Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Experiment No. 1

Aim: Introduction to Data analytics libraries in Python and R.

Objective- Understand the use of Python and R, To effectively use libraries for data science.

Description:

Why Choose Python?

Python is a general-purpose, open-source programming language used in various software domains, including data science, web development, and gaming.

Launched in 1991, Python is one of the most popular programming languages in the world, occupying the top position in several programming language popularity indices, such as the TIOBE Index and the PYPL Index.

One of the reasons for the worldwide popularity of Python is its community of users. Python is backed by a vast community of users and developers who ensure the smooth growth and improvement of the language, as well as the continuous release of new libraries designed for all kinds of purposes.

Python is an easy language to read and write due to its high similarity with human language. In fact, high readability and interpretability are at the heart of the design of Python. For these reasons, Python is often cited as a go-to programming language for newcomers with no coding experience.

Over time, Python has been gaining popularity in the field of data science thanks to its simplicity and the endless possibilities provided by the hundreds of specialized libraries and packages that support any kind of data science task, such as data visualization, machine learning, and deep learning.

Why Choose R?

R is an open-source programming language specifically created for statistical computing and graphics.

Since its first launch in 1992, R has been widely adopted in scientific research and academia. Today, it remains one of the most popular analytics tools used in both traditional data analytics and the rapidly-evolving field of business analytics. It ranks 11th and 7th position in the **TIOBE** Index and the **PYPL** Index, respectively.

Designed with statisticians in mind, with R, you can use complex functions within a few lines of code. All kinds of statistical tests and models are readily available and easily used, such as linear modeling, non-linear modeling, classifications, and clustering.

The extensive possibilities R offers are mostly due to its huge community. It has developed one of the richest collections of data-science-related packages. All of them are available via the Comprehensive R Archive Network (**CRAN**).

Another feature that makes R particularly remarkable is the power to generate quality reports with support for data visualization and its available frameworks to create interactive web applications. In this sense, R is widely considered the best tool for making beautiful graphs and visualizations

R vs Python: Key Differences

Purpose

While Python and R were created with different purposes –Python as a general-purpose programming language and R for statistical analysis–nowadays, both are suitable for any data science task. However, Python is considered a more versatile programming language than R, as it's also extremely popular in other software domains, such as software development, web development, and gaming.

Type of Users

As a general-purpose programming language, Python is the standard go-to choice for software developers breaking into data science. Plus, Python's focus on productivity makes it a more suitable tool to build complex applications.

By contrast, R is widely used in academia and certain sectors, such as finance and pharmaceuticals. It is the perfect language for statisticians and researchers with limited programming skills.

Learning curve

Python's intuitive syntax is considered one of the closest programming languages to English. This makes it a very good language for new programmers, with a smooth and linear learning curve. Although R is designed to run basic data analysis easily and within minutes, things get harder with complex tasks, and it takes more time for R users to master the language.

Overall, Python is considered a good language for beginner programmers. R is easier to learn when you start out, but the intricacies of advanced functionalities make it more difficult to develop expertise.

Popularity

Although new programming languages, like **Julia**, are recently gaining momentum in data science, Python and R remain the absolute kings in the discipline.

However, in terms of popularity –always a very slippery concept– the differences are striking. Python has consistently outranked R, especially in recent years. Python ranks first in several programming language popularity indexes. This is due to the widespread use of Python in multiple software domains, including data science. By contrast, R is mostly employed in data science, academia, and certain sectors.

Common Libraries

Both Python and R have robust and extensive ecosystems of packages and libraries specifically designed for data science. Most packages in Python are hosted in the Python Package Index (**PyPi**), whereas **R packages** are normally stored in the Comprehensive R Archive Network (**CRAN**).

Below you can find a list of some of the most popular data science libraries in R and Python.

R packages:

- **dplyr**: It is a data manipulation library for R.
- **tidyr**: a great package that will help you get your data clean and tidy.
- **ggplot2**: the perfect library for visualizing data.
- **Shiny**: It is the ideal tool for creating interactive web apps directly from R.
- **Caret**: one of the most important libraries for machine learning in R.

Python packages:

- **NumPy**: provides a large collection of functions for scientific computing.
- **Pandas**: perfect for data manipulation.
- **Matplotlib**: the standard library for data visualization.
- **Scikit-learn**: is a library in Python that provides many machine learning algorithms.
- **TensorFlow**: a widely used framework for deep learning.

Common IDEs

An IDE, or Integrated Development Environment, enables programmers to consolidate the different aspects of writing a computer program. They are powerful interfaces with integrated capabilities that allow developers to write code more efficiently.

In Python, the most popular IDEs in data science are Jupyter Notebooks and its modern version, JupyterLab, as well as Spyder.

As for R, the most commonly used IDE is RStudio. Its interface is organized so that the user can view graphs, data tables, R code, and output all at the same time.

