

POPULAR PYTHON LIBRARIES AND THEIR APPLICATION DOMAINS

A.L.Sayeth Saabith¹, T.Vinothraj², MMM.Fareez³

¹Centre for Information Communication Technology
Faculty of Science, Eastern University, Vantharoomulai, Sri Lanka

²Centre for Information Communication Technology
Faculty of Science, Eastern University, Vantharoomulai, Sri Lanka

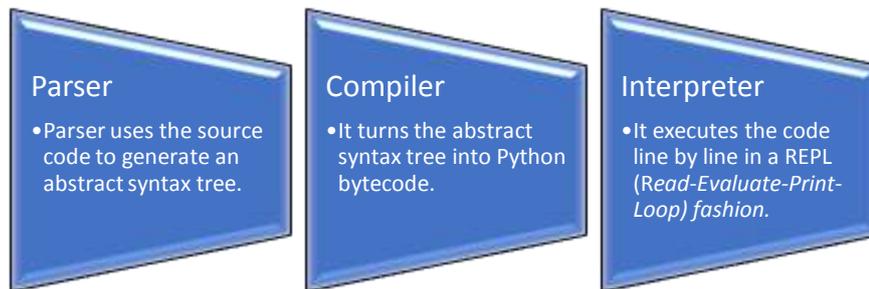
³Finance Department
Eastern University, Vantharoomulai, Sri Lanka

Abstract —Python is a general purpose, dynamic, high-level, and interpreted programming language. It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures. Python is supporting the development of a wide range of applications such as Build a website, develop a Game, Perform Computer Vision (Facilities like face-detection and color-detection), Implement Machine Learning (Give a computer the ability to learn), Enable Robotics, Perform Web Scraping (Harvest data from websites), Perform Data Analysis. Automate a web browser, Perform Scripting, Perform Scientific Computing, Build Artificial Intelligence. In this paper we first analyze you to Python programming language popularity and features. Moreover, this paper specifying applications areas where python can be applied and popular libraries which are using in Python applications.

Keywords-Python, Python Library, Artificial Intelligence, Machine Learning, and Data Science

I. INTRODUCTION

Python is a powerful high-level, interpreted, interactive, and object-oriented scripting language created by Guido Van Rossum in late 1980’s. The Python programming language is an object-oriented language, which means that it can model real-world entities. Python codes are compiled line-by-line which makes debugging errors much easier and efficient. Python architecture can be divided into three categories.



Python is considered as one of the most popular and in-demand programming language. A recent Stack Overflow survey showed that Python has taken over languages such as Java, C, C++ and has made its way to the top.

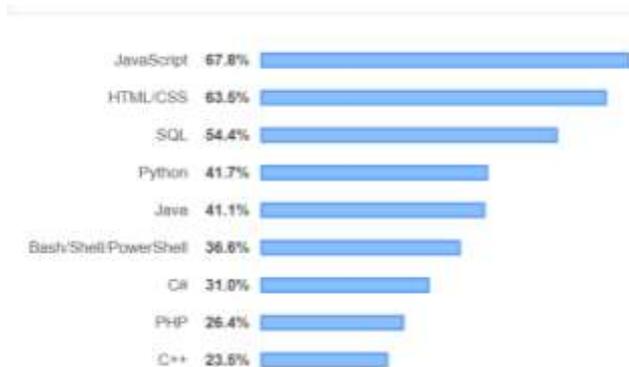


Figure 1: Most Popular Technology - 2019 [10]

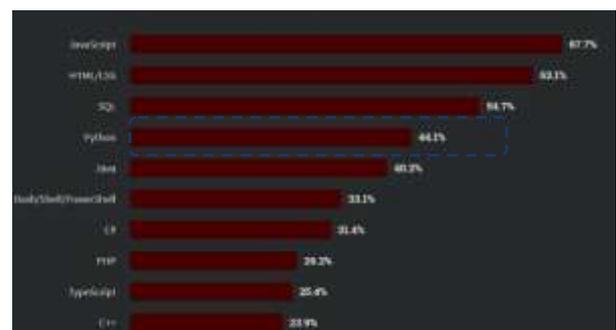


Figure 2: Most Popular Technology - 2020 [11]

| Nov 2020 | Nov 2019 | Change | Programming Language | Ratings | Change |
|----------|----------|--------|----------------------|---------|--------|
| 1 | 2 | ▲ | C | 16.21% | +0.17% |
| 2 | 3 | ▲ | Python | 12.12% | +2.27% |
| 3 | 1 | ▼ | Java | 11.68% | -4.57% |
| 4 | 4 | | C++ | 7.60% | +1.99% |
| 5 | 5 | | C# | 4.67% | +0.36% |
| 6 | 6 | | Visual Basic | 4.01% | -0.22% |
| 7 | 7 | | JavaScript | 2.03% | +0.10% |
| 8 | 8 | | PHP | 1.79% | +0.07% |

Figure 3: TIOBE Comparison Nov 2019 and Nov 2020 [15]

The following features and advantages that are the Python programming language is becoming so popular [2,3].

