A
SUMMER INTERNSHIP REPORT
On

DATA ANALYTICS AND MACHINE LEARNING

Submitted

by

Bokil Neel Milind

(201430116080)

Guided By:
Ms. Roshni Mandli

INFORMATION TECHNOLOGY DEPARTMENT

A Summer Internship Submitted to
Gujarat Technological University In fulfillment for the award of degree
of Bachelor of Engineering in Information Technology

ACADEMIC YEAR – 2023-2024

NEW L. J. INSTITUTE OF ENGINEERING AND TECHNOLOGY

Pakwan, Behind Rajpath Club Gate to Sindhu Bhavan Road,
Sarkhej - Gandhinagar Highway,
Ahmedabad, Gujarat 380054

CERTIFICATE

This is to certify that the Internship report submitted along with the project entitled Internship in **INFOLABZ IT SERVICES PVT. LTD.** has been Completed by **BOKIL NEEL MILIND** under my guidance in complete fulfilment for the Bachelor of Engineering in Information Technology Branch, 7th Semester of Gujarat Technological University, Ahmedabad during the academic year 2023-24

Date:

Place: NEW LJIET, Ahmedabad.

Signature and Name of Guide

Prof. Roshni Mandli

Assignment Professor, (IT),

IT Department,

NEW LJIET (143), Ahmedabad.

Signature and Name of H.O.D.

Dr. Gayatri S. Pandi

HOD, (CSE(AIML) & IT),

NEW LJIET (143), Ahmedabad.

Seal of Institute

DECLARATION

We hereby declare that the Internship report submitted along with the Internship entitled Internship in **INFOLABZ IT SERVICES PVT. LTD.** submitted in Complete for Bachelor of Engineering in Information Technology Branch to Gujarat Technological University, Ahmedabad, is a bonafide record of original Internship work Completed by me at **INFOLABZ IT SERVICES PVT. LTD.** under the supervision of External Guide **Mr. CHINTAN NAGRECHA** and that no part of this report has been directly copied from any students' reports or taken from any other source, without providing due reference.

Name of Student **Signature of Student**

BOKIL NEEL MILIND

ACKNOWLEDGEMENT

I wish to express my sincere gratitude to my External guide **Mr. Chintan Nagrecha** for continuously guiding me at the company and answering all my doubts with patience. I would also like to thank **Dr. Gayatri S. Pandi** (H.O.D. of IT Department) for motivating me every time whenever I get confused, I would also like to thank my Internal Guide **Ms. Roshni Mandli** for helping me through my internship by giving me the necessary suggestions and advices along with their valuable co-ordination in completing this Internship.

I also thank my parents, friends and all the members of the family for their precious support and encouragement which they had provided in completion of my work. In addition to that, I would also like to mention the company personals who gave me the permission to use and experience the valuable resources required for the Internship.

Thus, in conclusion to the above said, I once again thank the staff members of **INFOLABZ IT SERVICES PVT. LTD.** for their valuable support in completion of the Internship.

Thank You

Data Analytics and Machine Learning

Enrollment No.: 201430116080

Student Name: Bokil Neel Milind

NEW L. J. Institute of Engineering and Technology (143)

Semester: VII, Information Technology Department

Abstract

The Industrial training in 7th Semester has been a productive and imposing experience for me. This training experience will assist and usher me in my future endeavors. It would also help me to build a career in the field of data science and much more. This training helped me gain more knowledge and skills in the uprising fields of Computer Engineering and Information Technology. This experience didn't just honed my skills but it also helped me pave a new path leading to a brighter and consistent future. The training took place at INFOLABZ, a well renowned entity whose main objective is to sustain the exponential growth in the IT industry.

The principal area of interest in this practice was the widely used, versatile, high level and dynamic programming language – Python. Python is known for its simplicity and readability which helps the user to understand the code and its functionalities in a very easy manner. It is well-liked by the beginners and experts alike due to its lucid syntax and extensive libraries. It is suitable for web development, data analysis, artificial intelligence, and much more. Overall, Python's balance of simplicity, versatility and a strong community makes it a powerful tool for various programming needs. Its popularity has led to its use in diverse fields as mentioned above. For data analysis, Python offers tools like pandas for data manipulation, NumPy for numerical computations and Matplotlib/Seaborn for visualization. In Machine Learning, Python serves as a notable language due to its rich ecosystem of libraries like scikit-learn, TensorFlow and PyTorch. This training hence provided us the understanding of how modern technology can be utilized for professional lifestyle.

TABLE OF CONTENTS

CHAPTER: 1

INTRODUCTION

1.1 Introduction

Infolabz IT Services Pvt. Ltd. is a leading IT company that provides technical solutions and services.

INFOLABZ IT SERVICES PVT LTD

Contact Number: 8866662662

Email: info@infolabz.in

Website: www.infolabz.in

Address:

405, VRAJ AVENUE, ABOVE SAM'S PIZZA, NR

COMMECE SIX ROAD, NAVRANGPURA,

AHMEDABAD, 38000

1.2 Company Profile

Established in 2016, incorporation with our parent IT company, INFOLABZ IT SERVICES PVT. LTD. has managed to make its own position in IT Sector. We are involved in Web Development, App Development, Progressive Web Application Development, IOT solutions, Graphics & Designing, Digital Marketing, Domain & Hosting services, SMS services etc.

In the span of seven years we have managed to deliver all projects on time with utmost accuracy to our clients across the globe. We have dedicated teams of experienced and hardworking developers.

1.3 Company Products

Custom Software Development: Using company's extensive experience, we create customized solutions that propel your company's success to new heights.

Web Development: You need skilled web developers for robust web solutions. You can build a successful website with the assistance of our team of expert resources.

Mobile Apps Development: They keep up with cutting-edge technologies in this fast-paced era to increase productivity and develop more reliable mobile applications.

IOT Solution Providers: From small businesses to large corporations, the IoT is all around us. We use IoT to make your business stronger and more successful.

UI/UX Designs: With a perfect blend of professionalism and stunning creativity, our expert designers can establish a bright profile for your business in the digital world.

Data Science and ML Development: Discover new business opportunities and solve real-time problems by using data science and machine learning.

1.4 Company Mission and Vision

Our Mission is to deliver best-in-class services with top-notch quality every quarter. Our Vision is to sustain the exponential growth of the IT industry and to develop a product with one of its kind concepts which could be used by millions of people.

CHAPTER: 2

Role and Responsibilities During Internship

2.1 Objective

To acquire new skills, knowledge and practical experience in the field of data analysis and machine learning.

To learn to manage time effectively, meet deadlines and prioritize tasks in a professional environment.

To receive feedback from the supervisor to improve performance and skills.

To explore the Data analysis and Machine learning fields in order to get an insight about the strengths and interests regarding those field.

To gain real-world, hands-on experience related to Data analysis and Machine Learning.

To gain a better understanding of the company's culture, values, operations and goals.

2.2Introduction

Data analytics is the process of deriving insights about the data and helps us to make decisions based on that data. It makes use of various techniques to collect, clean, and transform data into a format that can be analyzed. Various statistical and computational techniques are then applied to discover patterns, trends, correlations, and anomalies within the data.

Data analytics is widely used across industries to optimize operations and improve strategies.

Machine Learning (ML) is a branch of AI, which enables the machine i.e. the system to learn and improve from experience. It is widely used to create algorithms which can be used to analyze and identify patterns in data which then can be used to make predictions and decisions without the need of the human interference. ML is trained on old data and is used to make predictions on the new data.

This technology is utilized in many applications such as healthcare diagnosis, image recognition, autonomous vehicles and much more.

2.3 Role and Responsibilities During Internship

To collect data and clean the data obtained from various sources.

To perform exploratory data analysis to uncover insights and understand the data in a better manner.

Create data visualization in order to represent the data using various charts, graphs and other visual elements so that the data can be understood in a more proper way.

Applying basic statistical analysis to draw necessary conclusions related to the data.

Prepare and preprocess the data for the machine learning models.

Develop and examine the data with machine learning algorithms.

Train the model, optimize the model and test whether the model is making the accurate predictions or not. Check the accuracy of the model.

Collaborate and communicate with the supervisor, mentor and colleagues during the internship to improve your overall performance.

CHAPTER: 3

Daily Task

3.1 Daily Task

Days	Date	Activities
Day 1	27/07/23	• Introduction about the Internship and Data Science • Basics of Data Analytics • Types of Data • Understanding Data Cleaning and Preprocessing • Introduction to Dictionary • Concept of API and Requests Package
Day 2	28/07/23	• API Handling • API Search
Day 3	31/08/23	• Data Visualization using Matplotlib library
Day 4	01/08/23	• Introduction to Data Visualization • API Data Visualization • Graphs plotted on API Data
Day 5	02/08/23	• Assignment -1 • News API • ISRO Spacecrafts API • ISRO Customers API • Dynamic API – PINCODE API

Day 6	03/08/23	• Introduction to Pandas library • Introduction to NumPy library • Pandas: Data Frames • Pandas: Excel • Pandas: API to CSV • XLRD: IPL Data Analysis
Day 7	04/08/23	• Introduction to Machine Learning • Introduction to Linear Regression • Linear Model: Mathematics • Linear Model: Implementation
Day 8	07/08/23	• Multiple Linear Regression • Multiple Linear Model: Mathematics • Multiple Linear Model: Implementation
Day 9	08/08/23	• Polynomial Linear Regression • Polynomial Linear Regression model: Implementation • Image Data
Day 10	09/08/23	• Assignment – 2 • OpenCV • Convolutional Neural Network Project
Day 11	10/08/23	• Conclusion and words of appreciation from the supervisor

CHAPTER: 4

Internship Work

4.1 Internship Work

Days	Date	Activities
Day 1	27/07/23	• Introduction about the Internship and Data Science • Basics of Data Analytics • Types of Data • Understanding Data Cleaning and preprocessing • Introduction to Dictionary • Concept of API and Requests Package

- **Introduction about the Internship and Data Science:**
  - A brief introduction about the internship and the work we will do in this program was discussed. The supervisor explained us about the company and the work they do there.
  - Data Science is the interdisciplinary field that involves extracting knowledge and insights from data using scientific methods, algorithms, processes, and systems. Data Science aims to uncover patterns, make predictions, and inform decision-making through data-driven approaches.