Python vs R: A Comparison

	R	**Python**
Purpose	Very popular in academia and research, finance and data science	Well-suited for many programming domains, including data science, web development, software development, and gaming
First Release	1993	1991
Type of Language	General-purpose programming language	General-purpose programming language
Open Source?	Yes	Yes

Ecosystem	Nearly 19,000 packages available in the Comprehensive R Archive Network (**CRAN**)	+300,000 available packages in the Python Package Index (**PyPi**)
Ease of Learning	R is easier to learn when you start out, but gets more difficult when using advanced functionalities.	Python is a beginner-friendly language with English-like syntax.
IDE	RStudio. Its interface is organized so that the user can view graphs, data tables, R code, and output all at the same time.	Jupyter Notebooks and its modern version, JupyterLab, and Spyder.
Advantages	· Widely considered the best tool for making beautiful graphs and visualizations. · Has many functionalities for data analysis. · Great for statistical analysis.	· General-purpose programming languages are useful beyond just data analysis. · Has gained popularity for its code readability, speed, and many functionalities. . · Has high ease of deployment and reproducibility.

Disadvantages	· More difficult to learn for people with no software development background. · Limited user community compared to Python · R is considered a computationally slower language compared to Python, especially if the code is written poorly. · Finding the right library for your task can be tricky, given the high number of packages available in CRAN	· Weak performance with huge amounts of data · Poor memory efficiency · Python does not have as many libraries for data science as R. · Python requires rigorous testing as errors show up in runtime. · Visualizations are more convoluted in Python than in R, and results are not as eye pleasing or informative.
Trends	11th in TIOBE and 7th in PYPL (December 2022)	1th in TIOBE and 1th in PYPL (December 2022)

Attach Libraries you searched in Lab session-

Python Libraries:

1. NumPy:
- Description: NumPy is akin to a mathematical powerhouse in Python, granting users the ability to manipulate arrays and perform numerical computations with finesse, serving as the backbone of scientific computing.

2. Pandas:
- Description: Pandas acts as a versatile data whisperer, providing an intuitive and powerful toolkit to tame, clean, and analyze datasets effortlessly using its DataFrame structures and rich functionality.

3. Matplotlib:

- Description: Matplotlib serves as the artist's palette, offering a vast canvas of plotting tools to visualize data in various forms, allowing users to craft expressive and customizable 2D graphics.

4. Scikit-learn:

- Description: Scikit-learn stands tall as the ML maestro in Python, featuring a trove of machine learning algorithms and tools for classification, regression, clustering, and more, simplifying the model-building journey.

5. TensorFlow:

- Description: TensorFlow emerges as the neural network virtuoso, providing a robust platform for building, training, and deploying sophisticated neural models and machine learning solutions.

6. Keras:

- Description: Keras, akin to a master orchestrator, offers a streamlined interface atop TensorFlow, empowering users to construct and experiment with intricate deep learning models with ease.

7. Seaborn:

- Description: Seaborn, like a visual storyteller, elevates data visualization by adding statistical sophistication to Matplotlib, enabling the creation of captivating and informative statistical plots.

8. NLTK (Natural Language Toolkit):

- Description: NLTK functions as a linguistic guidebook, equipping users with a comprehensive suite of tools for natural language processing tasks, facilitating exploration and analysis of human language data.

9. Beautiful Soup:

- Description: Beautiful Soup emerges as the web content curator, providing a flexible toolkit to navigate and extract information from HTML and XML documents, simplifying the process of web scraping.

10. PyTorch:

- Description: PyTorch serves as the creative playground for AI enthusiasts, offering a dynamic and flexible framework for deep learning research and development, encouraging innovation and experimentation.

R Libraries:

1. dplyr:

- Description: dplyr acts as the data choreographer in R, offering an elegant set of tools for data manipulation, making tasks like filtering, summarizing, and transforming data a seamless endeavor.

2. ggplot2:

- Description: ggplot2 emerges as the visual virtuoso in R, empowering users to create visually captivating and insightful plots using a structured grammar of graphics approach.

3. tidyr:

- Description: tidyr stands as the data tidying maestro in R, simplifying the process of restructuring and organizing messy data into a more structured format conducive to analysis and exploration.

4. caret:

- Description: caret plays the role of a model maestro in R, offering a unified interface for machine learning model training, evaluation, and tuning, simplifying the path towards predictive modeling.

5. randomForest:

- Description: randomForest harnesses the power of ensemble learning in R, constructing forests of decision trees for robust and accurate classification and regression tasks.

6. Shiny:

- Description: Shiny acts as R's gateway to interactive web applications, enabling users to transform R scripts into interactive web interfaces for data exploration and visualization.

7. Rvest:

- Description: Rvest simplifies the art of web scraping in R, providing a suite of tools to extract and parse data from HTML web pages, facilitating efficient data collection from the web.

8. ROCR:

- Description: ROCR emerges as a trustworthy guide for evaluating machine learning models in R, offering tools for visualizing and comprehending classifier performance for informed decision-making.

9. CaretEnsemble:

- Description: CaretEnsemble fosters model synergy in R by enabling the combination and ensemble of diverse machine learning models, enhancing predictive performance and robustness.

10. Tidytext:

- Description: Tidytext embodies structured text mining in R, offering a systematic approach to analyze and extract insights from textual data by adhering to tidy principles.

In summary, exploring the diverse spectrum of Python libraries unveiled a rich ecosystem catering to various needs. From NumPy and Pandas for efficient data handling to TensorFlow and Keras for machine learning enthusiasts, each library offered distinct functionalities, empowering users in scientific computing, data analysis, and machine learning endeavors. The versatility and depth of these libraries underscored their pivotal roles in driving innovation and facilitating complex tasks seamlessly.