



“Roadmap for learning Python and Relation between Numpy and Pandas”

(INDUSTRIAL REPORT-WEEK 4)

Prepared by

[Shivam Shriwastav]

Email id:Shivam808047@gmail.com

Executive Summary

This report provides details of the Industrial Internship provided by Upskill Campus and The IoT Academy in collaboration with Industrial Partner UniConverge Technologies Pvt Ltd (UCT).

This internship is focused on a project/problem statement provided by UCT. We had to finish the project including the report in 6 weeks' time.

This weekly report explains The Roadmap for Learning Python.

-What is the relation between Numpy and Pandas?

TABLE OF CONTENTS

| | |
|---|-------|
| “ Heading: The Roadmap for Learning Python & the Relation between Numpy and Pandas” | 2 |
| 1 Preface | 4 |
| 1.1.1 Information about the internship position | 4 |
| 2 Introduction: Python | 5 |
| 3. The Roadmap for Learning Python..... | 6-10 |
| 4. What is the relation between Numpy and Pandas? | 11-16 |
| #Python..... | 17 |
| #Python Libraries | 18 |
| * Code submission (Github link): | 18 |
| * Report submission(Github link): | 18 |
| 5.My learnings..... | 19 |
| What is Python? | 19 |
| Uses: | 19 |
| GOOD TO KNOW.... | 20 |
| Python Syntax compared to other programming languages..... | 20 |

1 Preface

Summary of the 4th week's work.

I undertook this internship project and completed the 4th -week internship report under the guidance of this associated company. I am grateful to all for their patience and assistance during my online training at their Virtual site named "Upskill Campus". It was a good learning experience for me to work on their weekly project, as the project involved many innovative practices.

1.1.1 Information about the internship position

I joined Upskill campus for an internship program in the position of a **Python Intern**. While the central focus was on focusing in this program wisely and learn effectively, I also handled various other tasks as they occurred.

I want to thank my advisers and everyone at the company for their patience and assistance during my on-site training. Thanks to their guidance, I was able to develop [**PYTHON SKILLS**] and learn about [**PYTHON**]. These skills would help me to expand my resume and advance my career.

2 Introduction:Python

Basic concepts:

Python is a widely used general-purpose, high level programming language. It was created by Guido van Rossum in 1991 and further developed by the Python Software Foundation. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

There are two major Python versions: Python 2 and Python 3. Both are quite different.

3. Roadmap for learning Python:

- Introduction

Python is among the most popular programming languages in use today, and for good reason. Because it is easy to learn, and flexible, it is a great language for beginners. Whether you want to build your first website, automate procedures, or begin working with data, Python has you covered. Businesses use Python, a programming language, in every industry. It is useful in machine learning and web development, too. It has the fastest-growing developer community. You can use it to automate all daily tasks, so Python is the most alluring language to learn. In this blog, we will guide you through the process of learning Python. Know **Python's advantages** and enhance your programming abilities.

- Why Learn Python?

- Among the most widely used languages
- Simple to learn how to program
- Super-versatile
- Simple logic for programming
- Many third-party libraries provide support
- High-paying positions available globally in web development, machine learning, and backend support

- A Strategy For Learning Python In 2023

- Set a target: Establish your goals and the reasons for wanting to learn Python. You can focus and stay motivated using this.

- Select a learning tool: Choose the most effective learning method, such as courses, books, or online tutorials. Make sure the resource you select fits your learning preferences and time constraints.
- Install Python and set up a workspace: Download the most recent version of Python, install it, and then create a workspace. Create the workspace using PyCharm or IDLE.
- Begin with the fundamentals: The fundamentals of syntax, data types, variables, operators, and control structures should be understood.
- Write code to improve: Start with easy exercises and progress to more difficult projects.
- Work with libraries and modules: Get knowledge of how to process and analyze data using well-known libraries. For instance, NumPy, Pandas, and Matplotlib.
- Create projects: Put your knowledge to use by creating simple projects like games, calculators, or web scrapers.
- Join a group: Join online forums, Discord groups, or Meetups to discuss projects, ask questions, and gain knowledge from others.
- Maintain improvement: Keep up with Python's most recent innovations and keep working to build your skills.