- **Basics of Data Analytics:**
    - Data Analytics is the process of examining, cleaning, transforming, and modeling data to discover useful information, draw conclusions, and support decision-making. It involves applying statistical and computational techniques to large and diverse data sets, aiming to uncover patterns, trends, correlations, and insights.

- **Concept of API and Requests Package:**
    - An API, or Application Programming Interface, is a set of rules and protocols that allows different software applications to communicate and interact with each other. In short, an API acts as a bridge that enables applications to request and exchange information or perform actions with other software, making it easier to create powerful and integrated software systems.
    - The ‘requests’ package in Python is a popular library used for making HTTP requests to web servers and interacting with APIs. It simplifies the process of sending requests, handling responses, and managing various aspects of web communication.
    - With the ‘requests’ package, you can easily retrieve data from web resources and perform actions like sending data, headers, and parameters. In short, the ‘requests’ package allows you to seamlessly integrate web-based functionalities into your Python applications by providing a user-friendly interface for working with HTTP requests and responses.

Days	Date	Activities
Day 2	28/07/23	• API Handling • API Search

- **API Handling:**
  - In order to understand the APIs and how their data is analyzed, we will study few different APIs and work on them accordingly.

COVID API:

- COVID API link: https://data.covid19india.org/data.json
- Import the API using requests package

Fig 2.1

- We use '.get()' method here to fetch the data from the API and '.json()' method is used because the API's URL where the data is stored uses json.
- Studying the COVID API, its content and its properties:

```
In [3]: print(type(covid_data))
        <class 'dict'>

In [4]: for i in covid_data:
            print(i)
        cases_time_series
        statewise
        tested

In [5]: print(covid_data["cases_time_series"][78]["date"])
        17 April 2020
```

Fig 2.2

- When the data is imported from the API, it is represented as a dictionary.
- The 'for' loop is used to print all the main keys of the COVID API.
- Here, 'covid_data["cases_time_series"][78]["date"]' returns the date at index 78.
- Now as given in the Fig 1.2.3 we print all the data in the COVID API using the 'for' loop. We use the keys in the specific manner which helps us print all the data of our API in a specific order and manner.

```
In [2]: #print all data
        for i in range(0, len(covid_data["cases_time_series"])):
            print("Date : ", covid_data["cases_time_series"][i]["date"],
                  "Cases : ", covid_data["cases_time_series"][i]["dailyconfirmed"],
                  "Deaths : ", covid_data["cases_time_series"][i]["dailydeceased"])

Date :  30 January 2020 Cases :  1 Deaths :  0
Date :  31 January 2020 Cases :  0 Deaths :  0
Date :  1 February 2020 Cases :  0 Deaths :  0
Date :  2 February 2020 Cases :  1 Deaths :  0
Date :  3 February 2020 Cases :  1 Deaths :  0
Date :  4 February 2020 Cases :  0 Deaths :  0
Date :  5 February 2020 Cases :  0 Deaths :  0
Date :  6 February 2020 Cases :  0 Deaths :  0
Date :  7 February 2020 Cases :  0 Deaths :  0
Date :  8 February 2020 Cases :  0 Deaths :  0
Date :  9 February 2020 Cases :  0 Deaths :  0
Date :  10 February 2020 Cases :  0 Deaths :  0
Date :  11 February 2020 Cases :  0 Deaths :  0
Date :  12 February 2020 Cases :  0 Deaths :  0
Date :  13 February 2020 Cases :  0 Deaths :  0
Date :  14 February 2020 Cases :  0 Deaths :  0
Date :  15 February 2020 Cases :  0 Deaths :  0
Date :  16 February 2020 Cases :  0 Deaths :  0
Date :  17 February 2020 Cases :  0 Deaths :  0
```

Fig 2.3

- **API Search:**
  - As we saw earlier, we now know how to handle the API data, but there is more to it. We can also learn of how to fetch the useful information from the API in more detail.
  - For **COVID API**, as we can see in the image, we print the data of only those days which are having cases recorded more than 100000.

```
In [7]: #print all data of days which are having cases more than 100000
        for i in range(0, len(covid_data["cases_time_series"])):
            if int(covid_data["cases_time_series"][i]["dailyconfirmed"]) >=100000:
                print("Date : ", covid_data["cases_time_series"][i]["date"],
                  "Cases : ", covid_data["cases_time_series"][i]["dailyconfirmed"])

Date :  4 April 2021 Cases :  103794
Date :  6 April 2021 Cases :  115312
Date :  7 April 2021 Cases :  126276
Date :  8 April 2021 Cases :  131878
Date :  9 April 2021 Cases :  144945
Date :  10 April 2021 Cases :  152565
Date :  11 April 2021 Cases :  169914
Date :  12 April 2021 Cases :  160838
Date :  13 April 2021 Cases :  185297
Date :  14 April 2021 Cases :  199584
Date :  15 April 2021 Cases :  216828
Date :  16 April 2021 Cases :  234002
Date :  17 April 2021 Cases :  260895
Date :  18 April 2021 Cases :  275063
Date :  19 April 2021 Cases :  257003
Date :  20 April 2021 Cases :  294365
Date :  21 April 2021 Cases :  315752
Date :  22 April 2021 Cases :  332531
Date :  23 April 2021 Cases :  345296
Date :  24 April 2021 Cases :  348996
Date :  25 April 2021 Cases :  354658
Date :  26 April 2021 Cases :  319471
Date :  27 April 2021 Cases :  362913
Date :  28 April 2021 Cases :  379404
Date :  29 April 2021 Cases :  386773
Date :  30 April 2021 Cases :  402014
```

Fig 2.4

  - We can also do the following on the API.

```
In [8]: #to print new cases of a specific date
        userdate = input("Enter Date : ")
        for i in range(0, len(covid_data["cases_time_series"])):
            if userdate == covid_data["cases_time_series"][i]["date"]:
                print("New Cases : ", covid_data["cases_time_series"][i]["dailyconfirmed"])
                break
        else:
            print("Date Not Found")

Enter Date : 6 June 2021
New Cases :  101209

In [9]: #print total number of days having cases greater than 100000
        count=0
        for i in range(0, len(covid_data["cases_time_series"])):
            if int(covid_data["cases_time_series"][i]["dailyconfirmed"]) >=100000:
                count=count+1
        print("Total Number of days having cases more than 100000 are : ", count)

Total Number of days having cases more than 100000 are :  63
```

Fig 2.5

Days	Date	Activities
Day 3	31/08/23	• Data Visualization using Matplotlib library

- **Data Visualization using Matplotlib library:**
  - Data visualization using the Matplotlib library involves creating graphical representations of data to provide insights and convey patterns, trends, and relationships. It provides a wide range of tools for creating static, interactive, and publication-quality visualizations, making it an essential tool for data analysts and scientists. A simple implementation of a line chart explains the use of Matplotlib library.

```
In [1]: from matplotlib import pyplot as plt

In [3]: a = [1,3 , 5, 7, 9]
        b = [2, 4, 6, 8, 10]
        plt.plot(a, b, color="red")
        plt.xlabel("X-axis")
        plt.ylabel("Y-axis")
        plt.title("Simple Line Chart")
        plt.show()
```

Fig 3.1

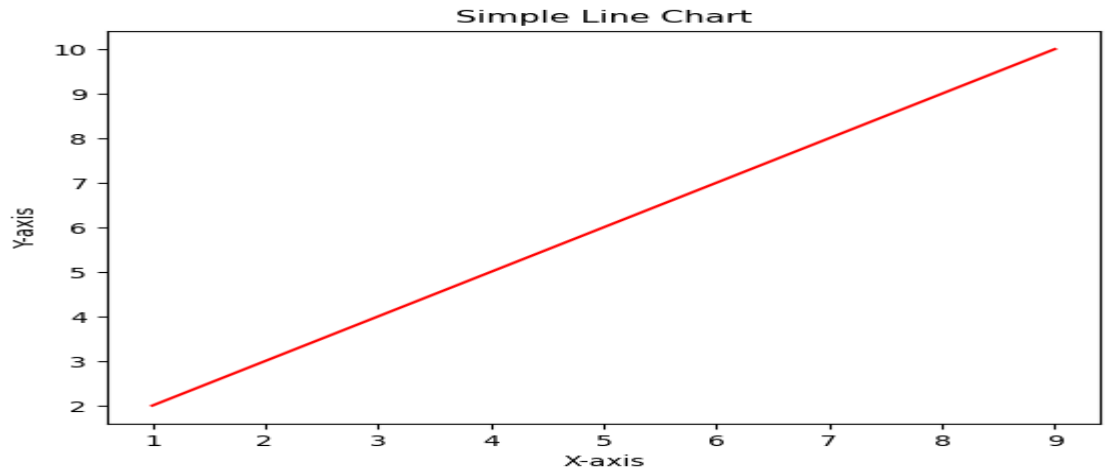


Fig 3.2

- **Types of graph:**

  1. **Bar Graph:**
     - The below given image shows a simple bar graph.

```
In [1]: from matplotlib import pyplot as plt

In [3]: #using lists
        cities = ["Ahmedabad", "Surat", "Rajkot"]
        cases = [275, 195, 75]
        plt.bar(cities, cases, color="blue")
        plt.xlabel("Cities")
        plt.ylabel("Covid Cases")
        plt.title("GUJARAT COVID CASES")
        plt.show()
```