- **Easy to Learn and Use:** Python is easy to learn as compared to other programming languages. Its syntax is straightforward and much the same as the English language. There is no use of the semicolon or curly-bracket, the indentation defines the code block. It is the recommended programming language for beginners.
- **Expressive Language:** Python can perform complex tasks using a few lines of code. A simple example, the hello world program you simply type `print("Hello World")`. It will take only one line to execute, while Java or C takes multiple lines.
- **Interpreted Language:** Python is an interpreted language; it means the Python program is executed one line at a time. The advantage of being interpreted language, it makes debugging easy and portable.
- **Cross-platform Language:** Python can run equally on different platforms such as Windows, Linux, UNIX, and Macintosh, etc. So, we can say that Python is a portable language. It enables programmers to develop the software for several competing platforms by writing a program only once.
- **Free and Open Source:** Python is freely available for everyone. It is freely available on its official website www.python.org. It has a large community across the world that is dedicatedly working towards make new python modules and functions. Anyone can contribute to the Python community. The open-source means, "Anyone can download its source code without paying any penny."
- **Object-Oriented Language:** Python supports object-oriented language and concepts of classes and objects come into existence. It supports inheritance, polymorphism, and encapsulation, etc. The object-oriented procedure helps to programmer to write reusable code and develop applications in less code.
- **Extensible:** Suppose that other languages such as C/C++ can be used to compile the code and thus it can be used further in our Python code. It converts the program into byte code, and any platform can use that byte code.
- **Large Standard Library:** It provides a vast range of libraries for the various fields such as machine learning, web developer, and also for the scripting. There are various machine learning libraries, such as Tensor flow, Pandas, Numpy, Keras, and Pytorch, etc. Django, flask, pyramids are the popular framework for Python web development.
- **GUI Programming Support:** Graphical User Interface is used for the developing Desktop application. PyQt5, Tkinter, Kivy are the libraries which are used for developing the web application.
- **Integrated:** It can be easily integrated with languages like C, C++, and JAVA, etc. Python runs code line by line like C, C++, Java. It makes easy to debug the code.
- **Embeddable:** The code of the other programming language can use in the Python source code. We can use Python source code in another programming language as well. It can embed other language into our code.
- **Dynamic Memory Allocation:** In Python, we don't need to specify the data-type of the variable. When we assign some value to the variable, it automatically allocates the memory to the variable at run time. Suppose we are assigned integer value 50 to x, then we don't need to write `int x = 50`. Just write `x = 50`.
- **Components Integrity:** Python is able to make an integrated connection between its codes and other components. Tools such as Swing and SIP can import the compiled codes of other languages for using Python.

II. TOP-NOTCH COMPANIES USING PYTHON

Python Programming language is widely used by companies and individuals in wide applications like building web apps, data science and analytics, automating operation via Develops and building scalable and reliable enterprise applications. Here in this article, we have listed down top companies in different industries using Python languages in various applications.

A. Top technologies companies using Python:

Google: Python is one of the core languages used by Google, along with Java and Golang. It is extensively used in multiple applications at Google, and receives priority support. Google uses Python as main language for its data team. Some of the key Python contributors are Googlers and they continue to use, promote, and support the language actively. Google is always sponsoring various Python conferences like PyCon. One of the examples is YouTube which uses Python language for view video, control templates for website, administer video, access to canonical data, and many more.

Facebook/Instagram: Python is the third most favored programming language used in Facebook after Php and C++. It is used to maintain thousands of libraries and FB infrastructure. Python is main programming language of Facebook/Instagram, especially in AI and Data Science. For example: Oculus team at Facebook uses Python and C++ for developing VR hardware and software.

Uber: Python is key programming language in Uber tech stack. Their back end uses Node.js and Python. Their data team uses both Python and R.

Reddit: Reddit is one of the largest sites at scale to use the Python as its main programming languages. Significant portion of Reddit is built using python language.