• Python Programming Language Roadmap

You can be a successful [Python developer](#) if you follow a proper strategy. You can begin your learning process by selecting one of these Python concepts in the sequence provided.

- Basics: Python syntax, keywords, and variable identifiers
- Data Types: Converting Numbers and String Types in Python Data Structures: List Tuple Set Dictionary
- Python Control Structures: If, if-else, nested if statements, For Loop, While Loop, Break & Continue Usage and Pass Keyword

- Python functions: Functioning variables, Recursive procedure, Unidentified action, Variable Scope, Modules, global keywords, and Packages
- Object-Oriented Programming in Python: Class and Objects Methods, Class Inheritance and Overriding
- Advanced Topics: Iterators, Generators, Closure, Decorators, Regular Expressions, Exception Handling. Also, it covers File Handling and Database Handling

• Why Python?

You should have a clear objective in mind before beginning your study of Python. Why do you want to learn Python? What do you hope to achieve with this language? Do you wish to create some online applications or automate some tedious or monotonous tasks? The most typical error made by new learners is to begin studying a language for the sake of doing so. It shouldn't be without having a specific goal in mind. Remember that learning a language is one thing, but using it to create a real-world application is quite another. As a programmer, your goal should be to be able to create things rather than learn languages.

• Python Language Use

- Python is used by professionals from a variety of fields to carry out a wide range of activities. For instance data analysis, visualization, automation, and the creation of AI and machine learning applications.
- Many repetitive operations can be carried out automatically using Python syntax or scripts. Such as transferring necessary folders and files, renaming them as necessary, and submitting them to a server.

- Not software programmers are using it. So do accountants, data analysts, network engineers, and scientists.

• Development Frameworks

Development Frameworks for the Web Python has a large number of web application frameworks. Some of which follows Django, Bottle, Tornado, Flask, and Pyramid.

- **Django:** A high-level web development framework mostly utilized by businesses and startups. You can utilize a variety of databases, including PostgreSQL, MySQL, SQLite, and Oracle, and it adheres to the MVC design. You will experience a lot of pain while learning Django if you are a total beginner and are unfamiliar with the terms authentication, URL routing, API, and models. But take your time, be patient, read through some additional resources, and comprehend every line of code. You will gradually and thoroughly comprehend everything. Django training could take two to five weeks.
- **Flask:** One of the simplest microframeworks to learn in Python is the flask. Flask is appropriate for creating straightforward, lightweight web applications. Although it lacks Django's strength and breadth, it nonetheless offers capabilities like support for developing REST APIs and unit testing. It will take 1 to 1.5 weeks to learn Flask.

• Develop Projects

Now the learning process is complete for Python, and the last step is constructing projects. Only if you can construct some projects does all your Python learning

make sense. Remember that working on a challenging project that resolves a problem is the greatest method to check your programming abilities. Starting simple will make it easier for beginners to complete a complex project. Create a straightforward project first, and then advance progressively. Start with a basic, simple module when building a big project, and keep adding features to it. You can track your development as you work on your project and learn how programmers approach it. Know how they resolve python related problems in the real world. Hands-on experience enhances the [scope of Python developer](#).

•

Tips To Remember

- Having patience is a virtue not while learning Python but when learning any new language. Recognize that it will take some time for things to sink in. Learning a first language always takes more time and effort.
- Keep your language and aim consistent. Don't merely pick up a new programming language and learn the syntax.
- Accepting pain and frustration as necessary parts of learning is preferable to avoid them.
- You may want to give up because of several difficult terms, mistakes, and problems. Avoid doing it as everyone in programming experiences it. Spend some time alone understanding the subject using various resources.
- If you are not consistent in anything, it will show. Be consistent; learning will take much more time and effort if you are not.
- Don't discount the significance of constructing a project. It is always beneficial for boosting confidence.

