The regression plots in Seaborn are primarily intended to add a visual guide that helps to emphasize patterns in a dataset during exploratory data analyses.

Matrix plots: A matrix plot is an array of scatterplots.

Multi-plot grids: It is a useful approach to draw multiple instances of the same plot on different subsets of the dataset.

Installation of seaborn library for Python: It can be installed with the following command:

pip install seaborn

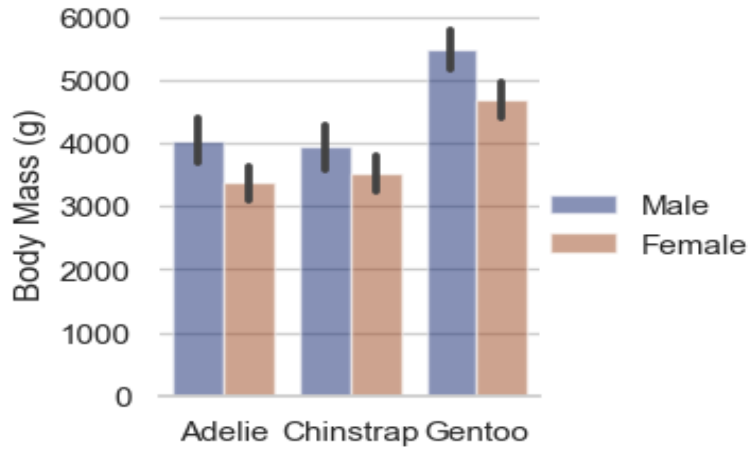
Example: Grouped Barplots

```
import seaborn as sns
sns.set_theme(style="whitegrid")

penguins = sns.load_dataset("penguins")

# Draw a nested barplot by species and sex
g = sns.catplot(
    data=penguins, kind="bar",
    x="species", y="body_mass_g", hue="sex",
    errorbar="sd", palette="dark", alpha=.5, height=3
)
g.despine(left=True)
g.set_axis_labels("", "Body Mass (g)")
g.legend.set_title("")
```

```
C:\Users\lenovo\AppData\Local\Programs\Python\Python311\Lib
  self._figure.tight_layout(*args, **kwargs)
```



6. TensorFlow:- TensorFlow is an open-source library of data flow graphs computations, which are sharpened for Machine Learning. It was designed to meet the high-demand requirements of Google environment for training Neural Networks [11]. It is a successor of DistBelief, a Machine Learning system, based on Neural Networks. TensorFlow is not only for scientific use in borders in Google but also used in a variety of real-world applications. The key feature of TensorFlow is its multi-layered nodes system that enables quick training of artificial neural networks on large datasets. This provides power for voice recognition and object identification from pictures in Google.

7. Plotly:- It is a web-based toolbox for building visualizations, exposing APIs to some programming languages in which python is one of them. There is a number of robust, out-of-box graphics on the plotly.com website. In order to use Plotly, you will need to set up your API key. The graphics will be processed at server side and will be posted on the internet.

8. Keras:- It is an open-source library for building Neural Networks at a high-level of the interface, and it is written in Python. It uses Theano or TensorFlow as its backends, but microsoft makes its efforts now to integrate CNTK (Microsoft's Cognitive Toolkit) as a new back-end.The minimalistic approach in design aimed at fast and easy experimentation through the building of compact systems. Keras is really eased to get started with and keep going with quick prototyping. It is written in pure Python. It is highly modular and extendable. The general idea of Keras is based on layers, and everything is built around these layers. Data is prepared in tensors which is the first layer and responsible for input of tensors. On the other hand last layer is responsible for output. The whole model is built in between these two layers.

9. NLTK:- The name of this suite of libraries stands for Natural Language Toolkit and, as the name implies, it is used for common tasks of symbolic and statistical Natural Language Processing. NLTK was intended to facilitate teaching and research of NLP and the related fields like Linguistics, Cognitive Science, Artificial Intelligence, etc.. The functionality of NLTK allows a lot of operations such as text tagging, classification, and tokenizing, name entities identification, building corpus tree that reveals inter and intra-sentence dependencies, stemming and semantic reasoning [12]. All of the building blocks allow for building complex research systems for different tasks such as sentiment analytics, automatic summarization etc.

10. Statsmodels:- Statsmodels is a python module that enables its users to conduct data exploration via the use of various methods of estimation of statistical models. It also performs statistical assertions and analysis. This module helps with analysing data and creating statistical models. The Python Statsmodels Library is one of the many computational pillars of Python geared for statistics, data processing and data science. It is built on SciPy, Matplotlib, and NumPy, but it includes more sophisticated statistical testing and modeling functions not found in SciPy or NumPy. The library also provides extensive plotting functions that are designed specifically for the use in statistical analysis and tweaked for good performance with big data sets of statistical data.

11. Theano- Theano is a Python package that defines multidimensional arrays similar to NumPy, along with math operations and expressions. The library is compiled, making it run efficiently on all architectures. Originally developed by the Machine Learning group of Université de Montréal, it is primarily used for the needs of Machine Learning. The important thing to note is that Theano tightly integrates with NumPy on low-level of its operations. The library also optimizes the use of GPU and CPU, making the performance of data-intensive computation even faster.

