

PROJECT REPORT

(Project Semester SEPTEMBER 2022 to DECEMBER 2022)

INSTITUTE



SEVEN MENTOR & TRAINING PVT LTD.

Chhatrapati Sivaji nagar, Pune

SUBMITTED BY

MR.ABHISHEK MORE

Under the Guidance of

Prof. Nishesh Gogia

Prof. Nishesh Gogia

(GUIDE)

Prof.Aarti Rahulkar

(HOD)

ACKNOWLEDGEMENT

There are many people we should like to acknowledge for their assistance and support throughout the pursuit of this project work. Unfortunately, many will go unnamed here. For those who are mentioned, please know that your words of encouragement made a difference.

It's indeed our immense pleasure to wish our deep sincere gratitude to Honourable of **Seven mentor pvt ltd.**, for their excellent guidance and continuous encouragement throughout the project work.

We also sincere thanks to **Prof.Aarti Rahulkar**, Head of Department, for his continuous cooperation.

We also express sincere thanks to our guide **Prof.Nishesh gogia**, seven mentor institute for his guidance and constant motivation throughout the course.

Last but not the least, I thank all other Faculty Members of our Department and also the Nonteaching staff and our friends for supporting us.

Name of Student

ABHISHEK MORE

INDEX

➤ PROJECT REPORT.....	1
➤ ACKNOWLEDGEMENT.....	2
➤ PROJECT_NO 1 URL CONVERTER	
➤ INTRODUCTION.....	4
➤ PROBLEM STATEMENT.....	5
➤ LIBRARIES USED.....	6
➤ ENVIRONMENT USED.....	7
➤ FLOW DIAGRAM.....	8
➤ CODE PICTURES.....	9-11
➤ CONCLUSION.....	12
➤ REFERENCES.....	13
➤ PROJECT_NO-2 ZOMATO DATA ANALYSIS	
➤ INTRODUCTION.....	14
➤ PROBLEM STATEMENT.....	15
➤ LIBRARIES USED.....	16-17
➤ CODE PICTURES.....	18-24
➤ CONCLUSION.....	25
➤ REFERENCES.....	26

PROJECT 1 . URL CONVERTER

INTRODUCTION

URL is the abbreviation of Uniform Resource Locator. It is the resource address on the internet. The URL (Uniform Resource Locator) is created by **Tim Berners-Lee** and the Internet Engineering working group in 1994. URL is the character string (address) which is used to access data from the internet. The URL is the type of URI (Uniform Resource Identifier).

The **shorten_url () function** takes the long URL as a parameter and returns a short link used to access the given resource. It first checks if the provided long URL already exists and gets its id. If it's not present in the database, store it and get the id.

When a user clicks on the short URL, the request should be redirected to the URL Shortening Service, which would then redirect it to the actual long URL location. The short domain in the short URL has to be mapped to the service like below.

Given a long URL, the service should generate a shorter and unique alias for it. When the user hits a short link, the service should redirect to the original link. Links will expire after a standard default time span. The system should be highly available.



Fig . 1 URL Converter

PROBLEM STATEMENT

`short_url = type_tiny.tinyurl.short (long_url) print ("The Shortened URL is: " + short_url)`
In the output, we get the **shortened URL** in a form like – “https://tinyurl.com/mbq3m”.
And the TinyURL is because the URL shortener package – Pyshortener uses Tinyurl API by Default.

```
import pyshorteners long_url = input ("Enter the URL to shorten: ") #TinyURL
shortener service type_tiny = pyshorteners.Shortener () short_url =
type_tiny.tinyurl.short (long_url) print ("The Shortened URL is: " + short_url) But we
can change it, and that is what we'll learn further in this tutorial.
```



Fig.2 Problem statement

LIBRARIES USED

Similarly, there are many library functions that operate on strings. These library functions are included in the program by importing the header file <string.h>. Let's have a look at the various library functions that operate on strings.

Python Random module is an in-built module of Python which is used to generate random numbers. These are pseudo-random numbers means these are not truly random. This module can be used to perform random actions such as generating random numbers, print random a value for a list or string, etc.

1. Random –

If you're looking for really random URLs, you might find the random string generator useful. Also try: or just create your own list.

The random library is a collection of functions that all have to do with randomization. This library uses a certain algorithm or equation to add randomness, so in a way, it is not true randomization. However, the library can be useful for small and personal projects. To generate a random whole number, we would use the randint () function.

Random URL Generator | Datarandom Generate random URL web addresses. Use to fill databases or other spaces. Free and easy to use. Create large sets of example URL addresses. Datarandom Numbers Numbers Number (Whole) Number (Fractional) Barcode Text Lorem Ipsum Random Words.

2. String –

URL (String protocol, String host, String file): Creates a URL object from the specified protocol, host, and file name. URL (String protocol, String host, int port, String file): Creates a URL object from protocol, host, port and file name. URL (URL context, String spec): Creates a URL object by parsing the given spec in the given context. Generally, the URL object can be passed to any method instead of a string, as most methods will perform the string conversion, that turns a URL object into a string with full URL. Search Params.

Most Web APIs require you to pass in configuration values via a URL query string. Creating these strings is a matter of reading the API's documentation, and then either doing the mind-numbing work of manually creating the query strings. Or using Python's urllib parsing modules to do it for you.

ENVIRONMENT USED

Jupyter Notebook =

Are you working with Jupyter Notebook and Python? Do you also want to benefit from virtual environments? In this tutorial you will see how to do just that with [Anaconda](#).

In this article, we are going to see how to set Virtual Environment in Jupyter. Sometimes we want to use the Jupyter notebook in a virtual environment so that only selected packages are available in the scope of the notebook. To do this we have to add a new kernel for the virtual environment in the list of kernels available for the Jupyter notebook. Let's see how to do that.

Jupyter Notebook uses the Python kernel developed for jupyter. If we want to add the virtual environment we created on our computer to the Jupyter notebook, we need to do extra operations for this.

In this article, we are going to see how to set Environment in Jupyter. Sometimes we want to use the Jupyter notebook in a environment so that only selected packages are available in the scope of the notebook. To do this we have to add a new kernel for the environment in the list of kernels available for the Jupyter notebook. Let's see how to do that.