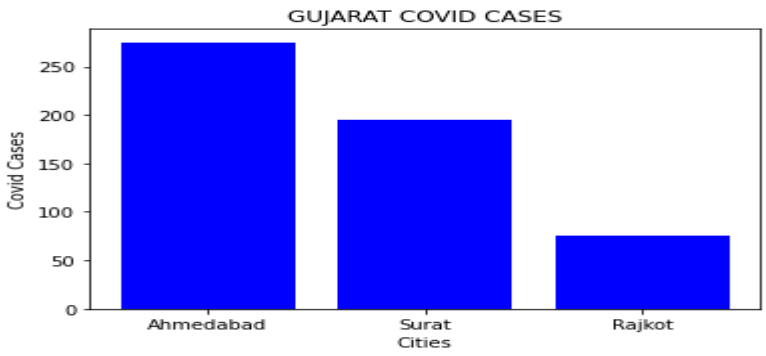


Fig 3.3

  2. **Multiple Line Chart:**
     - We can visualize multiple parameter data in Matplotlib with ease.
     - One of the graphs which helps us in it is Multiple Line Chart.

```
In [4]: dhoni = [15, 75, 20, 65, 99]
        rohit = [22, 7, 45, 19, 35]
        kohli = [2, 5, 13, 75, 4]
        matches = [1, 2, 3, 4, 5]

In [5]: plt.plot(matches, dhoni, label="Dhoni")
        plt.plot(matches, rohit, label="Rohit")
        plt.plot(matches, kohli, label="Kohli")
        plt.legend()
        plt.xlabel("Matches")
        plt.ylabel("Runs")
        plt.title("Multiple line chart")
        plt.show()
```

Fig 3.4

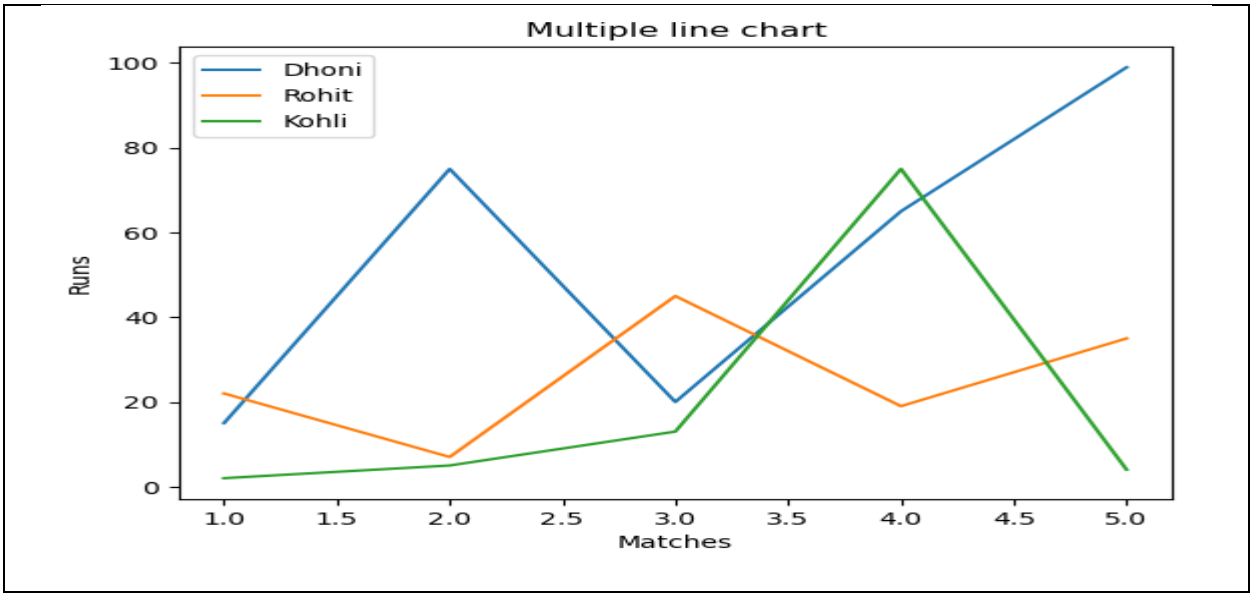


Fig 3.5

3. Multiple Bar Graph:

- A multiple bar graph, also known as a grouped bar chart, is a type of data visualization that displays multiple sets of data side by side in a single chart.
- Below given images describe the representation of a multiple bar graph.

```
In [3]: import numpy as np
        from matplotlib import pyplot as plt
        city_data = {"Ahmedabad": [200, 120, 10], "Surat": [250, 155, 30], "Rajkot": [75, 50, 15]}

        xaxis = np.arange(3)

        d1 = plt.bar(xaxis, city_data["Ahmedabad"], 0.3, label = "Ahmedabad", color ="red")
        d2 = plt.bar(xaxis + 0.3, city_data["Surat"], 0.3, label = "Surat", color="green")
        d3 = plt.bar(xaxis + 0.6, city_data["Rajkot"], 0.3, label = "Rajkot", color="blue")

        plt.xlabel("Cities")
        plt.ylabel("Cases")
        plt.title("Multiple Bar Graph")
        plt.legend((d1, d2, d3), ('Ahm', 'Sur', 'Rjk'))
        plt.show()
```

Fig 3.6

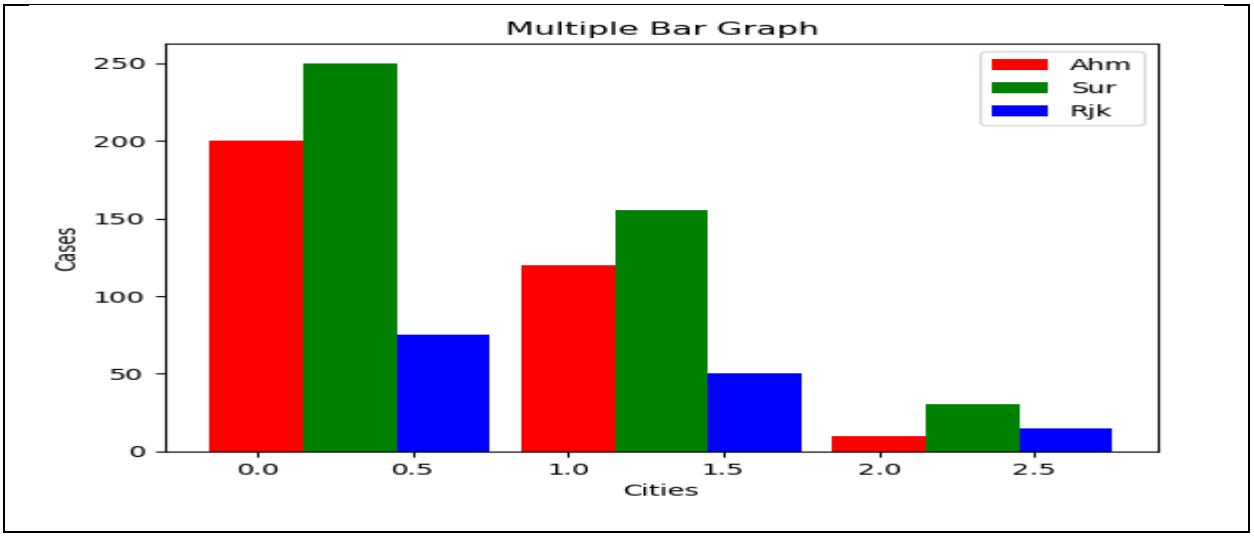


Fig 3.7

4. Pie Chart:

- It is one of the most basic and well-known charts used for data visualization. In this chart the numeric data for a specific parameter or with respect to a specific parameter is used which helps us plot the chart.
- We can understand this by a simple program of implementation of pie chart in python using matplotplib library.

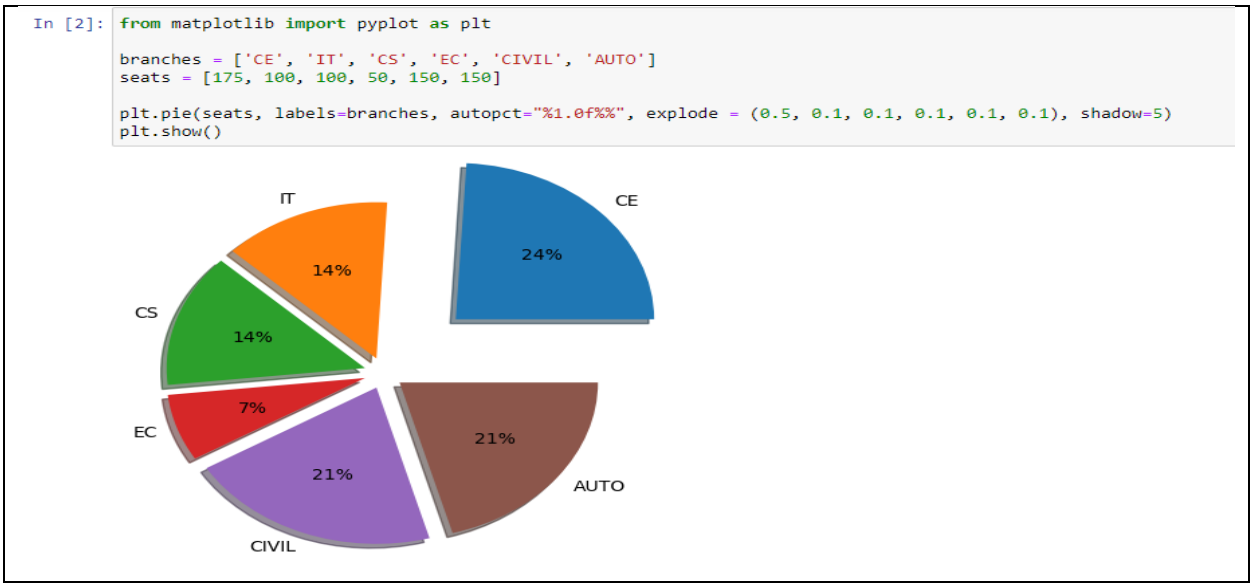


Fig 3.8

Days	Date	Activities
Day 4	01/08/23	• Introduction to Data Visualization • API Data Visualization • Graphs plotted on API Data

- **Introduction to Data Visualization:**
    - Data visualization is the graphical representation of data using visual elements such as charts, graphs, and maps. It's a fundamental aspect of data analysis and communication that transforms raw data into intuitive and meaningful visuals, aiding in understanding, pattern recognition, and insights extraction.