Dropbox: Dropbox uses Python across their application development, infrastructure and operations. Their platform is built by team of Python core developers.

Mozilla: Python is used in large scale to build system, CI configuration, test harnesses, command line tooling and countless other scripts, tools or GitHub projects at Mozilla.

Netflix: Netflix uses python as main programming language. It use Python through the full content lifecycle, from deciding which content to fund all the way to operating the CDN that serves the final video to 148 million+ members. Netflix is also an active contributor of many python packages.

B. Top Finance companies using python

Python is widely used by finance companies and institutions like large scale banks, hedge funds, regulators etc. Here are some of the list of top finance companies using python language.

Goldman Sachs: Goldman Sachs uses Python in their data team and as main programming language. For example: Goldman Sachs uses Python for natural language processing in their Fixed Income, Currencies & Commodities department. Similar applications/jobs of Python development can be checked on their career pages.

Paypal: Python is used to manage entire tech infrastructure. Their Braintree SDK for Express Checkout integration and other apis uses Python as main programming language. There are good numbers of python code and api kits available on their website as well on GitHub.

JP Morgan Chase: JP Morgan Chase is leading financial institutions dealing in investment banking, financial services and asset management. It uses Python and Java as main programming languages. In data analytics, python is widely used along with Hadoop and Hive. One example of Python is Python development using Enaml in Athen.

C. Top Government agencies using Python

Python is used by leading government agencies and organizations across world.

Consumer Financial Protection Bureau (CFPB): CFPB uses python language for running most of its applications. For example: Its collab project which is a Django application is used for storing and looking up information on employees and contractors.

Nasa: NASA uses Python extensively and open sources much of their software. Python is main language for data and image processing for spatial data at Nasa. Different applications of python can be checked at code.nasa.gov.

CIA: CIA is using Python in its state sponsored hacking tools. They regularly document and publish stats regarding python development at their agency.

III. PYTHON APPLICATION AND THEIR POPULAR LIBRARIES

Python is the most versatile language in the programming world, and it applicable in almost every domain of software development. It helps us in taking care of our current programming task as well as lets us focus on the core functionality of Python programming languages. Python makes its presence in every emerging field. It is the fastest-growing programming language and can develop any application. [1,3, 4, 5].

1. **Natural Language Processing:** Natural language processing (NLP) is about developing applications and services that are able to understand human languages. Some Practical examples of NLP are speech recognition for eg: google voice search, understanding what the content is about or sentiment analysis. Successfully implementations of NLP are Search engines like Google, Yahoo, Social websites feed like the Facebook news feed, Speech engines like Apple Siri, and Spam filters like Google spam filters. It's not just about the usual spam filtering, now spam filters understand what's inside the email content and see if it's a spam or not.



Python Library for NLP

- a) **NLTK:** Natural language toolkit (NLTK) is the most popular library for natural language processing (NLP) which was written in Python and has a big community behind it. NLTK also is very easy to learn, actually, it's the easiest natural language processing (NLP) library that you'll use.
 - b) **Spacy:** Open-source library for industrial-strength Natural Language Processing in Python. Built using Python and Cython, spaCy combines the best of both languages, the convenience from Python and the speed from Cython to deliver one of the best-in-class NLP experiences. Key takeaways from spaCy are Part-of-Speech Tagging, Tokenization, Dependency Parsing, Sentence Segmentation, Entity & Sentence Recognition, Seamless integration with Deep Learning, Methods for cleaning and normalizing text.
 - c) **CoreNLP:** Stanford CoreNLP is a suite of tools built for implementing a Natural Language Processing into your project. Originally written in Java, CoreNLP works with other languages such as Python, JavaScript and many more. It offers several linguistic features such as indicating sentiment, extracting relations between entities, mark up sentence structures, provide the base form of words, translate to Lemmatization, Part-of-Speech and Morphological Tagging, Named Entity Recognition, Tokenization and Sentence Split.
 - d) **Gensim:** Gensim is a library for Topic Modelling, Similarity Retrieval and Natural Language Processing written in Python. Gensim offers memory-independent implementation capabilities for several popular algorithms such as Latent Dirichlet Allocation (LDA), Random Projections (RP), Latent Semantic Analysis (LSA/LSI/LVD) and Hierarchical Dirichlet Process (HDP).
 - e) **PyNLPI:** Pronounced as "Pineapple", it is an open-source Natural Language Processing library for Python. PyNLPI is ideal for a variety of tasks ranging from building simplistic models and extraction of n-grams and frequency lists, with support for complex data types and algorithms.
 - f) **Pattern:** Pattern is primarily a web mining module for Python with the inclusion of tools for a host of purposes such as Data Mining, Natural Language Processing, Machine Learning and Network Analysis. Focusing on the NLP aspect of Pattern, it is well equipped with the essential features of any NLP such as Part-of-Speech Taggers, n-grams, Sentiment Analysis, WordNet, Text Classification and Tokenization.
 - g) **TextBlob:** Textblob is another open-source Python library for processing text-based data and offers smooth integration with other programming languages via its APIs. Textblob can be your savior if you're in a hurry to perform standard NLP operations. It provides several of the features standard to any Natural Language Processing library such as: Part-of-Speech Tagging, Sentiment Analysis, Classification, Tokenization, n-grams, Word Inflection, WordNet Integration, Language translation and detection powered by Google Translate, Word and phrase frequencies, Parsing, Spelling correction, Add new models or languages through extensions.
2. **Computer Vision:** Computer vision is the construction of explicit, meaningful descriptions of physical objects from their image. The output of computer vision is a description or an interpretation of structures in 3D scene.