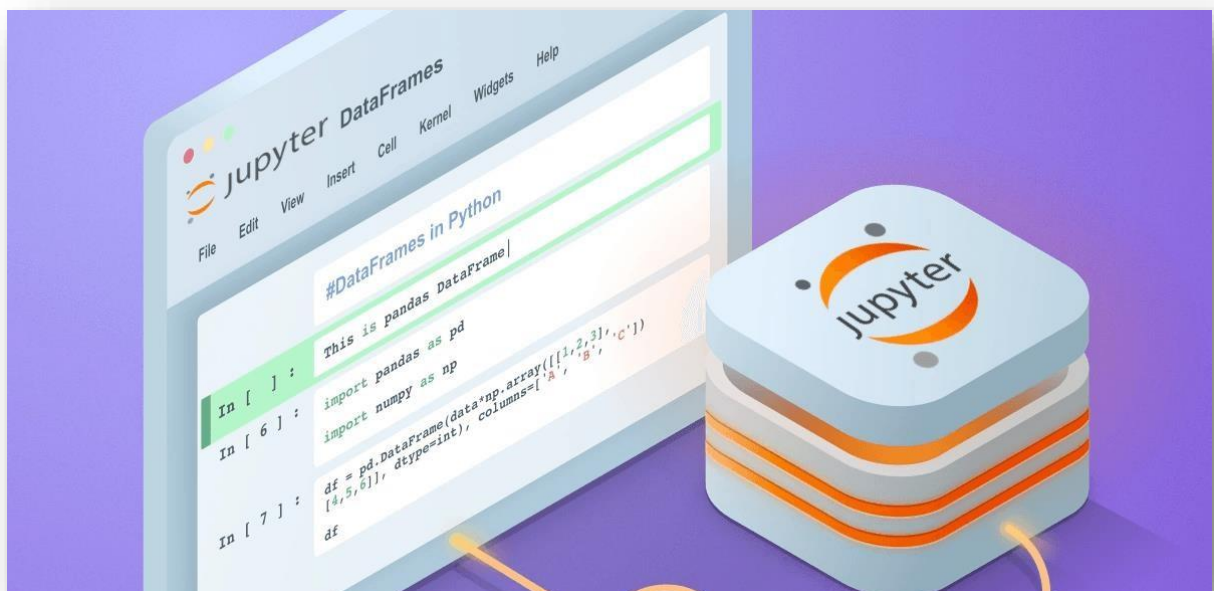


Fig.3 Jupyter Notebook

FLOW DIAGRAM

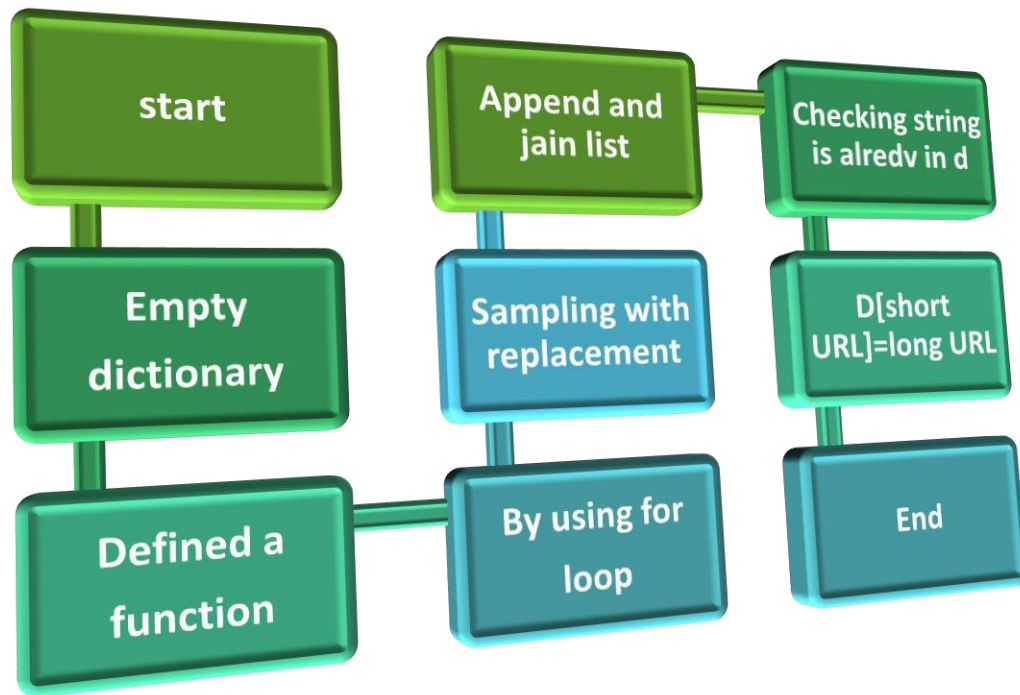


Fig.4 Flow Diagram

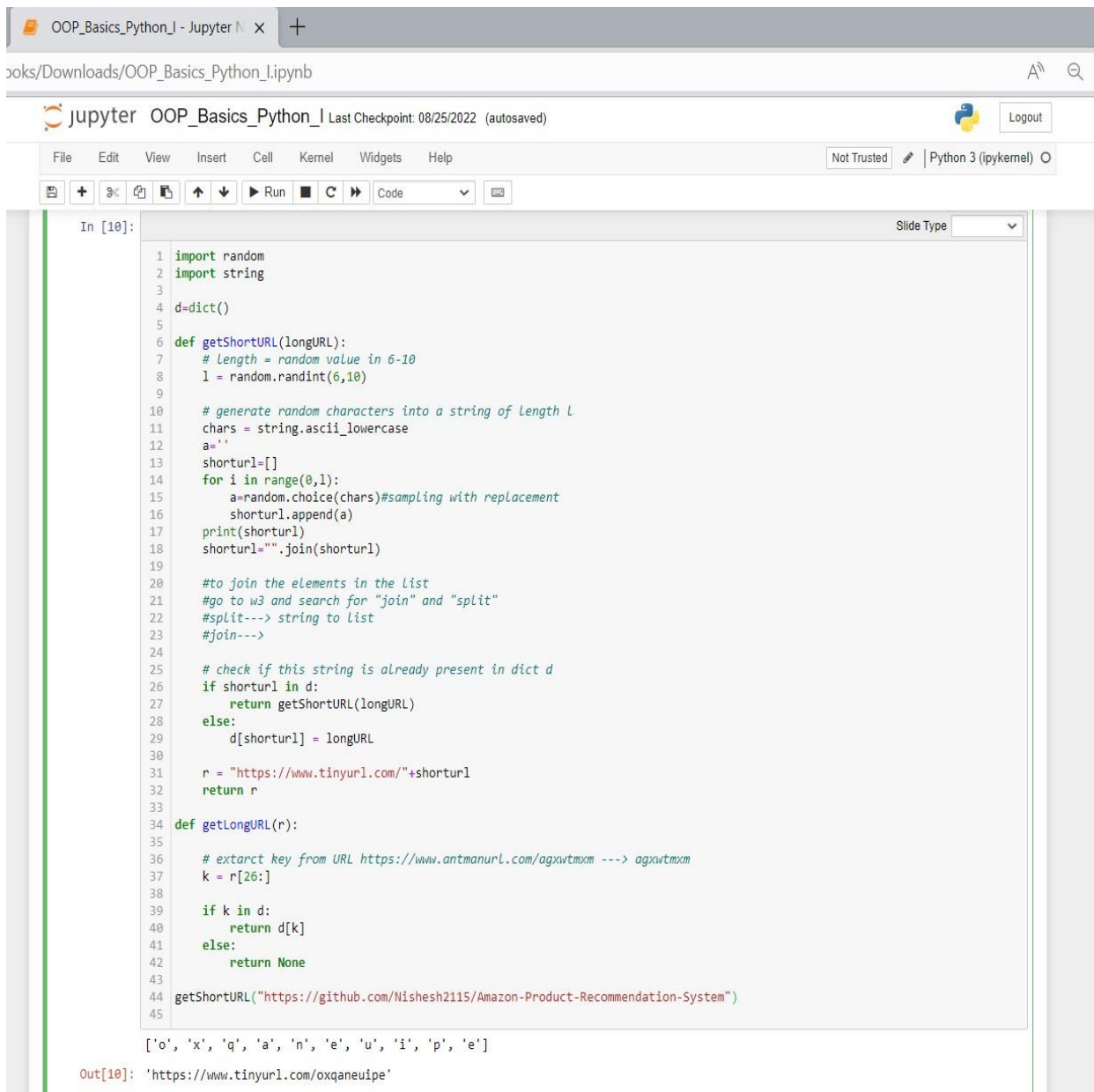
CODE

```
chars=string.ascii_lowercase
```

```
a=random.choice(chars)
```

```
print(a)
```

X



The screenshot shows a Jupyter Notebook window titled "OOP_Basics_Python_I - Jupyter". The address bar shows the file path "Downloads/OOP_Basics_Python_I.ipynb". The notebook interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help), a toolbar with icons for file operations and execution, and a status bar indicating "Not Trusted" and "Python 3 (ipykernel)". The main area displays a code cell with the following Python code:

```
In [10]:
1 import random
2 import string
3
4 d=dict()
5
6 def getShortURL(longURL):
7     # length = random value in 6-10
8     l = random.randint(6,10)
9
10    # generate random characters into a string of length l
11    chars = string.ascii_lowercase
12    a=''
13    shorturl=[]
14    for i in range(0,l):
15        a=random.choice(chars)#sampling with replacement
16        shorturl.append(a)
17    print(shorturl)
18    shorturl="".join(shorturl)
19
20    #to join the elements in the list
21    #go to w3 and search for "join" and "split"
22    #split--> string to list
23    #join-->
24
25    # check if this string is already present in dict d
26    if shorturl in d:
27        return getShortURL(longURL)
28    else:
29        d[shorturl] = longURL
30
31    r = "https://www.tinyurl.com/"+shorturl
32    return r
33
34 def getLongURL(r):
35