4. What is the relation between Numpy and Pandas?

Introduction

When it comes to the fields of data science and software development, [Python](#) is undoubtedly the best programming language. This is due to the several benefits that Python provides, including a user-friendly language and an easy-to-remember grammar. But in addition to that, Python has a substantial number of integrated libraries that let you complete a variety of jobs quickly. Two of these well-liked Python libraries are NumPy and Pandas. In this blog, we will explore the difference between NumPy and Pandas in detail, but before that, we will briefly introduce them.

What is NumPy?

NumPy stands for Numerical Python. One of the simplest and most effective Python libraries for producing and working with numerical objects is this one. The NumPy library was primarily created to accommodate massive multidimensional matrices. The use of one-dimensional and multi-dimensional arrays facilitates the execution of sophisticated mathematical operations and intricate computations. NumPy provides several features that reduce the difficult tasks of data analysis, data scientists, researchers, etc.

Key features of NumPy

Now that we know a little about what NumPy is, let's take a look at some of the key features it offers:

- The "ndarray" function for working with n-dimensional arrays and data structures is one of NumPy's most notable features.
- NumPy makes it easy to run n-dimensional array and matrix-related programs quickly.
- Based on LAPACK and BLAS (Basic Linear Algebra Subprograms), provides useful linear algebra calculations (Linear Algebra Package).
- In OpenCV, NumPy can be used as a general-purpose data structure for things like extracted function points, filter kernels, and images.
- The inability of NumPy to attach data objects to arrays as quickly as Python is one of the language's drawbacks.
- Numerous tools in NumPy are available for merging C/C++ and Fortran programming.
- In NumPy, arrays are homogeneous. includes a multidimensional container for general data (parameterized array data type). Complex operations on linear algebra, the Fourier transform, and random numbers can also be performed using NumPy.
- NumPy also consists of broadcast functions. This makes it extremely useful when working with arrays of irregular shapes, as it casts the shape of smaller arrays according to larger ones.
- NumPy has the ability to define data types to work with different databases.

Note that NumPy is not part of a standard Python installation; Consequently, you must manually install it. However, using PIP, it is quite simple to install and begin utilizing the most recent version of the NumPy library from the Python repository as demonstrated below:

```
...
```

```
!pip install numpy
```

```
...
```

What are pandas?

Pandas stands for Python Data Analysis Library. It is an open-source library specifically designed for data analysis and data manipulation in Python. Pandas is built on top of the NumPy package and relies heavily on NumPy.

Pandas allows us to read from multiple sources like Excel, CSV, SQL and many more. Pandas has two types of data objects:

Pandas DataFrame: This is a mutable two-dimensional data structure with labeled rows and columns, generally compared to Excel and SQL sheets.

Pandas Series: These are one-dimensional labeled arrays for storing heterogeneous data elements, generally compared to columns in MS Excel.

Before Pandas, python supported minimal data analysis, but now it allows various data operations and time series manipulation. Pandas can perform 5 basic operations for data analysis: Load, manage, prepare, model and analyze.

Key features of pandas

Now that we know a little about what Pandas is, let's take a look at some of the key features it offers:

- Pandas can help us transform and pivot datasets.
- It can also help us merge and join datasets.
- The Pandas DataFrame object allows data manipulation along with indexing.

- Pandas also provides good support for data alignment and integrated handling of missing data from datasets.
- Pandas also provides a wealth of tools for reading and writing data between in-memory data structures and various file formats.
- Pandas provides support for data filtering.
- Pandas also provides features such as label-based partitioning, fancy indexing, and subsets of large datasets.
- Pandas also provides engine-based grouping that allows you to split, apply, and combine operations on datasets.
- Pandas provides hierarchical axis indexing (Hierarchical indexing is a method of creating structured group relationships in data. These hierarchical indexes, or MultiIndexes, are highly flexible and offer a range of options when performing complex data queries) for working with high-dimensional data in a lower-dimensional data structure.