Python Library for Computer Vision

- a) **OpenCV:** OpenCV is a popular and open-source computer vision library that is focussed on real-time applications. The library has a modular structure and includes several hundreds of computer vision algorithms. OpenCV includes a number of modules including image processing, video analysis, 2D feature framework, object detection, camera calibration, 3D reconstruction.
- b) **Fastai:** It is a deep learning library that provides high-level components which can quickly and easily provide state-of-the-art results in standard deep learning domains. It also provides researchers with low-level components that can

be mixed and matched to build new approaches. fastai includes various features, such as a GPU-optimized computer vision library which can be extended in pure Python, a new type dispatch system for Python along with a semantic type hierarchy for tensors.

- c) **IPSDK**: It is an image processing library in C++ and Python. The library offers a complete range of image processing features to process datasets as well as a comprehensive and optimized range of functionalities for 2D and 3D image processing. IPSDK automatically adjusts itself to the architecture and capabilities of the processor. The features of this library include full PC cluster support, high performance and high availability computing.
 - d) **Imutils**: Imutils is a computer vision package that includes a series of OpenCV + convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, displaying Matplotlib images, sorting contours, detecting edges, among others quite easy.
 - e) **Pytesseract**: Pytesseract or Python-tesseract is an optical character recognition (OCR) tool for the Python language. This tool is a wrapper for Google's Tesseract-OCR Engine and helps in recognising and reading the text embedded in an image. The tool is also useful as a stand-alone invocation script to tesseract, as it can read all image types supported by the Pillow and Leptonica imaging libraries, including jpeg, png, gif, bmp, tiff, and others.
 - f) **PyTorchCV**: It is a PyTorch-based framework for computer vision tasks. The framework is a collection of image classification, segmentation, detection, and pose estimation models. There are a number of implemented models in this framework, including AlexNet, ResNet, ResNeXt, PyramidNet, SparseNet, DRN-C/DRN-D and more.
 - g) **SimpleCV**: It is one of the popular machine vision frameworks for building computer vision applications. Written in Python, this library helps in getting access to several high-powered computer vision libraries such as OpenCV. The framework is a collection of libraries and software that can be used to develop vision applications. It provides a concise, readable interface for cameras, image manipulation, feature extraction and format conversion. It also allows user to work with the images or video streams that come from webcams, Kinects, FireWire and IP cameras, or mobile phones.
3. **Graphical User Interface**: Python is an interactive programming language and getting started with programming a GUI (Graphical User Interface) framework is not much of a difficult task. Python has a diverse range of options for GUI frameworks.