36    # extract key from URL https://www.tinyurl.com/agxvtmxm ---> agxvtmxm
37    k = r[26:]
38
39    if k in d:
40        return d[k]
41    else:
42        return None
43
44 getShortURL("https://github.com/Nishesh2115/Amazon-Product-Recommendation-System")
45
```

The output of the code cell is shown below the code:

```
['o', 'x', 'q', 'a', 'n', 'e', 'u', 'i', 'p', 'e']
Out[10]: 'https://www.tinyurl.com/oxqaneuip'
```

Fig.5 Import

```
1 # NON-OOP (a.k.a Procedural) based implementation of a ShortURL System.
2
3 import random
4 import string
5
6 d = dict()
7
8 # given a Long URL, get a short URL
9 def getShortURL(longURL):
10     # Length = random value in 6-10
11     l = random.randint(6,10);
12
13
14     # generate random characters into a string of length l
15     chars = string.ascii_lowercase
16     a=''
17     shorturl=[]
18     for i in range(0,l):
19         a=random.choice(chars)#sampling with replacement
20         shorturl.append(a)
21     shorturl="".join(shorturl)
22
23     #to join the elements in the list
24     #go to w3 and search for "join" and "split"
25     #split--> string to list
26     #join-->
27
28     # check if this string is already present in dict d
29     if shortURL in d:
30         return getShortURL(longURL);
31     else:
32         d[shortURL] = longURL;
33
34
35     r = "https://www.shortURL.com/"+shortURL
36     return r;
37
38 def getLongURL(shortURL):
39
40     # extract key from URL https://www.MERAUURL.com/mxzmuis --> mxzmuis
41     k = shortURL[25:]
42
43
44     if k in d:
45         return d[k]
46     else:
47         return None
48
49 ..
```