- **API Data Visualization:**
    - API data visualization refers to the practice of retrieving data from an external source or service using an API (Application Programming Interface) and then visualizing that data using various graphical representations, such as charts, graphs, maps, and dashboards. This process enables you to leverage external data and present it in a visually meaningful and informative way.

- **Graphs plotted on API Data:**
    - From the COVID API, below given is a horizontal bar graph of state-wise cases comparison of the covid data.

```
In [2]: import requests
        url = requests.get("https://data.covid19india.org/data.json")
        covid_data = url.json()
        from matplotlib import pyplot as plt

In [13]: state_api = []
         totalconfirm_api = []
         for i in range(1, len(covid_data["statewise"])):
             state_api.append(covid_data["statewise"][i]["state"])
             totalconfirm_api.append(int(covid_data["statewise"][i]["confirmed"]))

In [14]: plt.figure(figsize = (8,8), dpi = 50)
         plt.barh(state_api,  totalconfirm_api, color="purple")
         plt.xlabel("Total_Confirmed")
         plt.ylabel("States")
         plt.title("STATE WISE ANALYSIS")
         plt.xticks([])
         plt.show()
```

Fig 4.1

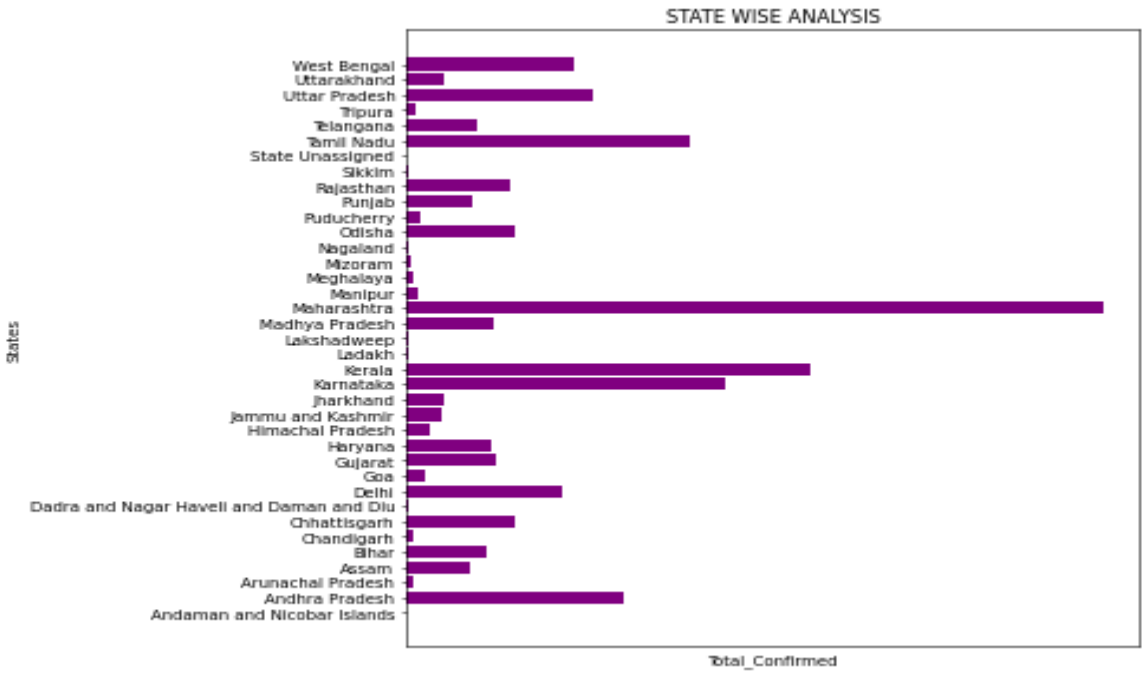


Fig 4.2

- Now a line graph representing the comparison between daily new cases, daily recovered and daily deaths.

```
In [1]: #Covid API
        import requests
        url = requests.get("https://data.covid19india.org/data.json")
        covid_data = url.json()
        from matplotlib import pyplot as plt
```

```
In [2]: newcases_api = []
        recovered_api = []
        death_api = []
        dates_api = []
        for i in range(0, len(covid_data["cases_time_series"])):
            newcases_api.append(int(covid_data["cases_time_series"][i]["dailyconfirmed"]))
            recovered_api.append(int(covid_data["cases_time_series"][i]["dailyrecovered"]))
            death_api.append(int(covid_data["cases_time_series"][i]["dailydeceased"]))
            dates_api.append(covid_data["cases_time_series"][i]["date"])
```

```
In [4]: plt.figure(figsize = (10,8), dpi = 60)
        plt.plot(dates_api, newcases_api, color="blue", label="DailyNewCases")
        plt.plot(dates_api, recovered_api, color="green", label="DailyRecovered")
        plt.plot(dates_api, death_api, color="red", label="DailyDeaths")
        plt.title("COVID Daily Data Analysis")
        plt.xticks([])
        plt.legend()
        plt.show()
```

Fig 4.3

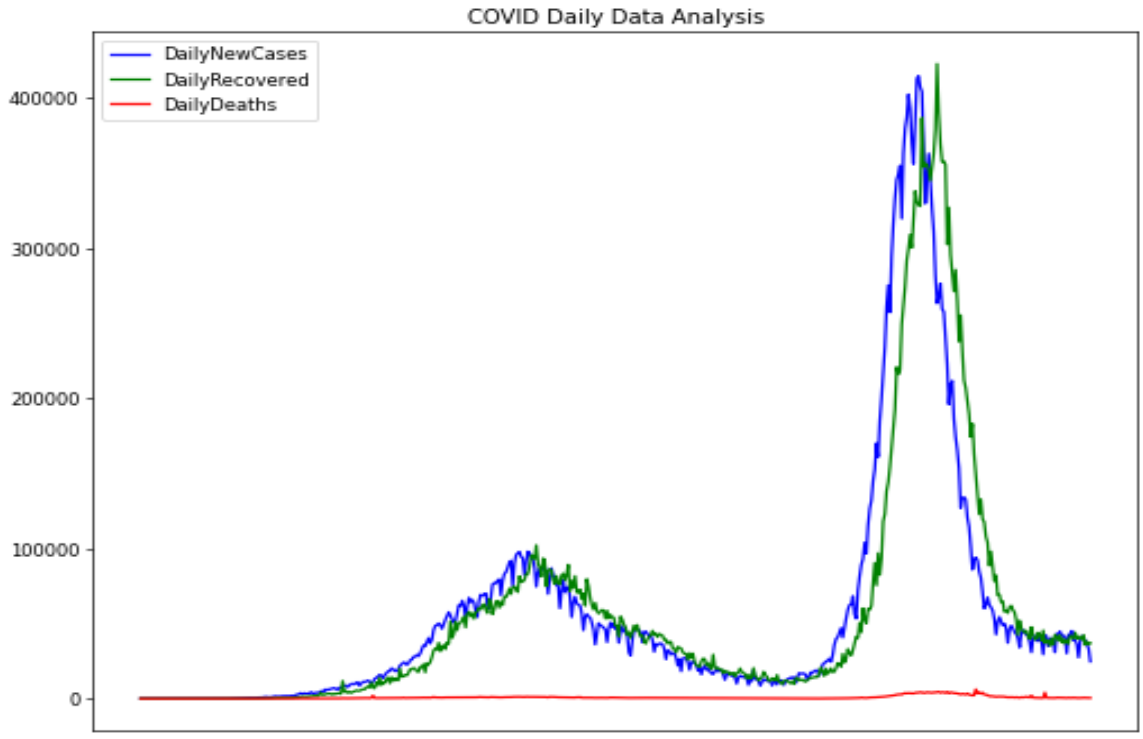


Fig 4.4

- A line graph of number of covid confirmed cases starting from a date up to another date, dates are user inputs.

```
In [12]: s_date = str(input("Enter start Date : "))
         e_date = str(input("Enter end Date : "))
         cases_api = []
         date_api = []

         a = None
         b = None

         for i in range(0, len(covid_data["cases_time_series"])):
             if (covid_data["cases_time_series"][i]["date"] == s_date):
                 a = i
                 break

         for i in range(0, len(covid_data["cases_time_series"])):
             if (covid_data["cases_time_series"][i]["date"] == e_date):
                 b = i
                 break

         for i in range(a, b):
             cases_api.append(covid_data["cases_time_series"][i]["dailyconfirmed"])
             date_api.append(covid_data["cases_time_series"][i]["date"])

         plt.figure(figsize = (8, 8), dpi = 75)
         plt.plot(cases_api, date_api, label = "Cases")
         plt.xlabel("Cases")
         plt.ylabel("Dates")
         plt.show()
```

Fig 4.5

```
Enter start Date : 27 February 2020
Enter end Date : 4 March 2020
```