Python Library for Computer Vision

- a) **Kivy**: Kivy is an OpenGL ES 2 accelerated framework for the creation of new user interfaces. It supports multiple platforms namely Windows, MacOSX, Linux, Android iOS and Raspberry Pi. It is open source and comes with over 20 widgets in its toolkit.
 - b) **PyQT**: PyQT is one of the favoured cross-platform Python bindings implementing the Qt library for the Qt (owned by Nokia) application development framework. Currently, PyQT is available for Unix/Linux, Windows, Mac OS X and Sharp Zaurus. It combines the best of Python and Qt and it up to the programmer to decide whether to create a program by coding or using Qt Designer to create visual dialogs. It is available in both, commercial as well as GPL license. Although some features may not be available in the free version, if your application is open source, then you can use it under the free license.
 - c) **Tkinter**: Tkinter is commonly bundled with Python, using Tk and is Python's standard GUI framework. It is popular for its simplicity and graphical user interface. It is open source and available under the Python License. One of the advantages of choosing Tkinter is that since it comes by default, there is an abundance of resources, both codes and reference books. Also, with the community being old and active, there are many users who can help you out in case of doubts.
 - d) **WxPython**: WxPython is an open source wrapper for cross-platform GUI library WxWidgets (earlier known as WxWindows) and implemented as a Python extension module. With WxPython you as a developer can create native applications for Windows, Mac OS and Unix.
 - e) **PyGUI**: PyGUI is a graphical application cross-platform framework for Unix, Macintosh and Windows. Compared to some other GUI frameworks, PyGUI is by far the simplest and lightweight of them all, as the API is purely in sync with Python. PyGUI inserts very less code between the GUI platform and Python application, hence the display of the application usually displays the natural GUI of the platform.
 - f) **PySide**: PySide is a free and cross-platform GUI toolkit Qt initiated and sponsored by Nokia, Qt is a UI framework and a cross-platform application. PySide currently supports Linux/X11, Mac OS X, Maemo and Windows and, support for Android is in the plans for the near future. PySide provides tools to works with multimedia, XML documents, network, databases and GUI. A key feature of PySide is its API compatibility with PyQt4, so if you wish to migrate to PySide then the process will be hassle-free.
4. **Game Development**: Python is also used in the development of interactive games. There are libraries such as PySoy which is a 3D game engine supporting Python 3, PyGame which provides functionality and a library for game development. Games such as Civilization-IV, Disney's Toontown Online, Vega Strike etc. have been built using Python.

Python Library for Game Development

- a) **Pygame:** Pygame is an open-source Python library for making multimedia applications like games built on top of the excellent SDL library. This library is a combination of C, Python, Native and OpenGL. Pygame enables users to build fully featured games as well as multimedia programs with Python programming. It is highly portable and runs on nearly every platform and operating system. Pygame features are Multicore CPUs can be used easily, uses optimized C and Assembly code for core functions, Simple and Portable, and A small amount of code is needed.
 - b) **Pyglet:** Pyglet is an open-sourced, cross-platform windowing and multimedia library for Python. It is a powerful Python library that can be used for developing games and other visually-rich applications on Windows, Mac OS X and Linux. Pyglet supports windowing, user interface event handling, Joysticks, OpenGL graphics, loading images and videos, and playing sounds and music. Pyglet runs under Python 3.5+ and also works on other Python interpreters such as PyPy.
 - c) **PyOpenGL:** It is one of the most common cross-platform Python bindings to OpenGL and related APIs. In PyOpenGL, the binding is created using the standard ctypes library. PyOpenGL is interoperable with a large number of external GUI libraries for Python including PyGame, PyQt, Raw XLib.
 - d) **Panda3D:** It is Written in Python and C++, Panda3D is an open-source and completely free-to-use engine for realtime 3D games, visualizations, simulations, experiments and more. Panda3D includes command-line tools for processing and optimizing source assets that allow users to automate and script the content production pipeline. It supports many popular third-party libraries, such as the Bullet physics engine, Assimp model loader, OpenAL and FMOD sound libraries, and more.
 - e) **Kivy:** Kivy is an open-source and cross-platform Python library for rapid development of applications. Kivy runs on Linux, Windows, OS X, Android, iOS, and Raspberry Pi and can natively use most inputs, protocols and devices including WM_Touch, WM_Pen, Mac OS X Trackpad and Magic Mouse, Mtdev, Linux Kernel HID, and TUIO. The library is GPU accelerated, and it comes with more than 20 widgets that are all highly extensible.
5. **Web Application:** Python can be used to make web-applications at a rapid rate. Why is that? It is because of the frameworks Python uses to create these applications. There is common-backend logic that goes into making these frameworks and a number of libraries that can help integrate protocols such as HTTPS, FTP, SSL etc. and even help in the processing of JSON, XML, E-Mail and so much more.