Fig.6 NON-OOP

```
1 # Class variables shared by all objects
2
3 # Class: group all variables/attributes and functions/methods into a single logical unit
4 import random
5 import string
6 class ShortURL1:
7
8     URLPrefix = "https://www.shortURL.com/"; # class variable shared by all objects
9
10    def __init__(self): # constructor; not must, but, good to have; initialize all attributes here
11        self.d=dict();
12
13    # given a Long URL, get a short URL
14    def getShortURL(self, longURL): # first argument to all methods is "self" => this object
15        # length = random value in 6-10
16        l = random.randint(6,10);
17
18        # generate random characters into a string of length l
19        chars = string.ascii_lowercase
20        shortURL = ''.join(random.choice(chars) for i in range(l))
21
22        # check if this string is already present in dict d
23        if shortURL in self.d:
24            return getShortURL(longURL);
25        else:
26            self.d[shortURL] = longURL;
27
28        r = self.URLPrefix + shortURL
29        return r;
30
31    def getLongURL(self, shortURL):
32
33        # print(self.d); # print statement for debugging
34
35        # extract key from URL https://www.shortURL.com/mxzmuis ---> mxzmuis
36        k = shortURL[25:];
37
38        if k in self.d:
39            return self.d[k];
40        else:
41            return None;
```

Fig.7 class Variables

CONCLUSION

In this paper, various phases of data analysis including data collection, cleaning and analysis are discussed briefly. Explorative data analysis is mainly studied here. For the implementation, Python programming language is used. For detailed research, jupyter notebook is used. Different Python libraries and packages are introduced. Using various analysis and visulaization methods, numerous results are extracted. The dataset “World Happiness Record 2022” is used and extract important informations like the difference in the score of happiness of different countries, the dependence of one attribute in building up the score, how a variable affects another variable, etc. are seen in this analysis and various graphs has been plotted using various attributes in the dataset and draw conclusions in an easy way.



Fig.8 Conclusion

REFERENCES

<https://www.kaggle.com>

<https://tinyurl.com>

- <https://en.wikipedia.org/wiki/TinyURL>
- <https://tiny.cc>
- <https://www.geeksforgeeks.org/how-to-design-a-tiny-url-or-url-shortener>
- <https://bitly.com>



Fig.9 Social Media

PROJECT NO-2 ZOMATO DATA ANALYSIS

INTRODUCTION

Data are those raw facts and figures with no proper information hence need to be processed to get the desired information. While information is those results which we get after processing the raw data in different levels or extracted conclusions from a given dataset through a process called data analysis.

Data Analysis is simply the analysis of various data means cleaning the data, transforming it into understandable form, and then modeling data to extract some useful information for business use or an organizational use. It is mainly used in taking business decisions. Many libraries are available for doing the analysis. For example, NumPy, Pandas, Seaborn, Matplotlib, Sklearn, etc.

- NumPy: NumPy is a library written in Python, used for numerical analysis in Python. It stores the data in the form of nd-arrays (n-dimensional arrays).
- Pandas: Pandas is mainly used for converting data into tabular form and hence, makes the data more structured and easily to read.
- Matplotlib: Matplotlib is a data visualisation and graphical plotting package for Python and its numerical extension NumPy that runs on all platforms.
- Seaborn: Seaborn is a Python data visualisation package based on matplotlib that is tightly connected with pandas data structures. The core component of Seaborn is visualisation, which aids in data exploration and comprehension



Fig.1 Zomato

PROBLEM STATEMENT

Problem Statement : In this challenge, we are analysing the Zomato Restaurant dataset to find the more insights about the Restaurant business.

The basic idea of analysing the Zomato dataset is to get a fair idea about the factors affecting the aggregate rating of each restaurant, establishment of different types of restaurant at different places, pune being one such city has more than 12,000 restaurants with restaurants serving dishes from all over the world. With each day new restaurants opening the industry hasn't been saturated yet and the demand is increasing day by day. In spite of increasing demand it however has become difficult for new restaurants to compete with established restaurants. Most of them serving the same food. Pune being an IT capital of India. Most of the people here are dependent mainly on the restaurant food as they don't have time to cook for themselves. With such an overwhelming demand of restaurants it has therefore become important to study the demography of a location.



Fig.2 Problem

LIBRARIES USED

Data Analysis of Zomato using pandas, NumPy, seaborn, Matplotlib library.

1. NumPy =

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

2.Pandas =

Pandas is an open-source library 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. Pandas is fast and it has high performance & productivity for users.

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 1.4.1

3.Seaborn =

Is a library mostly used for statistical plotting in Python. It is built on top of Matplotlib and provides beautiful default styles and color palettes to make statistical plots more attractive.

In this tutorial, we will learn about Python Seaborn from basics to advance using a huge dataset of seaborn basics, concepts, and different graphs that can be plotted.

After the installation let us see an example of a simple plot using Seaborn. We will be plotting a simple line plot using the iris dataset. Iris dataset contains five columns such as Petal Length, Petal Width, Sepal Length, Sepal Width and Species Type. Iris is a flowering plant, the researchers have measured various features of the different iris flowers and recorded them digitally.

4. Matplotlib =

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hard copy formats and interactive environments across platforms. It can be used in Python scripts, The Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface tool kits. It tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatterplots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.



Fig.3 Top Python Machine Learning Libraries

CODE

Importing libraries:

Libraries that would be used in the process of analysis are to be imported first. Here are the codes to import the libraries. import pandas as pd import numpy as np import matplotlib.pyplot as plt import seaborn.

```
In [1]: 1 #importing libraries
        2 import numpy as np
        3 import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
        4 import seaborn as sns
        5 import matplotlib.pyplot as plt
        6
        7
        8
        9
```

```
host:8888/notebooks/Project/zomato_data_analysis%20Abhi%20More%20.ipynb

jupyter zomato_data_analysis Abhi More Last Checkpoint: 09/12/2022 (autosaved)
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

3 warnings.filterwarnings('ignore')

In [3]: 1 #Loading the csv file
        2 df= pd.read_csv("zomato_outlet_final (11).csv")
        3 df.head()

Out[3]:
```

	link	rest_name	rest_type	loc	dine_rating	dine_reviews	delivery_rating	delivery_reviews	cuisi
0	https://www.zomato.com/pune/mix@36-the-westin-...	Mix@36 - The Westin	Lounge	The Westin, Mundhwa	4.2	(707 Reviews)	NaN	NaN	Indian, Continental, Asian, Fing
1	https://www.zomato.com/pune/kangan-the-westin-...	Kangan - The Westin	Fine Dining	The Westin, Mundhwa	4.1	(344 Reviews)	NaN	NaN	North Indian, Mughl
2	https://www.zomato.com/pune/coriander-kitchen-...	Coriander Kitchen - Conrad Pune	Fine Dining	Conrad Pune, Bund Garden Road	4.5	(541 Reviews)	NaN	NaN	Mediterranean, Asian, Noi
3	https://www.zomato.com/pune/the-market-the-westin-...	The Market - The Westin	Fine Dining	The Westin, Mundhwa	4.0	(55 Reviews)	NaN	NaN	Asian, European, North Ind
4	https://www.zomato.com/pune/vandaag-vivanta-pu-...	Vandaag - Vivanta Pune	Pub, Casual Dining	Vivanta Pune, Hinjewadi	3.9	(113 Reviews)	NaN	NaN	Finger Food, Continent

```
In [4]: 1 df.columns
```