12. Scrapy; - Scrapy is a library for making crawling programs, also known as spider bots, for retrieval of the structured data, such as contact info or URLs, from the web. It is open-source and written in Python. It was originally designed particularly for scraping, as its name indicate, but it has evolved in the full-fledged framework with the ability to gather data from APIs and act as general-purpose crawlers. The architecture of Scrapy is built around Spider class, which encapsulates the set of instruction that is followed by the crawler.

13. Bokeh-Another great visualization library is Bokeh, which is aimed at interactive visualizations. The main focus of Bokeh is interactivity and it makes its presentation via modern browsers in the style of Data-Driven Documents.

14. Streamlit:- Streamlit help you to turn data scripts into shareable web apps in very less time. It is all Python, open-source, and free. After creating an app, you can use community cloud platform to deploy, manage, and share your app. Streamlit is the easiest way especially for people with no front-end knowledge to put their code into a web application.

No front-end (html, js, css) experience or knowledge is required for users. It is not needed to spend days or months to create a web app, anyone can create a really beautiful machine learning or data science app in only a few hours or even minutes. It is compatible with the majority of Python libraries (e.g. pandas, matplotlib, seaborn, plotly, Keras, PyTorch, SymPy(latex)). It is an open source python based framework for developing and deploying interactive data science dashboards and machine learning models. Streamlit was founded in 2018 by ex-Google engineers. It is built on top of Python and supports many of the mainstream Python libraries such as matplotlib, plotly and pandas.

15. SciPy:- SciPy (pronounced "Sigh Pie") is an open-source software for mathematics, science, and engineering. It includes modules for statistics, optimization, integration, linear algebra, Fourier transforms, signal and image processing, ODE solvers, and many more. SciPy is built to work with NumPy arrays, and provides many user-friendly and efficient numerical routines, such as routines for numerical integration and optimization. They run on all popular operating systems easily free of charge. The SciPy library supports integration, gradient optimization, special functions, ordinary differential equation solvers, parallel programming tool [14]. SciPy implementation exists in almost every complex numerical computation. The scipy is a data-processing and system-prototyping environment as similar to MATLAB. It is easy to use and provides great flexibility to scientists and engineers.

Conclusion

This research paper highlights the effectiveness of using Python libraries for machine learning jobs. Machine learning is the prime research area of the present time. Machine learning makes the systems able to make decisions based on their training. Python has very useful packages for implementing various machine learning techniques and algorithms. In the paper, we have outlined these Python packages with a brief introduction and their significant use for machine learning. It is concluded that Python is a powerful programming language that provides various packages used for machine learning under one roof.

References

[1] Thilaga, P. J., Khan, B. A., Jones, A. A., & Kumar, N. K. (2018, April). Modern Face Recognition with Deep Learning. In 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT) (pp. 1947-1951). IEEE.

[2] T. E. Oliphant, A guide to NumPy, USA: Trelgol Publishing, 2006.

[3] E. Jones, T. E. Oliphant, P. Peterson, et al. SciPy: Open Source Scientific Tools for Python, 2001.

[4] S. Browne, J. Dongarra, E. Grosse and T. Rowan, The Netlib Mathematical Software Repository, D-Lib Magazine, 1995.

[5] KDnuggets, (2019, February 4th), https://www.kdnuggets.com/.

[6] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. S. Seljebotn and K. Smith, Cython: The Best of Both Worlds, Computing in Science and Engineering, 13, 31-39, 2011.

[7] GitHub, (2019, February 14th), https://github.com/. [8] W. Mckinney, pandas: a Foundational Python Library for Data Analysis and Statistics. Python High Performance Science Computer, 2011.

[8] J. D. Hunter, Matplotlib: A 2D graphics environment, Computing In Science & Engineering, 9(3), 90-95, 2007.

[9] Bokeh Development Team, Bokeh: Python library for interactive visualization, 2018.

[10] ggplot, (2019, February 4th), http://ggplot.yhathq.com/.

[11] Dash, (2019, February 4th), https://plot.ly/products/dash/ [18] F. Pedregosa et al., Scikit-learn: Machine Learning in {P}ython, Journal of Machine Learning Research, 12, 2825-2830, 2011.

[12] S. Raschka, MLxtend: Providing machine learning and data science utilities and extensions to Python's scientific computing stack, The Journal of Open Source Software, 3(24), 2018.

[13] S. Sonnenburg, et al., The SHOGUN Machine Learning Toolbox, Journal of Machine Learning Research, 11, 1799-1802, 2010.

[14] D. Albanese, R. Visintainer, S. Merler, S. Riccadonna, G. Jurman and C. Furlanello, mlpy: Machine Learning Python, 2012.

[15] Y. Jia, et al. Caffe: Convolutional Architecture for Fast Feature Embedding, MM 2014 - Proceedings of the 2014 ACM Conference on Multimedia, 2014.