Python Library for WebApplication

- a) **Django:** It is a free and open-source Python framework, enables developers to develop complex code and apps quickly. Django framework assists in developing quality web applications. It is among the best python frameworks and is used for the quick development of APIs and web applications. Key Features of Django are Assists you to define patterns for the URLs in your app, Built-in authentication system, Simple yet powerful URL system, Object-oriented programming language database that offers the best data storage and recovery, The automatic admin interface feature enables the functionality of editing, adding, and deleting things with customization, and Cache framework accompanies multiple cache mechanisms.
- b) **CherryPy:** This is almost ten years old now, has proved to be exceptionally quick and stable. It is an open-source Python web development framework that embeds its own multi-hung server. It can run on any working framework that supports Python. A moderate web framework empowers you to use any kind of technology for data access, templating, and so on. Yes, it can do everything that a web framework is capable of, for example, taking care of sessions, file uploads, static, cookies, CherryPy enables developers to develop web apps similarly they would develop any other object-oriented Python program. This results in the development of smaller source code in less time. Key Features of CherryPy are A consistent, HTTP/1.1-compliant, WSGI thread-pooled webserver, Easy to run various HTTP servers (for example on multiple ports) at once, Runs on Python 2.7+, 3.5+, PyPy, Jython and Android, Built-in tools for encoding, sessions, caching, authentication, static content, and many more, A powerful configuration system for developers and deployers alike, Built-in profiling, coverage, and testing support.
- c) **Pyramid:** Pyramid's popularity is progressively growing. Most of the experienced developers are embracing it. Pyramid frameworks run on Python 3. This framework is flexible and allows users to develop basic web apps via a minimalistic approach. Key Features of Pyramid are Seamless development of single-file applications, Generation of URLs, Extendable configuration, Alterable authentication and authorization, Asset specifications and templating.
- d) **Sanic:** Sanic is an open-source and one of the basic python frameworks. It was intended to offer fast HTTP responses by asynchronous request handling. It enables the use of the async and awaits syntax included in Python 3.5, which makes the code a lot simpler, fast, and non-blocking. The purpose of the project is to have an easy way to get up and run, simple to build, extend, and eventually scale the HTTP server. Key Features of Sanic are Provides plugin support, Class-based views, Able to write and read cookies, Allows various types of logging like error log and access log.
- e) **Grok:** Grok framework is a web framework based on the Zope toolkit technology. It gives an agile development experience to developers by concentrating on two general principles – convention over configuration and DRY (Don't Repeat Yourself). It is an open-source framework, developed to speed up the application development process. Developers can choose from a wide scope of network and independent libraries as indicated by the task needs. Grok's UI (user interface) is like other full-stack Python frameworks such as Pylons and TurboGears. Key Features of Grok are Offers a strong foundation for developing powerful and extensible web applications, Enables web developers to

take advantage of the power of Zope 3, A powerful object database for storage, Integrated security to ensure your application and grant access to specific users, Grok component architecture helps developers lessen the unpredictability of development, Offers the building blocks and other essential assets to develop custom web applications for business needs.

- f) **Flask:** Flask is a Python framework accessible under the BSD license, which is inspired by the Sinatra Ruby framework. Flask relies upon the Werkzeug WSGI toolbox and Jinja2 template. The main purpose is to help develop a strong web application base. Developers can develop the Python backend framework any way they need, however, it was designed for applications that are open-ended. Flask has been used by big companies, which include LinkedIn and Pinterest. Compared to Django, Flask is best suited for small and easy projects. Thus, you can expect a web server development, support for Google App Engine as well as in-built unit testing. Key Features of Flask are Built-in development server and debugger, RESTful request dispatching, Integrated unit testing support (code with quality), Uses Jinja2 templating (tags, filters, macros, and more), 100% WSGI 1.0 compliant, multiple extensions provided by the community that eases the integration of new functionalities.
6. **Data Science:** Python is the most widely used programming language today. When it comes to solving data science tasks and challenges, Python never ceases to surprise its users. Most data scientists are already leveraging the power of Python programming every day. Python is an easy-to-learn, easy-to-debug, widely used, object-oriented, open-source, high-performance language, and there are many more benefits to Python programming. Python has been built with extraordinary Python libraries that are used by programmers every day in solving problems.