Fig.4 Importing libraries

```
In [7]: 1 df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5433 entries, 0 to 5432
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype  
---  --
0   rest_name              5433 non-null  object 
1   rest_type              4336 non-null  object 
2   loc                    5433 non-null  object 
3   dine_rating            5433 non-null  float64
4   dine_reviews           4829 non-null  object 
5   delivery_rating        2539 non-null  float64
6   delivery_reviews       2539 non-null  object 
7   cuisine                5433 non-null  object 
8   Cost (RS)              5433 non-null  int64  
9   cost                   5433 non-null  object 
10  liked                  2603 non-null  object 
dtypes: float64(2), int64(1), object(8)
memory usage: 467.0+ KB

We have some null values in some columns, we will handle them step by step
```

Fig.5 Some null values in some columns

Data types:

Datatype refers to the type of data- int, object, float are the basic datatypes in python. Printing the types of data of all the columns in the dataset using dtypes df.data. dtypes.

OBSERVATIONS

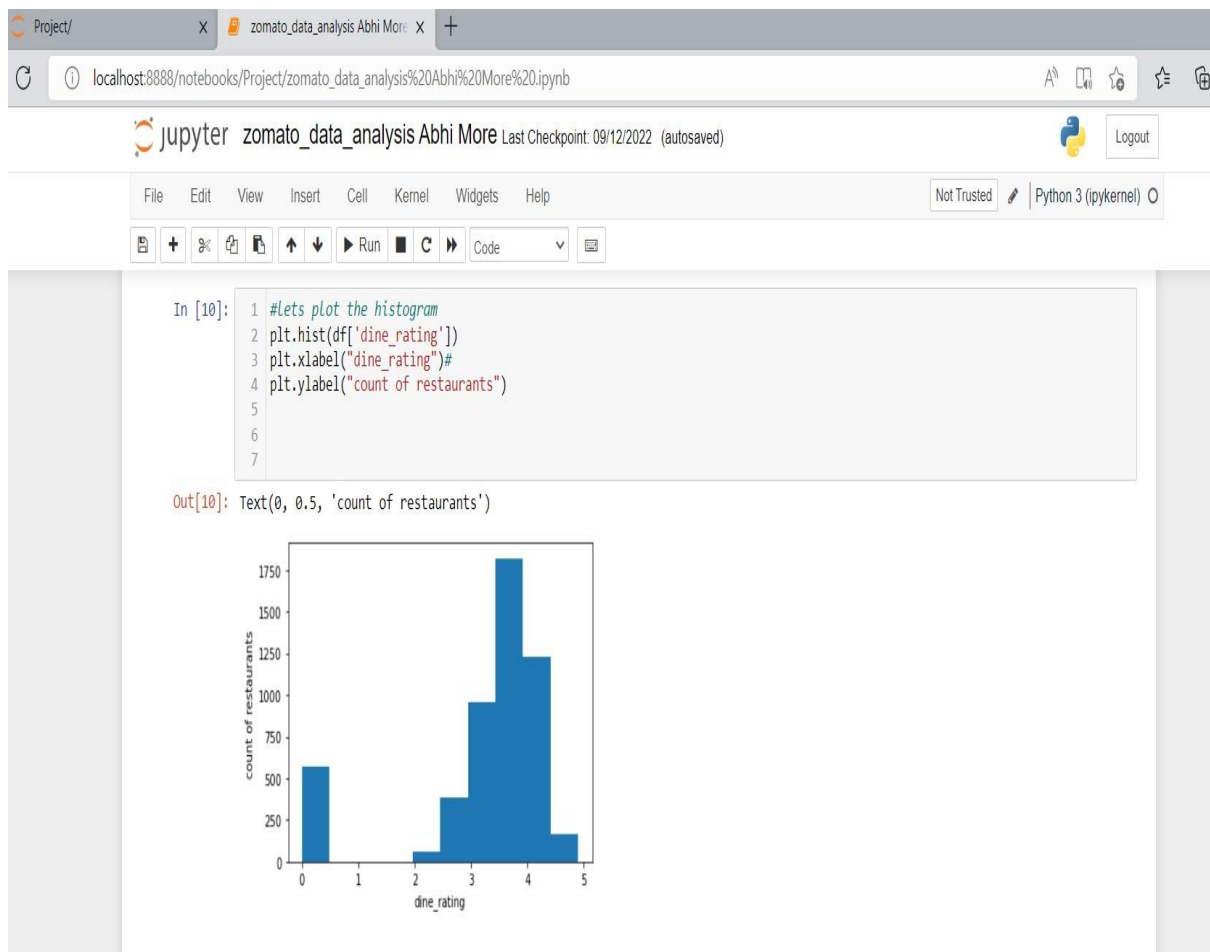


Fig.6 Histogram

Exploratory Data Analysis

Lets do the EDA on Zomato to understand the data in depth.

We gonna do Univariate Data Analysis and Bivariate Data Analysis.