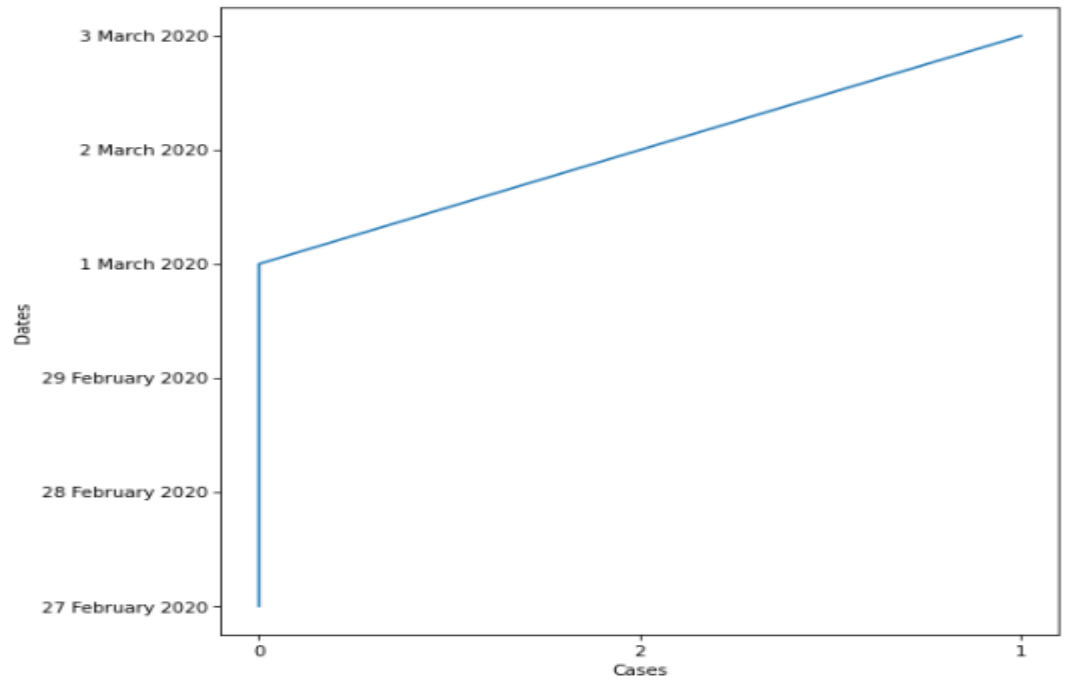


Fig 4.6

Days	Date	Activities
Day 5	02/08/23	• Assignment -1 • News API • ISRO Spacecrafts API • ISRO Customers API • Dynamic API – PINCODE API

- **Assignment – 1**
  - **Task 1:**
  - INSHORTS NEWS API
  - API: https://inshortsapi.vercel.app/news?category=all
    1) How many main keys are there in this API? Extract and print all keys.
    2) How many news are available in this API?
    3) Print all news in below format:

       News content, Author: AUTHOR NAME, DATE: Date of news

       News content, Author: AUTHOR NAME, DATE: Date of news

       News content, Author: AUTHOR NAME, DATE: Date of news
  - **Task 2:**
  - ISRO SPACECRAFTS API
  - API: https://isro.vercel.app/api/spacecrafts
    1) Print all main keys and total number of main keys as well.
    2) Print all spacecrafts names.
    3) Allow user to enter name of spacecraft. Print spacecraft is found or spacecraft not found.
  - **Task 3:**
  - ISRO CUSTOMER API

- API: https://isro.vercel.app/api/customer_satellites

  To generate a pie chart indicating the percentage of ISRO's own spacecrafts vs customer satellites from above API(s) for analysis of domestic vs foreign customer involvement.
- **Task 4:**
- Dynamic API:
- API: https://api.postalpincode.in/pincode/380001
- Allow user to insert PINCODE.
- Print name of all areas which fall under that PINCODE.
- PINCODE entered by user shall be merged in below URL'S XXXXXX part.
- URL: https://api.postalpincode.in/pincode/XXXXXX

- **Task 1:**

1) How many main keys are there in this API? Extract and print all keys.

```
In [1]: import requests
        url = requests.get("https://inshortsapi.vercel.app/news?category=all")
        inshorts_data = url.json()

In [2]: count=0
        for i in inshorts_data:
            count=count+1
        print("The total number of main keys in this API are : ",count)
        print("All the main keys are as follows : ")
        for i in inshorts_data:
            print(i)

        The total number of main keys in this API are :  3
        All the main keys are as follows :
        category
        data
        success
```

Fig 5.1

2) How many news are available in this API?

```
In [3]: x = len(inshorts_data["data"])
        print(x)

        9
```

Fig 5.2

3) Print all news in below format:

   News content, Author: AUTHOR NAME, DATE: Date of news

   News content, Author: AUTHOR NAME, DATE: Date of news

   News content, Author: AUTHOR NAME, DATE: Date of news

```
In [10]: for i in range(0, len(inshorts_data["data"])):
             print("News content : ", inshorts_data["data"][i]["content"])
             print("Author : ", inshorts_data["data"][i]["author"])
             print("DATE : ", inshorts_data["data"][i]["date"])
             print(" ")
```

Fig 5.3

```
News content :  The Thar Desert, which stretches across Rajasthan and parts of Pakistan and is known for its arid landscape, ma
y turn green by century's end due to climate change, a study stated. The mean rainfall in semi-arid northwest regions of India
and Pakistan increased by 10-50% from 1901-2015. Under moderate greenhouse gas scenarios, this rainfall could surge by 50-200%,
researchers said.
Author :  Swati Dubey
DATE :  Thursday, 17 August, 2023

News content :  The rupee weakened 19 paise to close at an all-time low of 83.15 against the US dollar on Thursday as rising US
Treasury Yields and a risk-averse environment weighed on the Indian currency. The rupee had touched its lifetime low of 83.29 o
n October 20, 2022 during intra-day trade. It closed at 82.96 on Monday.
Author :  Srishty Choudhury
DATE :  Thursday, 17 August, 2023

News content :  A group of Australian surfers who went missing at sea were located off the Indonesian coast after 36 hours. The
surfers were aboard wooden longboats between Nias Island and the Pinang Surf resort on Banyak Islands when they ran into bad we
ather. An Indonesian sailor with the group is still missing, while two other Indonesian crew have been rescued.
Author :  Disha Jana
DATE :  Thursday, 17 August, 2023

News content :  Unacademy has removed its teacher named Karan Sangwan who advised students to vote for a politician who is well
-educated. There were calls to boycott Unacademy on social media over his remarks. "Don't elect someone who only knows changing
names," he had said. Several X (Twitter) users accused him of pushing a political agenda.
Author :  Deepika Bhatt
DATE :  Thursday, 17 August, 2023

News content :  Bill Gates-backed UK biotech company Oxitec has engineered "super mosquitoes" that combat disease-spreading cou
nterpart responsible for 6,00,000 annual deaths. Homeowners can purchase these kits to protect their yards. These male mosquito
es carry a gene preventing female survival, curbing malaria transmission. If these modified males mate with wild females, all f
emale offspring allegedly die.
Author :  Swati Dubey
DATE :  Thursday, 17 August, 2023
```

Fig 5.4

- **Task 2:**

  1) Print all main keys and total number of main keys as well.

```
In [1]: import requests
        url = requests.get("https://isro.vercel.app/api/spacecrafts")
        isro_data = url.json()

In [2]: count=0
        for i in isro_data:
            count=count+1
        print("The total number of main keys in this API are : ",count)
        print("All the main keys are as follows : ")
        for i in isro_data:
            print(i)

        The total number of main keys in this API are :  1
        All the main keys are as follows :
        spacecrafts
```

Fig 5.5

2) Print all spacecrafts names.

```
In [3]: for i in range(0, len(isro_data["spacecrafts"])):
            print(isro_data["spacecrafts"][i]["id"], ": ", isro_data["spacecrafts"][i]["name"])

        1 :  Aryabhata
        2 :  Bhaskara-I
        3 :  Rohini Technology Payload (RTP)
        4 :  Rohini Satellite RS-1
        5 :  Rohini Satellite RS-D1
        6 :  APPLE
        7 :  Bhaskara-II
        8 :  INSAT-1A
        9 :  Rohini Satellite RS-D2
        10 :  INSAT-1B
        11 :  SROSS-1
        12 :  IRS-1A
        13 :  SROSS-2
        14 :  INSAT-1C
        15 :  INSAT-1D
        16 :  IRS-1B
        17 :  SROSS-C
        18 :  INSAT-2A
        19 :  INSAT-2B
```

Fig 5.6

3) Allow user to enter name of spacecraft. Print spacecraft is found or spacecraft not found.

```
In [4]: name_sc = input("Enter the name of the spacecraft: ")
        for i in range(0, len(isro_data["spacecrafts"])):
            if(isro_data["spacecrafts"][i]["name"] == name_sc):
                print("Spacecraft is Found")
                break
        else:
            print("Spacecraft is NOT Found")

        Enter the name of the spacecraft: INSAT-2D
        Spacecraft is Found
```

Fig 5.7

- **Task 3:**

  To generate a pie chat indicating the percentage of ISRO's own spacecrafts vs customer satellites from above API(s) for analysis of domestic vs foreign customer involvement.

```
In [2]: import requests
        from matplotlib import pyplot as plt
        url1 = requests.get("https://isro.vercel.app/api/customer_satellites")
        url2 = requests.get("https://isro.vercel.app/api/spacecrafts")
        ownsc_data = url2.json()
        customersc_data = url1.json()

In [3]: x = len(ownsc_data["spacecrafts"])
        print(x)

        112

In [4]: y = len(customersc_data["customer_satellites"])
        print(y)

        53

In [7]: satellites = ['Domestic', 'Foreign_Customers']
        num = [x, y]

In [13]: plt.pie(num, labels=satellites, shadow=5, autopct="%1.0f%%")
         plt.show()
```

Fig 5.8

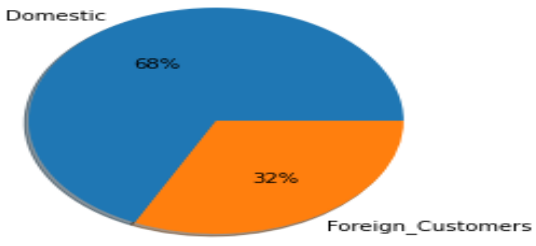


Fig 5.9

- **Task 4:**

  Allow user to insert pincode.

  Print name of all areas which fall under that pincode.

  Pincode entered by user shall be merged in below URL'S XXXXXX part.

