[Your College Logo]

GGI

Sharpening Minds. Brightening Tomorrow

GULZAR GROUP OF INSTITUTES

[Your College Logo]

“Introduction to Numpy and Pandas & their Operations”

(INDUSTRIAL REPORT-WEEK 5)

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 Introduction to Numpy and its Operations & Introduction to Pandas and its Operations.

TABLE OF CONTENTS

1 Preface

Summary of the 5th week's work.

I undertook this internship project and completed the 5th -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. Introduction to Numpy and its Operations:

- Introduction:
  - NumPy is a Python library.
  -
  - NumPy is used for working with arrays.
  -
  - NumPy is short for "Numerical Python".

3.1What is NumPy?

NumPy is a Python library used for working with arrays.
It also has functions for working in domain of linear algebra, fourier transform, and matrices.
NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely.
NumPy stands for Numerical Python.

3.2 Why Use NumPy?

In Python we have lists that serve the purpose of arrays, but they are slow to process.

NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.

The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.

Arrays are very frequently used in data science, where speed and resources are very important.

3.3Installation of NumPy

If you have Python and PIP already installed on a system, then installation of NumPy is very easy.

Install it using this command:

```
C:\Users\Your Name>pip install numpy
```

If this command fails, then use a python distribution that already has NumPy installed like, Anaconda, Spyder etc.

3.4Import NumPy

Once NumPy is installed, import it in your applications by adding the `import` keyword:

```
import numpy
```

Now NumPy is imported and ready to use.

3 ExampleGet your own Python Server

```
import numpy

arr = numpy.array([1, 2, 3, 4, 5])

print(arr)
```

Try it Yourself »

3.5 NumPy as np

NumPy is usually imported under the `np` alias.

alias: In Python alias are an alternate name for referring to the same thing.

Create an alias with the `as` keyword while importing:

```
import numpy as np
```

Now the NumPy package can be referred to as `np` instead of `numpy`.

4 Example

```
import numpy as np

arr = np.array([1, 2, 3, 4, 5])

print(arr)
```

Try it Yourself »

3.6Checking NumPy Version

The version string is stored under `__version__` attribute.

5 Example

```
import numpy as np

print(np.__version__)
```

4. Introduction to Pandas and its Operations

Pandas is an open-source library in Python that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. This library is built on top of the NumPy library of Python. Pandas is fast and it has high performance & productivity for users.

History of Pandas Library

Pandas were initially developed by Wes McKinney in 2008 while he was working at AQR Capital Management. He convinced the AQR to allow him to open source the Pandas. Another AQR employee, Chang She, joined as the second major contributor to the library in 2012.

Over time many versions of pandas have been released. The latest version of the pandas is 1.5.3, released on Jan 18, 2023.

- ## Why Use Pandas?
  - Fast and efficient for manipulating and analyzing data.
  - Data from different file objects can be easily loaded.
  - Flexible reshaping and pivoting of data sets
  - Provides time-series functionality.
- ## What can you do using Pandas?

Pandas are generally used for data science but have you wondered why? This is because pandas are used in conjunction with other libraries that are used for data science. It is built on the top of the **NumPy** library which means that a lot of structures of NumPy are used or replicated in Pandas. The data produced by Pandas are often used as input for plotting functions of **Matplotlib**, statistical analysis in **SciPy**, and machine learning algorithms in **Scikit-learn**. Here is a list of things that we can do using Pandas.

- Data set cleaning, merging, and joining.
- 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.
- Data Visulaization

- ## Getting Started
- ### Installing Pandas

The first step of working in pandas is to ensure whether it is installed in the system or not. If not then we need to install it in our system using the **pip command**. Type the cmd command in the search box and locate the folder using the cd command where **python-pip file** has been installed. After locating it, type the command:

For more reference take a look at this article on installing pandas follows.

- **Importing Pandas**

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

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.

- **Pandas Data Structures**

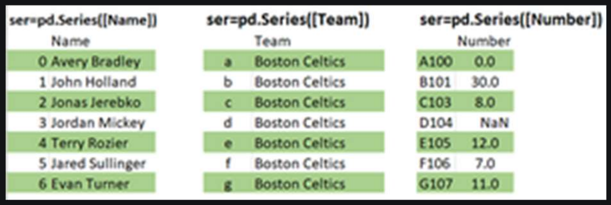
Pandas generally provide two data structures for manipulating data, They are:

- **Series**
- **DataFrame**

- **Series**

Pandas Series is a one-dimensional labeled array capable of holding data of any type (integer, string, float, python objects, etc.). The axis labels are collectively called indexes.