Note that individual columns in Pandas are referred to as "Series" and multiple series in a collection are called "DataFrames". Since Pandas is not included in the standard Python installation, you have to install it externally using PIP.

```
...
```

```
!pip install pandas
```

```
...
```

The key difference between Pandas vs. NumPy

Let's discuss some of the main key differences between Pandas and NumPy:

Data objects in NumPy and Pandas

The primary data object in NumPy is an array, more specifically an ndarray. It is an N-dimensional array that supports various computations and computations. These

arrays are much faster than python list based arrays as they do not involve looping. The primary data object in Pandas is also an array. An array is a one-dimensional indexed array. By joining row objects, one can produce DataFrames, a common data type in pandas. n-dimensional indexed arrays are what DataFrames are. Very similar to NumPy's ndarrays, but indexed.

Data type supported in NumPy and Pandas

The NumPy library is mainly used to perform numerical computations and calculations. With a number of functions provided in this module, we can perform complex calculations on fields quickly and easily. At the same time, the pandas library is primarily for data analysis by allowing us to work with CSV, Excel, SQL, etc. It even has some data plotting and visualization features built in.

Uses in deep learning and machine learning

NumPy is one of the core modules on top of which most other python modules are built. The most popular [machine learning tool](#), sci-kit learning modules, can only be fed (accept input as) NumPy arrays. The same is true for complex deep learning tools like TensorFlow. It also takes a NumPy array as input and gives an array as output. Pandas data objects cannot be used directly as input to machine learning and deep learning tools. Before we feed them into the machine learning module, we have to go through several pre-processing steps.

Performance on complex operations

NumPy performs best in complex mathematical calculations on multidimensional arrays. It is insanely faster than pandas in calculations like solving linear algebra, gradient search, matrix multiplication, data vectorization, etc. Doing these

calculations on dataframes and serial objects in pandas is tedious and difficult. However, it should be noted that NumPy works best with 50,000 or fewer rows in a dataset, while pandas does best with 500,000 or more rows when manipulating data.

Indexing in Pandas and NumPy

Data rows are not indexed in NumPy arrays by default. However, this is not the case with pandas. By default, data rows are indexed or labeled. You can play with and manipulate indexes. You can use a column as an index or change the label names in the indexes. This is not entirely possible in NumPy.

Conclusion

So in conclusion, even though Pandas was built on top of NumPy, the two Python libraries have significant differences. Both Pandas and NumPy simplify matrix multiplication and are widely used in data science, especially machine learning model development. Therefore, we would recommend all current budding programmers who want to become data scientists, machine learning researchers, or machine learning practitioners to learn these libraries. This will not only open the doors for them to get a job in some of the biggest companies in the world, but also help them in their day-to-day calculations to become good experts in machine learning and data science

What is Python for everyone?

- Develop programs to gather, clean, analyze, and visualize data. This

Specialization builds on the success of the Python for Everybody

course and will introduce fundamental programming concepts

including data structures, networked application program interfaces,

and databases, using the Python programming language.

#Python

Python is a programming language widely used by Data Scientists.

Python has in-built mathematical libraries and functions, making it easier to calculate mathematical problems and to perform data analysis.

We will provide practical examples using Python.

To learn more about Python, please visit our [Python Tutorial](#).

#Python Libraries

Python has libraries with large collections of mathematical functions and analytical tools.

In this course, we will use the following libraries:

- Pandas - This library is used for structured data operations, like import CSV files, create dataframes, and data preparation
- [Numpy](#) - This is a mathematical library. Has a powerful N-dimensional array object, linear algebra, Fourier transform, etc.
- [Matplotlib](#) - This library is used for visualization of data.
- [SciPy](#) - This library has linear algebra modules

We will use these libraries throughout the course to create examples.

***Code submission (Github link) :**

https://github.com/shivam808047/python_internship/blob/main/quiz%20game%20by%20python.py

***Report submission(Github link):**

https://github.com/shivam808047/python_internship

5. My learnings....

What is Python?

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

Uses:

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-oriented way or a functional way.

GOOD TO KNOW....

- The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
- In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

Python Syntax compared to other programming languages

- Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.