Python Library for Data Science

- a) **TensorFlow:** TensorFlow is a library for high-performance numerical computations with around 35,000 comments and a vibrant community of about 1,500 contributors. It's used across various scientific fields. TensorFlow is a framework for defining and running computations that involve tensors, which are partially defined computational objects that eventually produce a value. Key features of TensorFlow are Better computational graph visualizations, reduces error by 50 to 60 percent in neural machine learning, Parallel computing to execute complex models, Seamless library management backed by Google, and Quicker updates and frequent new releases to provide you with the latest features, TensorFlow is particularly useful for Speech and image recognition, Text-based applications, Time-series analysis, and Video detection.
- b) **NumPy:** NumPy (Numerical Python) is the fundamental package for numerical computation in Python; it contains a powerful N-dimensional array object. It has around 18,000 comments on GitHub and an active community of 700 contributors. It's a general-purpose array-processing package that provides high-performance multidimensional objects called arrays and tools for working with them. NumPy also addresses the slowness problem partly by providing these multidimensional arrays as well as providing functions and operators that operate efficiently on these arrays. Key features of NumPy are Provides fast, precompiled functions for numerical routines, Array-oriented computing for better efficiency, Supports an object-oriented approach, and Compact and faster computations with vectorization. NumPy is particularly useful for Extensively used in data analysis, Creates a powerful N-dimensional array, Forms the base of other libraries, such as SciPy and scikit-learn, and Replacement of MATLAB when used with SciPy and matplotlib
- c) **SciPy:** SciPy (Scientific Python) is another free and open-source Python library extensively used in data science for high-level computations. SciPy has around 19,000 comments on GitHub and an active community of about 600 contributors. It's widely used for scientific and technical computations because it extends NumPy and provides many user-friendly and efficient routines for scientific calculations. Key features of SciPy are Collection of algorithms and functions built on the NumPy extension of Python, High-level commands for data manipulation and visualization, Multidimensional image processing with the SciPy.ndimage submodule, and Includes built-in functions for solving differential equations. SciPy is particularly useful for Multidimensional image operations, Solving differential equations and the Fourier transform, Optimization algorithms, and Linear algebra
- d) **Pandas:** Pandas (Python data analysis) is a must in the data science life cycle. It is the most popular and widely used Python library for data science, along with NumPy in matplotlib. With around 17,00 comments on GitHub and an active community of 1,200 contributors, it is heavily used for data analysis and cleaning. Pandas provide fast, flexible data structures, such as data frame CDs, which are designed to work with structured data very quickly and intuitively. Key features of Pandas are Eloquent syntax and rich functionalities that gives you the freedom to deal with missing data, enables you to create your function and run it across a series of data, High-level abstraction, and Contains high-level data structures and manipulation tools. SciPy is particularly useful for General data wrangling and cleaning, ETL (extract, transform, load) jobs for data transformation and data storage, as it has excellent support for loading CSV files into its data frame format, Used in a variety of academic and commercial areas, including statistics, finance, and neuroscience, Time-series-specific functionality, such as date range generation, moving window, linear regression, and date shifting.
- e) **Matplotlib:** Matplotlib has powerful yet beautiful visualizations. It's a plotting library for Python with around 26,000 comments on GitHub and a very vibrant community of about 700 contributors. Because of the graphs and plots that it produces, it's extensively used for data visualization. It also provides an object-oriented API, which can be used to embed those plots into applications. Key features of Matplotlib are Usable as a MATLAB replacement, with the advantage of being free and open-source, Supports dozens of backends and output types, which means you can use it

regardless of which operating system you're using or which output format you wish to use, Pandas itself can be used as wrappers around MATLAB API to drive MATLAB like a cleaner, and Low memory consumption and better runtime behavior. Matplotlib is particularly useful for Correlation analysis of variables, visualize 95 percent confidence intervals of the models, Outlier detection using a scatter plot, and Visualize the distribution of data to gain instant insights.

7. **Machine Learning:** Python seems to be winning battle as preferred language of Machine Learning. The availability of libraries and open source tools make it ideal choice for developing ML models. Python has been the go-to choice for Machine Learning and Artificial Intelligence developers for a long time. Python offers some of the best flexibilities and features to developers that not only increase their productivity but the quality of the code as well, not to mention the extensive libraries helping ease the workload.

