**Project Report**

Data analysis is the process of cleaning, transforming, and modelling data to discover useful information. It helps you identify your business's weak points, leverage strengths, and make data-driven decisions. Data analysis: It can be very effective for companies to understand their customers better, evaluate advertising campaigns, personalise content, create content strategies, and develop products. Businesses that want to increase their performance and profitability should not underestimate data analysis.

It is prevalent to use Python and pythons' libraries for Data analysis and visualisation. This report will examine these libraries; Tkinter, Pandas, and Matplotlib.

**Python Programming Language**

Python programming language is an object-oriented programming language developed by Guido van Rossum in 1990. It is a language frequently used in data analysis, big data, deep learning, machine learning, and it is more straightforward to learn than other languages (Goldwasser & Letscher, 2008). It is one of the most popular software languages today. There are some reasons why it is so popular;

* Open source is freely accessible and available to developers.
* It is object-oriented software.
* The written language is quite plain and straightforward. The syntax is simple. Therefore, it is easy to learn.
* Its usage areas are extensive such as data analysis and data visualisations.
* It has a comprehensive and valuable library infrastructure.
* It can be easily integrated with other software languages (C, C++, Java).

On the other hand, Python has some disadvantages such as (Tulchak & Маrchuk, 2016);

* Python is an interpreted language, so that it may be slower than some other popular languages
* It is a high-level language that is not suitable for writing hardware programs;
* For some tasks, implicit memory allocation can be a disadvantage

1. ***Anaconda***

Anaconda is an integrated python distribution system for developers who want to develop similar scientific applications such as data science, analysis, and machine learning using Python.

It includes software packages such as artificial intelligence, data science, analysis, environment manager and more than 1500 open source packages, and integrated development environment (IDE) such as Spyder, Jupyterlab and Jupyter Notebook.

1. ***Python Libraries***

Python libraries can perform different functions and are used for software development. Python is a multifunctional programming language with a large number of libraries. When it wants to use these libraries, each of which is quite capable, it can call with the necessary code and take action. It is also possible to call and use more than one library simultaneously. Now we will briefly talk about the libraries and functions of Python and the necessary code directories to use the libraries.

**Tkinter**

Tkinter is the standard GUI library for Python. GUI applications can be made when Python is used with Tkinter.

To create a GUI application using Tkinter, all it has to do is follow the steps below.

* Import the Tkinter module.
* Create the GUI application main window.
* Add buttons and more to the GUI application.
* Write the main loop for each user-triggered event.

Example:

from tkinter import \*

window = Tk()

window.title("Test")

window.geometry("400x200")

window.mainloop()

It can add Tkinter Canvas; A canvas is a rectangular area designed for drawing pictures or other complex layouts (Grayson, 2000). It can place graphics, text, widgets or frames on the canvas. Also, Canvas can use with Matplotlib to show charts on GUI. Before use with Matplotlib, It has to import;

from tkinter import \*

from matplotlib.figure import Figure

from matplotlib.backends.backend\_tkagg import FigureCanvasTkAgg,

NavigationToolbar2Tk*).*

**Numpy**

NumPy is the basic package used for scientific calculations in Python. It is a Python library that enables us to perform fast operations on arrays, including multidimensional arrays, various derived objects, and lots of mathematical, logical, shape manipulation, sorting, and selection. Statistics operations and simulations can be done using NumPy (Cakmak, and Cuhadaroglu, 2018).

NumPy arrays are created with a fixed size. It is impossible to change the scope dynamically afterwards, as with the list. If you want to change the size of the array you created, this action deletes the existing array and makes a new one.

NumPy arrays facilitate advanced math and other operations on large numbers of data. Similar functions are more difficult without using the NumPy package.

Example:

import numpy as np

arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])

a = arr.reshape(3,4) # 3 X 4 matrix olusturur.

print(“Shape: “, a.shape) # (3, 4)

print(a)

**Pandas**

After collecting the data, making the data suitable for analysis is very important for data analysis. Data preprocessing and data cleaning is the phase that takes the most time from data scientists. Analyses are made more accessible and faster by organising data with the Pandas library. Pandas is one of Python's most important libraries.