  URL: https://api.postalpincode.in/pincode/XXXXXX

```
In [1]: import requests
        user_inp = input("Enter the PINCode: ")
        url = requests.get("https://api.postalpincode.in/pincode/"+ user_inp)
        postal_data = url.json()
        print("Areas under pincode", user_inp, "are:")
        for i in range(0, len(postal_data[0]["PostOffice"])):
            print(postal_data[0]["PostOffice"][i]["Name"])
```

Fig 5.10

```
Enter the PINCode: 380058
Areas under pincode 380058 are:
Ambli
Bopal
Ghuma
Shela
```

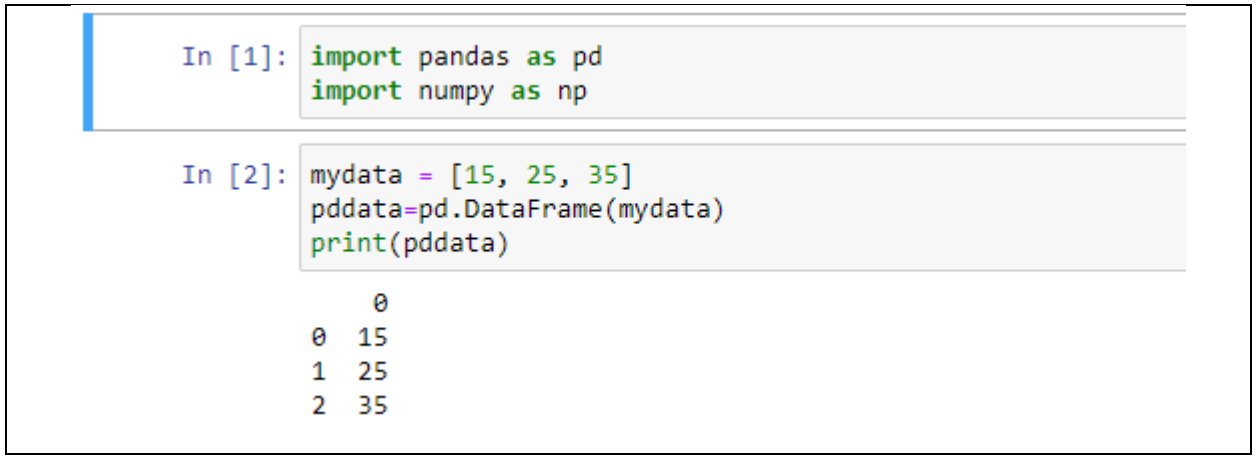
```
Enter the PINCode: 380059
Areas under pincode 380059 are:
Shilaj
Thaltej
```

Fig 5.11

Days	Date	Activities
Day 6	03/08/23	• Introduction to Pandas and NumPy library • Pandas: Data Frames • Pandas: Excel • Pandas: API to CSV • XLRD: IPL Data Analysis

- **Introduction to Pandas and Numpy Library**
  - Pandas is a powerful and versatile open-source Python library for data manipulation and analysis.
  - It provides data structures and functions that make it easy to work with structured data, making it a fundamental tool for data scientists, analysts, and researchers.Pandas is a foundational library for data manipulation and analysis in Python.
  - NumPy (Numerical Python) is a fundamental open-source library for numerical computations in Python.
  - It provides support for large, multi-dimensional arrays and matrices, along with a wide range of mathematical functions to operate on these arrays.
  - NumPy forms the foundation of many scientific and data analysis libraries in the Python ecosystem.

- **Pandas: DataFrames:**
  - DataFrame is one of the core data structure in the Pandas library designed to organize and manipulate data in a tabular format which is similar to a database or a spreadsheet.
  - It offers a versatile and potent approach for handling structured data, encompassing activities like data analysis, manipulation, conversion, and graphical representation.
  - Here are a few examples of working with DataFrames:



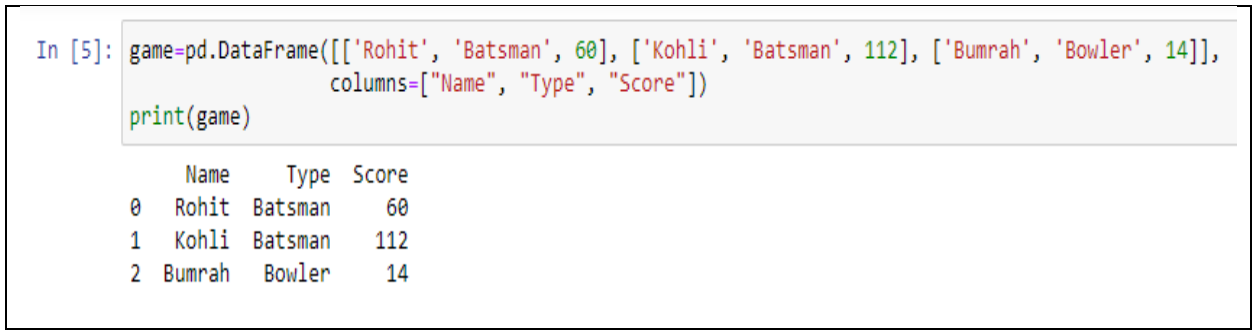
```
In [1]: import pandas as pd
        import numpy as np

In [2]: mydata = [15, 25, 35]
        pddata=pd.DataFrame(mydata)
        print(pddata)

            0
        0  15
        1  25
        2  35
```

Fig 6.1

  - Other way to enter the data in DataFrames is:

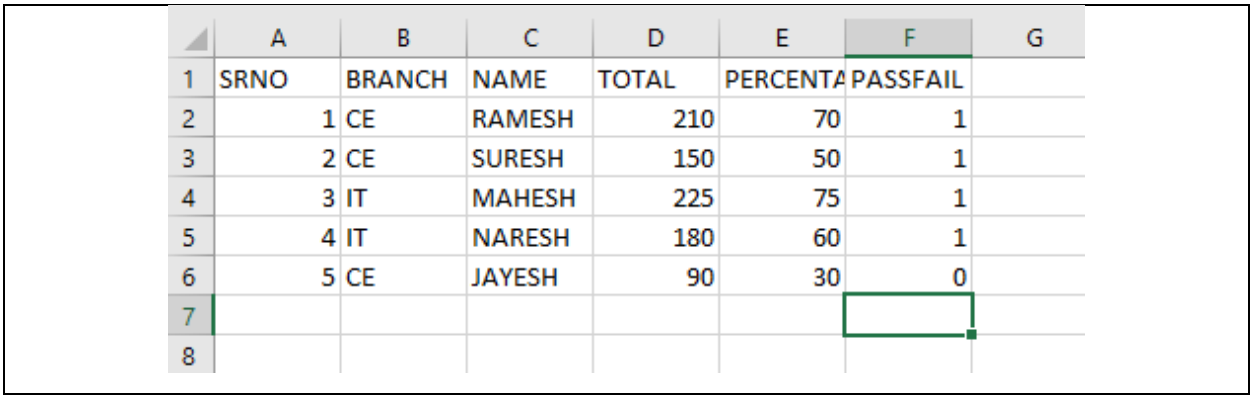


```
In [5]: game=pd.DataFrame([['Rohit', 'Batsman', 60], ['Kohli', 'Batsman', 112], ['Bumrah', 'Bowler', 14]],
                          columns=["Name", "Type", "Score"])
        print(game)

             Name     Type  Score
        0   Rohit  Batsman     60
        1   Kohli  Batsman    112
        2  Bumrah   Bowler     14
```

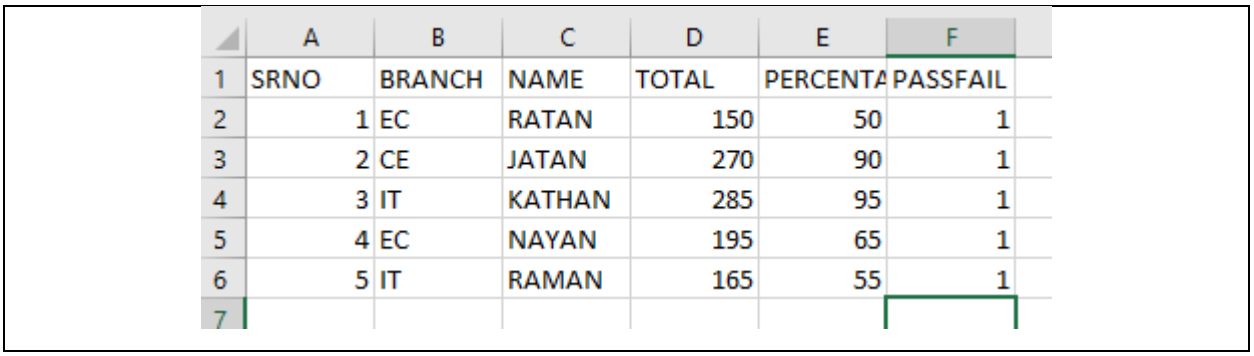
Fig 6.2

- **Pandas: Excel:**
  - Using excel files grants us the ability to write, modify, read, and manipulate data in a tabular format. Through Pandas, we can easily and efficiently manipulate, analyze and modify data. Through the example given below, we can understand how we work on excel file using Pandas.



	A	B	C	D	E	F	G
1	SRNO	BRANCH	NAME	TOTAL	PERCENTA	PASSFAIL	
2	1	CE	RAMESH	210	70	1	
3	2	CE	SURESH	150	50	1	
4	3	IT	MAHESH	225	75	1	
5	4	IT	NARESH	180	60	1	
6	5	CE	JAYESH	90	30	0	
7							
8							

Fig 6.3



	A	B	C	D	E	F
1	SRNO	BRANCH	NAME	TOTAL	PERCENTA	PASSFAIL
2	1	EC	RATAN	150	50	1
3	2	CE	JATAN	270	90	1
4	3	IT	KATHAN	285	95	1
5	4	EC	NAYAN	195	65	1
6	5	IT	RAMAN	165	55	1
7						

Fig 6.4

  - In this example there are two excel files namely, ‘Result1’ and ‘Result2’.
  - Printing the data of these files by importing them using Pandas.

```
In [1]: import pandas as pd
        import numpy as np
        from matplotlib import pyplot as plt

In [2]: file_1 = pd.read_excel("RESULT1.xlsx")
        print(file_1)
        print("========================")
        print(type(file_1))
```

```
   SRNO BRANCH    NAME  TOTAL  PERCENTAGE  PASSFAIL
0     1     CE  RAMESH    210          70         1
1     2     CE  SURESH    150          50         1
2     3     IT  MAHESH    225          75         1
3     4     IT  NARESH    180          60         1
4     5     CE  JAYESH     90          30         0
========================
<class 'pandas.core.frame.DataFrame'>
```

```
In [3]: file_2 = pd.read_excel("RESULT2.xlsx")
        print(file_2)
        print("========================")
        print(type(file_2))
```

```
   SRNO BRANCH    NAME  TOTAL  PERCENTAGE  PASSFAIL
0     1     EC   RATAN    150          50         1
1     2     CE   JATAN    270          90         1
2     3     IT  KATHAN    285          95         1
3     4     EC   NAYAN    195          65         1
4     5     IT   RAMAN    165          55         1
========================
<class 'pandas.core.frame.DataFrame'>
```