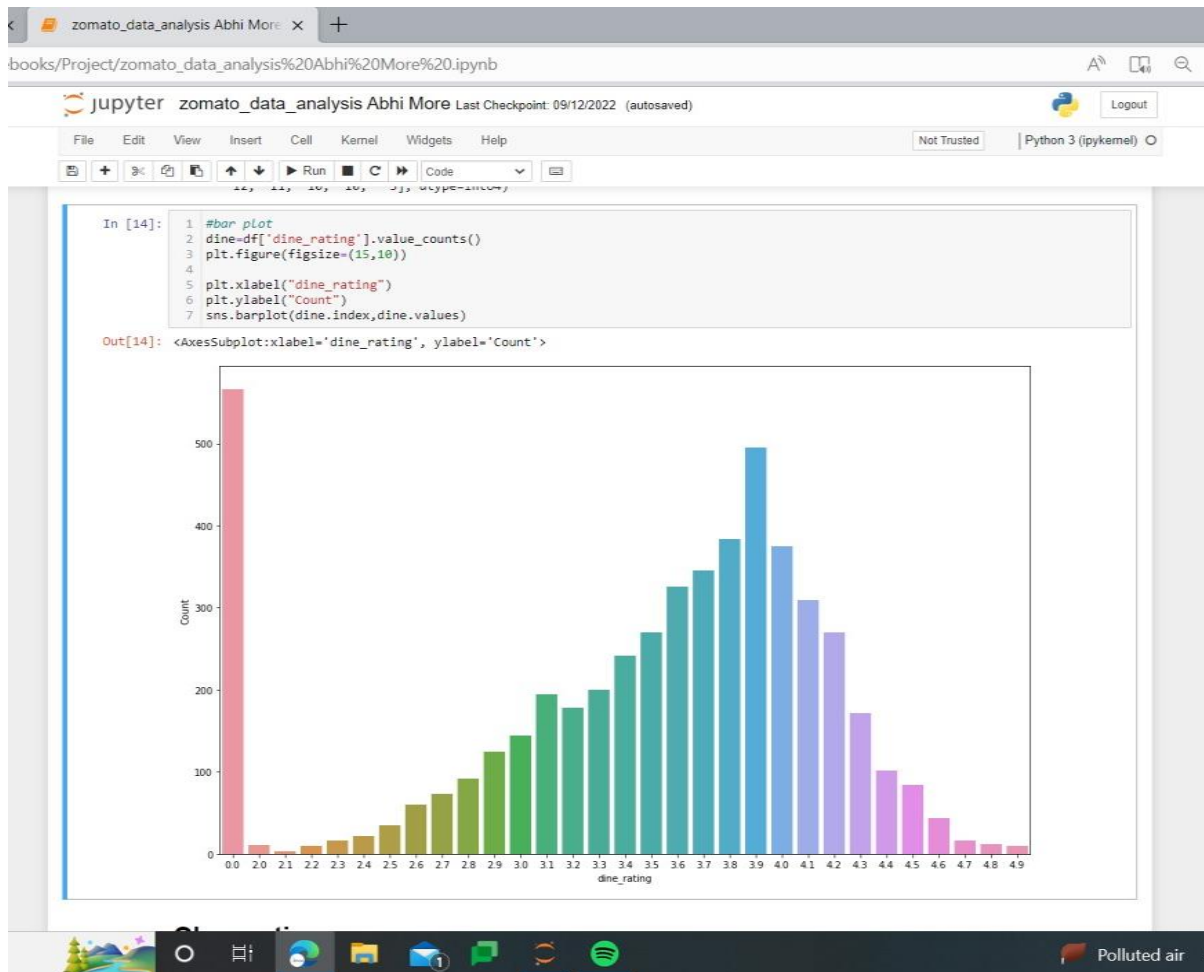


Fig.7 bar plot

Observation

Most of the customers have ordered 3.9 rating.

There are lots of restaurants which got 0 ratings .