Pandas has methods similar to those in NumPy. While NumPy works with the same data types, Pandas can work differently. A data set written in Excel or a SQL table data can be easily analysed with Pandas. Pandas has been an open-source library since 2010. This library is constantly updated by developers worldwide (Nelli, 2018). The advantage of Pandas;

* It is a library for using fast and efficient dataFrames.
* Switching between files is very easy.
* It can open and examine CSV and text files, and our results are comfortably saved in these file types.
* It makes our job easier for missing data
* We can reshape and use the data more effectively.
* Very helpful in time series data analysis
* Also, most notably speed, pandas is a fast library optimised for speed.

Example:

import pandas as pd # importing pandas from Python

emp= {

“name”: [“matt”, “sue”, “ross”],

“age” : [15, 16, 18],

“share” : [100, 150, 500] }

dataFrame1 = pd.DataFrame(emp)

Pandas has disadvantages, and it is crucial to know them when implementing the project, so here are the disadvantages of utilising Pandas.

* A complex syntax that is not always in line with Python:
* Inadequate documentation for beginners.
* Insufficient 3D matrix compatibility.

Despite having rare disadvantages, Pandas advantages consistently will outrank the cons. Pandas have many essential features that make them popular in the industry. To develop your concepts, you must know them.

Also, there are some points to be reckoned with while loading data. These;

* The file format of your data and the format you want to read in pandas are the same.
* Your data and the python worksheet you opened must be in the same place.

Major reading styles and commands available in pandas

CSV: read\_csv,

JSON: read\_json,

HTML: read\_html,

MS Excel : read\_excel

**Matplotlib**

Visualisation is the most straightforward method of analysing data and drawing precise conclusions. These graphics make it easier to grasp a complex topic. Data visualisations assist us in detecting correlations, structures, and outliers. It also prepares our data for EDA (Exploratory Data Analysis) and Machine Learning techniques.

Python has data visualisation libraries. The Matplotlib library is arguably the most straightforward application phase and the most feature-rich. Matplotlib is a 2D drawing package that assists analysts in visualising data (Hunter, 2007). Matplotlib employs Matlab-like visuals and visualisations; however, Matlab is paid and mixed gives Matplotlib an advantage. Matplotlib's most notable features are:

* The most widely used visualisation library is Matplotlib.
* It has total command over a figure.
* It has the sense of Matlab's graphic sketching, but it is more accessible and freer.

First, we start by installing the matplotlib library from the terminal.

pip install matplotlib

Then we import matplotlib and the Pandas package, which will come in handy later, and we load our dataset into the variable "df." Because of the way most people use it, the maplotlib library is imported using the alias "plt.".

import matplotlib.pyplot as plt

import pandas as pd

df = pd.read\_csv("Put dataset name")

Python Matplotlib: Chart Types

The Python matplotlib library may be used to build a variety of charts. Some of these graphics used in this project:

***Pie Plot***: Using matplotlib, create a pie chart in Python. A Pie Chart is a circular statistical layout that can only show a limited data collection. The chart's area represents the total percentage of the supplied data. The area of the pie slices represents the percentage of data segments.

***Area Plot:*** A stacked area plot magnifies the underlying area plot to represent the growth of the value of numerous groups on the same chart. Each group's values are placed one above the other. Where the cumulative sum is unnecessary, this is the best chart to use to show the distribution of categories as parts of an entire area.

***A bar graph plot*** depicts comparisons between many categories. The chart's one axis represents the individual types being compared, while the other axis indicates a numerical value.

Data visualisation is the practice of analysing and understanding data by visualising it in order to identify patterns, trends, and connections that might otherwise go undetected.

Python has several excellent graphics libraries, each with its own set of capabilities. It has been examined at Tkinter Matplotlib, Pandas visualisation, and Nupy in this post.

# References

Goldwasser, M. H. & Letscher, D., 2008. *Object-Oriented Programming in Python.*Pearson Prentice Hall.

Tulchak, L. V. & Маrchuk, A. O., 2016. *History of Python.*Doctoral dissertation, ВНТУ

Grayson, J.E., 2000. *Python and Tkinter programming*. Manning Publications Co. Greenwich.

Cakmak, U.M. and Cuhadaroglu, M., 2018. *Mastering Numerical Computing with NumPy: Master scientific computing and perform complex operations with ease*. Packt Publishing Ltd.

Nelli, F., 2018. The pandas Library—An Introduction. In *Python Data Analytics* (pp. 87-139). Apress, Berkeley, CA.

Hunter, J.D., 2007. Matplotlib: A 2D graphics environment. *Computing in science & engineering*, *9*(03), pp.90-95.