Pandas Series is nothing but a column in an Excel sheet. Labels need not be unique but must be a hashable type. The object supports both integer and label-based indexing and provides a host of methods for performing operations involving the index.



```
ser=pd.Series([Name])       ser=pd.Series([Team])       ser=pd.Series([Number])
      Name                          Team                        Number
   0 Avery Bradley           a     Boston Celtics        A100     0.0
   1 John Holland            b     Boston Celtics        B101    30.0
   2 Jonas Jerebko           c     Boston Celtics        C103     8.0
   3 Jordan Mickey           d     Boston Celtics        D104     NaN
   4 Terry Rozier            e     Boston Celtics        E105    12.0
   5 Jared Sullinger         f     Boston Celtics        F106     7.0
   6 Evan Turner             g     Boston Celtics        G107    11.0
```

Series Data Frame

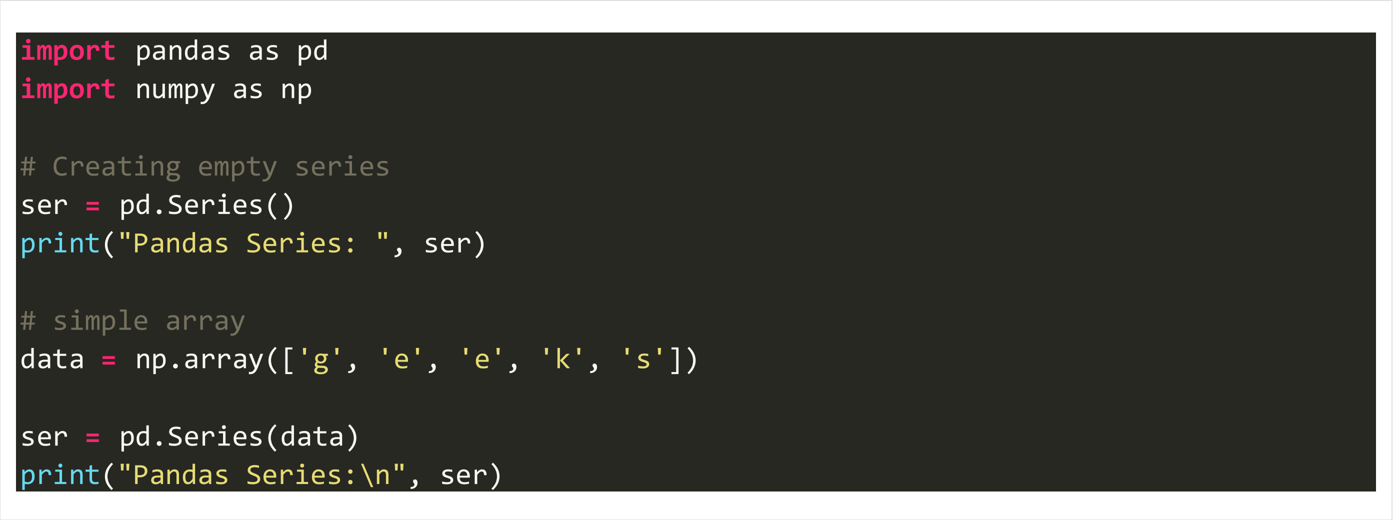
Python | Pandas Series

- **Creating a Series**

In the real world, a Pandas Series will be created by loading the datasets from existing storage, storage can be SQL Database, CSV file, or an Excel file. Pandas Series can be created from lists, dictionaries, and from scalar values, etc.

Example:

- Python3



```
import pandas as pd
import numpy as np

# Creating empty series
ser = pd.Series()
print("Pandas Series: ", ser)

# simple array
data = np.array(['g', 'e', 'e', 'k', 's'])

ser = pd.Series(data)
print("Pandas Series:\n", ser)
```

Output:

Creating a Pandas Series

- **DataFrame**

Pandas DataFrame is a two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns. Pandas DataFrame consists of three principal components, the data, rows, and columns.

5. 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

6.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.