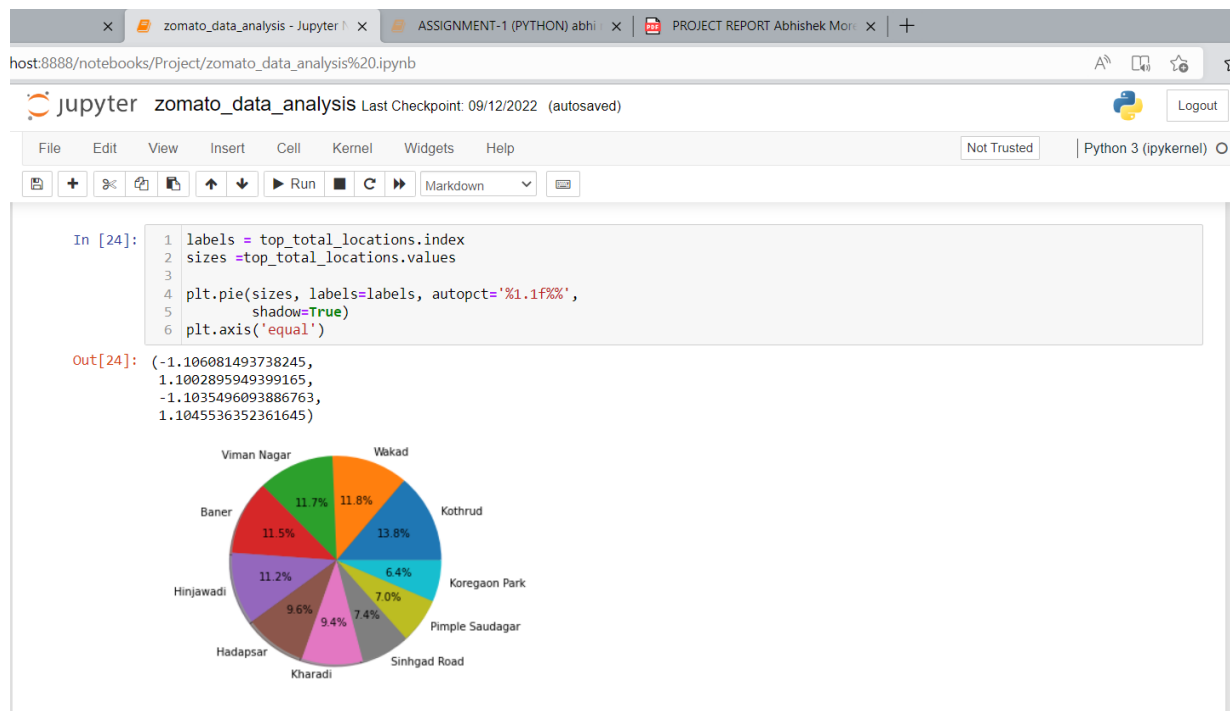


Fig.8 Pie Chart

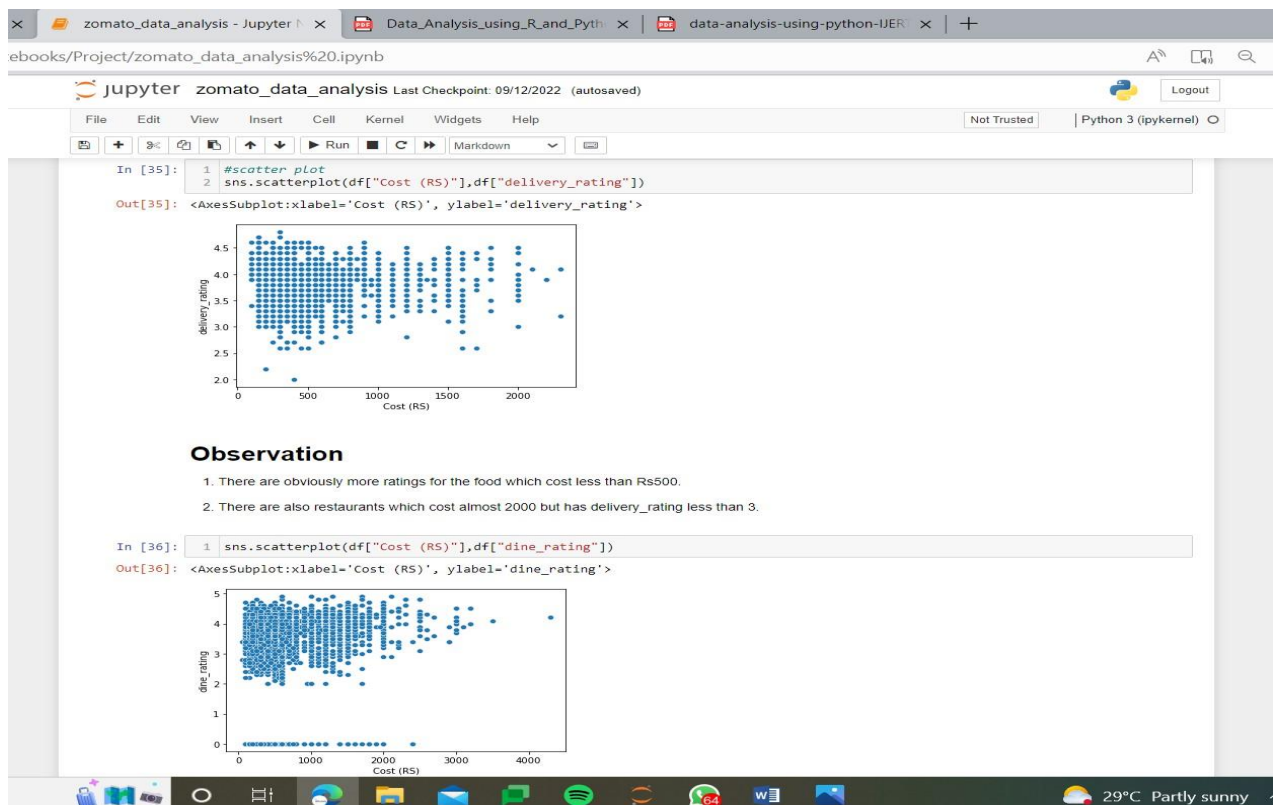
Observation

1. Kothrud covers 13.8% of top 10 locations in pune.
2. Viman Nagar and Baner covers almost 11.5% of the top 10 locations in pune.



Observation

1. Most of the restaurant gets 3.9 delivery rating.
2. There are very few restaurants who get less than 2.5 rating.



1. Obviously there are more ratings for the food which cost less than 1000.
2. There are restaurants which does have rating irrespective of cost.