Fig 6.5

- Concatenate these files using Pandas by using the ‘pd.concat’ method.

```
In [5]: files_data = pd.concat([file_1, file_2])
        print(files_data)
```

```
   SRNO BRANCH    NAME  TOTAL  PERCENTAGE  PASSFAIL
0     1     CE  RAMESH    210          70         1
1     2     CE  SURESH    150          50         1
2     3     IT  MAHESH    225          75         1
3     4     IT  NARESH    180          60         1
4     5     CE  JAYESH     90          30         0
0     1     EC   RATAN    150          50         1
1     2     CE   JATAN    270          90         1
2     3     IT  KATHAN    285          95         1
3     4     EC   NAYAN    195          65         1
4     5     IT   RAMAN    165          55         1
```

Fig 6.6

- This grants us a very quick and efficient way to concatenate our files.
- To plot a bar-graph for the students vs their total.

```
In [6]: allname = list(files_data['NAME'])
        total = list(files_data['TOTAL'])
        plt.figure(figsize = (8, 8), dpi = 75)
        plt.bar(allname, total)
        plt.xlabel("Names")
        plt.ylabel("Total")
        plt.show()
```

Fig 6.7

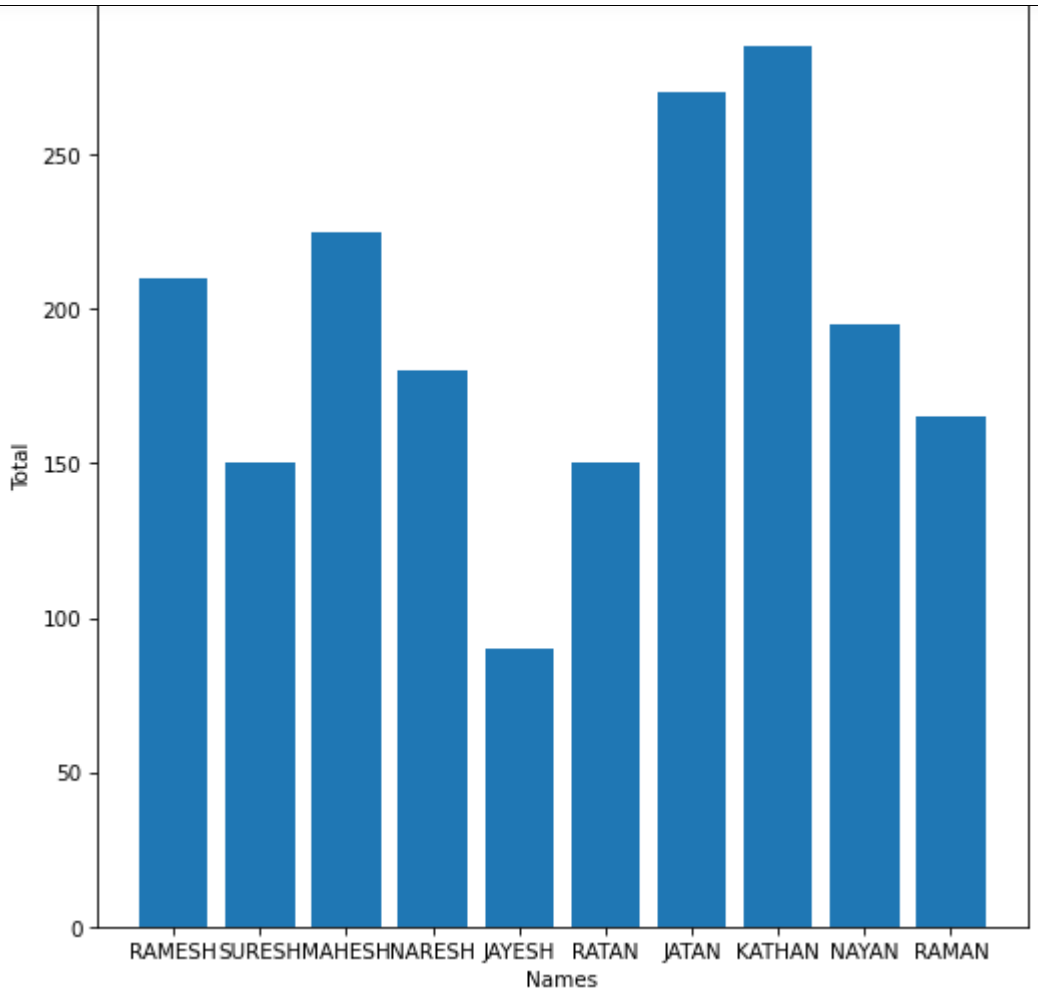


Fig 6.8

- **Pandas: API to CSV:**
    - In order to convert an API into CSV file we have to keep a few steps in mind.
    - Firstly, we fetch the data from the API. Then we convert the obtained json data into Pandas DataFrame. Then we make use of the '.to_csv' method of Pandas to convert our DataFrame into CSV file.
    - The instance given below explains the following:

```
In [1]: import pandas as pd
        import requests

In [2]: url = requests.get("https://data.covid19india.org/data.json")
        simple_data = url.json()

In [3]: df = pd.DataFrame(simple_data["cases_time_series"])

In [4]: #api to csv
        df.to_csv("dailycovidcases.csv")
```

Fig 6.9

    - The 'dailycovidcases.csv' file is too large as it contains all the COVID API data in it.

Days	Date	Activities
Day 7	04/08/23	• Introduction to Machine Learning • Introduction to Linear Regression • Linear Model: Mathematics • Linear Model: Implementation

- **Linear Model: Implementation:**
  - For the Implementation of the Simple Linear Regression Model, we need the following libraries- NumPy, Pandas, Matplotlib or Seaborn.
  - Consider the example as given below for 'Exam.csv' file.

```
In [1]: import pandas as pd
        from matplotlib import pyplot as plt
        from sklearn import linear_model

In [2]: df=pd.read_csv("Exam.csv")

In [3]: from sklearn.linear_model import LinearRegression
        #load csv data using pandas
        data_csv=pd.read_csv("Exam.csv")
        #extract x and y values from the data
        X=data_csv['hours'].values.reshape(-1, 1)
        Y=data_csv['score'].values
        #create a linear regression model
        rmodel=LinearRegression()
        #fit the model to the data
        rmodel.fit(X, Y)
        #make predictions using the model
        pred=rmodel.predict(X)
        #plot the original data and the regression line
        plt.scatter(X, Y, label='Original Data', color="purple")
        plt.plot(X, pred, color='pink', label='Regression Line')
        plt.xlabel('Hours_Studied')
        plt.ylabel('Score_Obtained')
        plt.legend()
        plt.title('Linear Regression')
        plt.show()
```

Fig 7.1

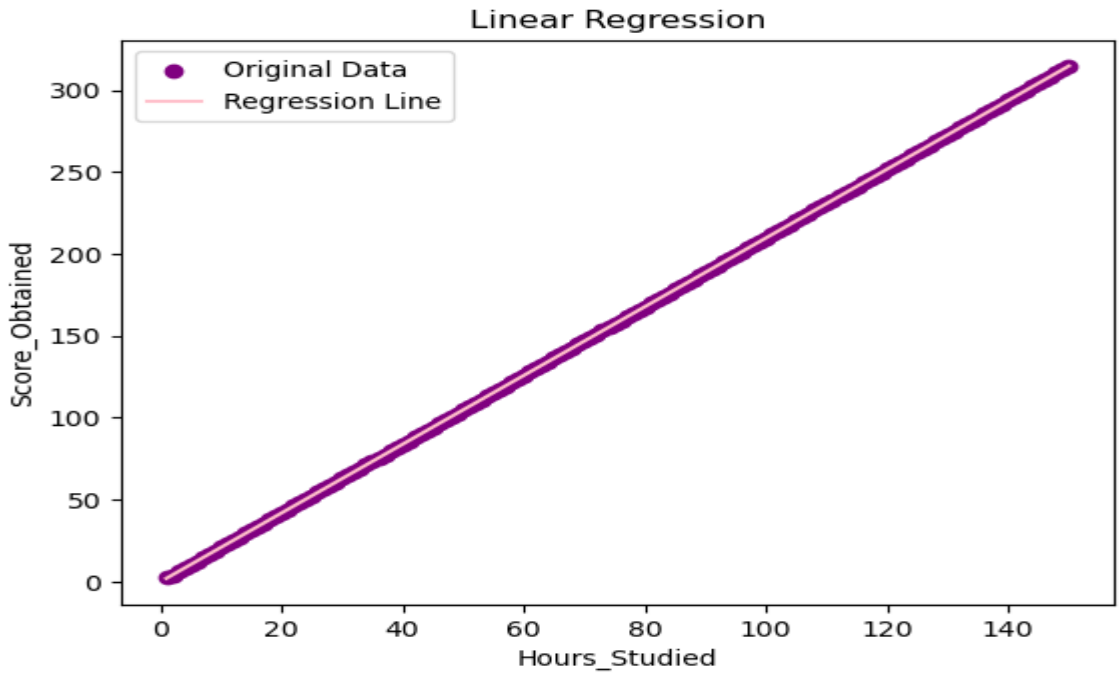


Fig 7.2

- Afterwards, to predict the score based on new data of hours.

```
In [16]: reg=linear_model.LinearRegression()
         reg.fit(df[['hours']], df[['score']])
         print(reg.predict([[360]]))

         [[755.67939073]]
```

Fig 7.3

- As you can see the predicted score value is 755.67939073 for the hours value 360.