Python Library for Machine Learning

- a) **Keras:** Keras is one of the most popular and open-source neural network libraries for Python. Initially designed by a Google engineer for ONEIROS, short for Open-Ended Neuro Electronic Intelligent Robot Operating System, Keras was soon supported in TensorFlow's core library making it accessible on top of TensorFlow. Keras is a very popular Machine Learning library for Python. It is a high-level neural networks API capable of running on top of TensorFlow, CNTK, or Theano. It can run seamlessly on both CPU and GPU. Keras makes it really for ML beginners to build and design a Neural Network. One of the best things about Keras is that it allows for easy and fast prototyping. Keras extends the usability of TensorFlow with these additional features for ML and DL programming. With a helpful community and a dedicated Slack channel, getting support is easy. Support for the convolutional and recurrent neural network also exists along with standard neural networks.
- b) **PyTorch:** PyTorch is one of the few machine learning libraries for Python. Apart from Python, PyTorch also has support for C++ with its C++ interface if you're into that. Considered among the top contenders in the race of being the best Machine Learning and Deep Learning framework, PyTorch faces tough competition from Tensor Flow. PyTorch is a popular open-source Machine Learning library for Python based on Torch, which is an open-source Machine Learning library which is implemented in C with a wrapper in Lua. It has an extensive choice of tools and libraries that supports on Computer Vision, Natural Language Processing (NLP) and many more ML programs. It allows developers to perform computations on Tensors with GPU acceleration and also helps in creating computational graphs.
- c) **Scikit-learn:** Scikit-learn is another actively used machine learning library for Python. It includes easy integration with different ML programming libraries like NumPy and Pandas. Scikit-learn is one of the most popular ML libraries for classical ML algorithms. It is built on top of two basic Python libraries, viz., NumPy and SciPy. Scikit-learn supports most of the supervised and unsupervised learning algorithms. Scikit-learn can also be used for data-mining and data-analysis, which makes it a great tool who is starting out with ML.
- d) **MMLib:** Developed by Apache, Spark MMLib is a machine learning library that enables easy scaling of your computations. It is simple to use, quick, easy to set up and offers smooth integration with other tools. Spark MMLib instantly became a convenient tool for developing machine learning algorithms and applications. Some of the popular algorithms and APIs that programmers working on Machine Learning using Spark MMLib can utilize are Regression, Clustering, Optimization, Dimensional Reduction, Classification, Basic Statistics, and Feature Extraction.
- e) **Theano:** Theano is a powerful Python library enabling easy defining, optimizing and evaluation of powerful mathematical expressions. We all know that Machine Learning is basically mathematics and statistics. Theano is a popular python library that is used to define, evaluate and optimize mathematical expressions involving multi-dimensional arrays in an efficient manner. It is achieved by optimizing the utilization of CPU and GPU. It is extensively used for unit-testing and self-verification to detect and diagnose different types of errors. Theano is a very powerful library that has been used in large-scale computationally intensive scientific projects for a long time but is simple and approachable enough to be used by individuals for their own projects.

IV. CONCLUSION

In this paper, we discussed the python programming as a more suitable choice for beginners and professional expert. This paper elaborated the why this python more popular in this real-world era, features of this programming language such as fast, easy, powerful, portable, simple, and free open source language that supports other technologies, and discussed the applicable various domains of the python. This study has also specially analyzed popular python libraries which are using their application domains.

REFERENCES

- [1] K. R. Srinath, "Python – The Fastest Growing Programming Language," International Research Journal of Engineering and Technology (IRJET), vol. 4, Issue 12 pp. 354–357, December 2017.
- [2] Akshansh Sharma, Firoj Khan, Deepak Sharma, Dr. Sunil Gupta, "Python: The Programming Language of Future" INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN TECHNOLOGY, May 2020 | IJIRT | Volume 6 Issue 12 | ISSN: 2349-6002

- [3] A.L.Sayeth Saabith, T.Vinothraj, MMM.Fareez, “PYTHON CURRENT TREND APPLICATIONS- AN OVERVIEW”, International Journal of Advance Engineering and Research Development, Volume 6, Issue 10, October 2019, E-ISSN (O): 2348-4470
- [4] Kalyani Adawadkar, “Python Programming-Applications and Future,” International Journal of AdvanceEngineering and Research Development(IJAERD), Special Issue SIEICON-2017,pp. 1–4, April -2017.
- [5] QuantInsti, Python Basics With Illustrations from the Financial Markets, QuantInsti Quantitative Learning Pvt. Ltd.
- [6] Pinky Sodhi, Naman Awasthi, Vishal Sharma, “Introduction to Machine Learning and Its Basic Application in Python”, Proceedings of 10th International Conference on Digital Strategies for Organizational Success, January, 2019
- [7] <https://www.geeksforgeeks.org/history-of-python/>
- [8] <https://www.python.org/doc/essays/blurb/>
- [9] <https://docs.python.org/3/library/>
- [10]<https://insights.stackoverflow.com/survey/2019>
- [11]<https://insights.stackoverflow.com/survey/2020>
- [12]<https://trends.google.com/trends/>
- [13]<https://towardsdatascience.com/>
- [14]<https://analyticsindiamag.com/10-best-python-libraries-for-computer-vision/>
- [15]<https://www.tiobe.com/tiobe-index/>