Fig.9 Scatter plot

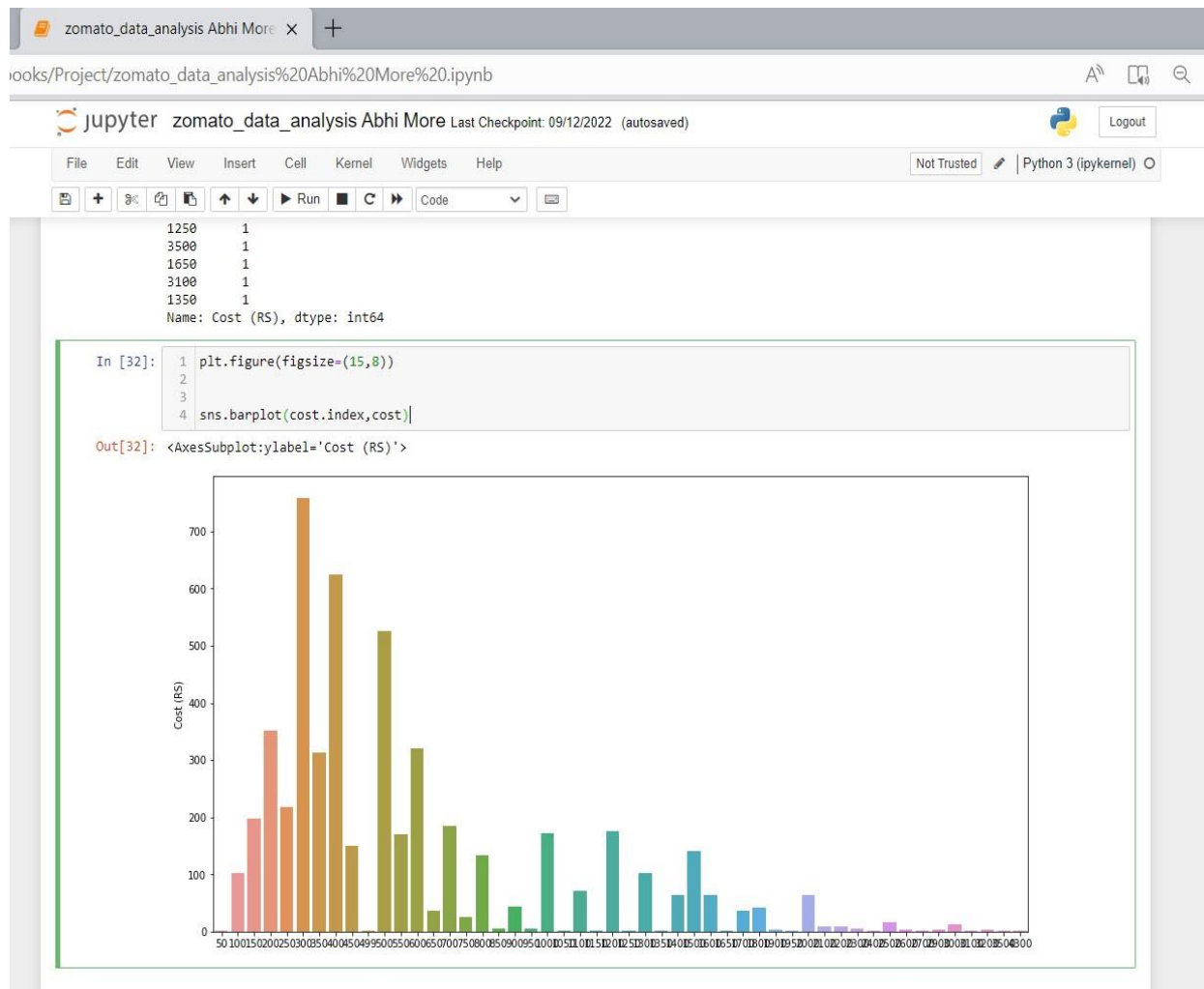


Fig.10 Sns.Barplot

Observation

1. There are 759 restaurants which that cost Rs300 for 2 people.
2. There are 624 restaurants which that cost Rs400 for 2 people.

CONCLUSION

From this data processing, we can get this following conclusion: There are 105624 restaurants registered on Zomato Apps based in India. Almost 18% of registered restaurants in India are located at Pune (5000 restaurants), 13.8% Kothrud (264 restaurants), 11.8% in Wakad (225 restaurants), 11.7% VimanNagar (224 restaurants), 11.5% Baner (219 restaurants), 11.2% Hinjawadi (213 restaurants), 9.6% Hadapsar (183 restaurants), 9.4% Kharadi (180 restaurants). 7.4% Sinhgad Road (142 restaurants), 7.0% Pimple Saudagar (134 restaurants), 6.4% Koregaon Park (123 restaurants).

- Kothrud covers 13.8% of top 10 locations in pune.
- Viman Nagar and Baner covers almost 11.5% of the top 10 locations in pune
- Kothrud has the maximum number of restaurants followed by wakad and viman nagar.



Fig.11 Conclusion

REFERENCES

<http://www.edupristine.com/blog/zomato-acquires-food-portal-urbanspoon>

<http://techstory.in/zomato-story-the-facts-and-the-future%20/> <http://cik.it>

[learnpython.org/](https://www.learnpython.org/) <https://www.analyticavidhya.com> <https://pythonprogramming.net/>

<https://courses.edx.org> <https://www.youtube.com/>

<https://www.newagepublishers.com/samplechapter 001938.pdf>

<http://mathfaculty.fullerton.edu/mathews/n2003/LogisticEquationMod.html>

<https://www.ima.umn.edu/-tmlai/M119-Section47.pdf>

<http://www.public.iastate.edu/~swillson/Logistic.pdf> <https://stackoverflow.com>

<https://google.co.i>

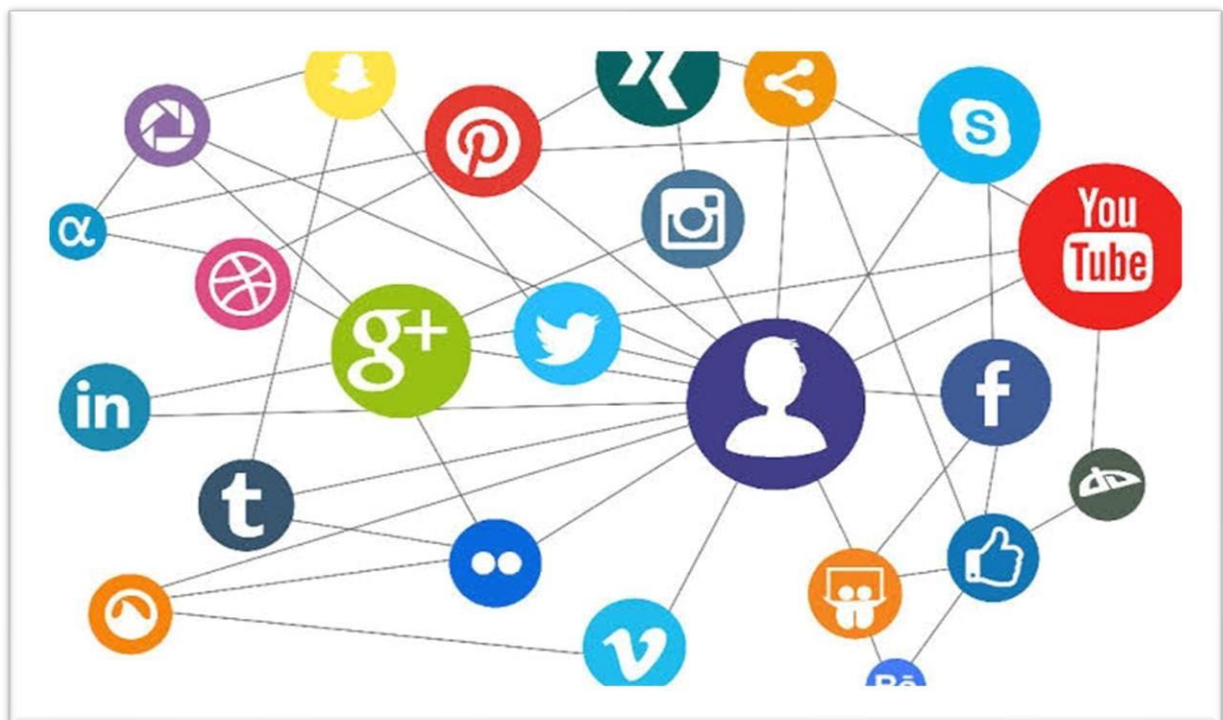


Fig.12 Social Media