Days	Date	Activities
Day 8	07/08/23	• Multiple Linear Regression • Multiple Linear Model: Mathematics • Multiple Linear Model: Implementation

- **Multiple Linear Model: Implementation:**
  - The libraries required in this type of regression model are the same as that of simple linear regression model that is NumPy, Pandas, and Matplotlib or Seaborn.
  - The below mentioned example explains the use of this model, the 'Advertisement.csv' file is used here.

```
In [1]: import pandas as pd
        from matplotlib import pyplot as plt
        from sklearn import linear_model

In [2]: df=pd.read_csv("Advertising.csv")
        print(df)

            tv  radio  newspaper  sales
0    230.1   37.8       69.2   22.1
1     44.5   39.3       45.1   10.4
2     17.2   45.9       69.3    9.3
3    151.5   41.3       58.5   18.5
4    180.8   10.8       58.4   12.9
..     ...    ...        ...    ...
195   38.2    3.7       13.8    7.6
196   94.2    4.9        8.1    9.7
197  177.0    9.3        6.4   12.8
198  283.6   42.0       66.2   25.5
199  232.1    8.6        8.7   13.4

[200 rows x 4 columns]
```

Fig 8.1

```
In [3]: regr=linear_model.LinearRegression()
        regr.fit(df[['tv', 'radio', 'newspaper']], df.sales)
        print(regr.predict([[350.6, 56.7, 22.8]]))

        [29.64997118]
```

Fig 8.2

Days	Date	Activities
Day 9	08/08/23	• Polynomial Linear Regression • Polynomial Linear Regression model: Implementation • Image Data

- **Polynomial Linear Regression model: Implementation:**
  - A csv file named 'new_order.csv' is used here to better understand the implementation concept of Polynomial linear regression.

```
1  orders,slot,amount
2  daily_orders,1,44500
3  half weekly orders,2,127000
4  weekly orders,3,304500
5  half monthly orders,4,667500
6  monthly orders,5,1350000
7  half quarterly orders,6,2047500
8  quarterly orders,7,4041000
```

Fig 9.1

  - First, we import all the necessary libraries and the file and then print all the data.

```
In [7]: import pandas as pd
        from matplotlib import pyplot as plt
        import numpy as np
        from sklearn.linear_model import LinearRegression
        from sklearn.preprocessing import PolynomialFeatures
        from sklearn.model_selection import train_test_split

In [8]: df=pd.read_csv("new_order.csv")
        print(df)

                      orders  slot   amount
0            daily_orders     1    44500
1     half weekly orders      2   127000
2           weekly orders     3   304500
3    half monthly orders      4   667500
4          monthly orders     5  1350000
5  half quarterly orders      6  2047500
6       quarterly orders      7  4041000
```

Fig 9.2

  - Now plot a simple chart using this model.

```
In [9]: plt.scatter(df.slot, df.amount)
        plt.show()
```

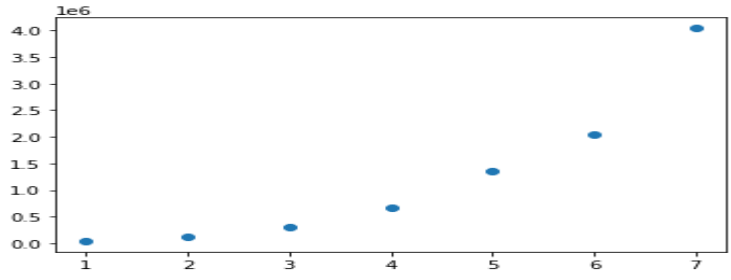


Fig 9.3

- **Image Data:**
  - In machine learning, image data plays a crucial role as a type of input for training and evaluating models. It consists of visual information represented as pixel values in a grid, where each pixel corresponds to a specific color or intensity.

```
In [6]: import tensorflow as tf
        import numpy as np
        import matplotlib.pyplot as plt
        import cv2
        import os
        from tensorflow.keras.preprocessing import image
        from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

Fig 9.4

```
In [11]: img = image.load_img("Image_Data/Training/Laptop/L (6).jpg")
         plt.imshow(img)
Out[11]: <matplotlib.image.AxesImage at 0x1c93cc21890>
```

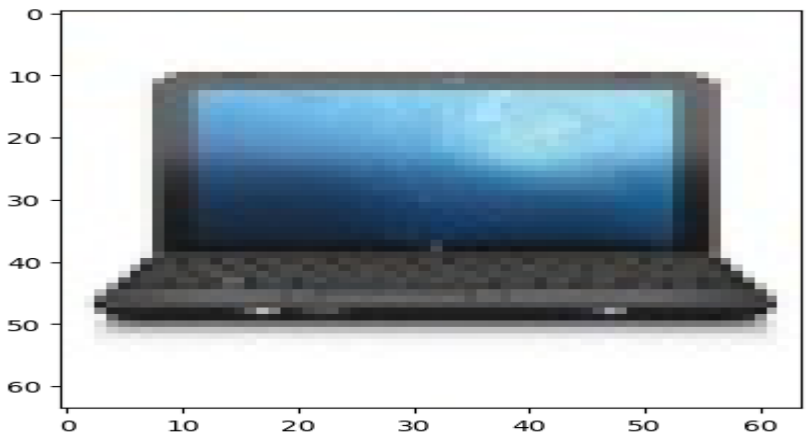


Fig 9.5

Days	Date	Activities
Day 10	09/08/23	• Assignment – 2 • OpenCV • Convolutional Neural Network Project

- **Assignment – 2:**
  - Write script to Open Camera.
  - Write script to Open Camera in gray scale mode (black and white).
  - Write script to Capture image on click of key ‘c’ while camera is open.
  - Create a folder: Image data
    - Add three folders training, validation and testing
    - Add two folders laptop and mobile in training and validation folders.
    - Add images in all 4 folders (respective images laptop/mobile)
    - Develop a Convolutional Neural Network (CNN) which could predict the images from the testing folder. (whether it is a mobile or a laptop)

- **OpenCV:**
  - To start the camera.

```
In [2]: #write script to Open Camera
        import cv2
        camera = cv2.VideoCapture(0)
        while(True):
            ret, frame = camera.read()

            cv2.imshow('RESULT', frame)
            if cv2.waitKey(1) & 0xFF==ord('c'):#use 'c' to close the camera
                break

        camera.release()
        cv2.destroyAllWindows()
```

Fig 10.1

- To start camera in gray scale mode.

```
In [5]: #write script to Open Camera in gray scale mode (black and white)
        import cv2

        camera = cv2.VideoCapture(0)
        while(True):
            ret, frame = camera.read()
            gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
            cv2.imshow('RESULT', gray)
            if cv2.waitKey(1) & 0xFF==ord('c'):#use 'c' to close the camera
                break

        camera.release()
        cv2.destroyAllWindows()
```

Fig 10.2

- To click a picture by clicking the ‘c’ key.

```
In [9]: #write script to Capture image on click of key "c" while camera is open
        import cv2
        camera = cv2.VideoCapture(0)
        img_counter = 0
        while(True):
            ret, frame = camera.read()

            if not ret:
                print("Failed to Grab a frame")
                break

            cv2.imshow('Test', frame)

            k = cv2.waitKey(1)
            if k%256==27:
                print("Escape key is used for closing the app")
                break

            elif k%256==ord('c'):
                img_name = "opencv_frame_{}.png".format(img_counter)
                cv2.imwrite(img_name, frame)
                print("Image Taken")
                img_counter+=1

        camera.release()
        cv2.destroyAllWindows()

        Image Taken
        Escape key is used for closing the app
```

Fig 10.3

- **CNN Task:**

In [1]:
```
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
import os
import matplotlib.pyplot as plt
import numpy as np
from PIL import Image
```

In [2]:
```
train_dir = 'Image_Data/Training/'
validation_dir = 'Image_Data/Validation/'
test_dir = 'Image_Data/Testing/'
```

In [4]:
```
image_width = 50
image_height = 50
batch_size = 32
epochs = 10
```

In [5]:
```
train_datagen = ImageDataGenerator(
    rescale=1.0/255.0,  # Rescale pixel values to [0, 1]
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)
```

In [6]:
```
validation_datagen = ImageDataGenerator(rescale=1.0/255.0)
```

In [7]:
```
train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(image_width, image_height),
    batch_size=batch_size,
    class_mode='binary'
)
```

```
Found 175 images belonging to 2 classes.
```

In [8]:
```
validation_generator = validation_datagen.flow_from_directory(
    validation_dir,
    target_size=(image_width, image_height),
    batch_size=batch_size,
    class_mode='binary'
)
```

```
Found 27 images belonging to 2 classes.
```

Fig 10.4

In [9]:
```
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(image_width, image_height, 3)),
    MaxPooling2D(2, 2),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Flatten(),
    Dense(512, activation='relu'),
    Dropout(0.5),
    Dense(1, activation='sigmoid')
])
```

In [10]:
```
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

Fig 10.5

In [11]:
```
history = model.fit(
    train_generator,
    steps_per_epoch=train_generator.samples // batch_size,
    epochs=epochs,
    validation_data=validation_generator,
    validation_steps=validation_generator.samples // batch_size
)
```

```
Epoch 1/10
5/5 [==============================] - 10s 654ms/step - loss: 0.7074 - accuracy: 0.4755
Epoch 2/10
5/5 [==============================] - 1s 168ms/step - loss: 0.7080 - accuracy: 0.4685
Epoch 3/10
5/5 [==============================] - 1s 189ms/step - loss: 0.6951 - accuracy: 0.4895
Epoch 4/10
5/5 [==============================] - 1s 189ms/step - loss: 0.6981 - accuracy: 0.4825
Epoch 5/10
5/5 [==============================] - 1s 168ms/step - loss: 0.6791 - accuracy: 0.5664
Epoch 6/10
5/5 [==============================] - 1s 167ms/step - loss: 0.6775 - accuracy: 0.5944
Epoch 7/10
5/5 [==============================] - 1s 164ms/step - loss: 0.6668 - accuracy: 0.6014
Epoch 8/10
5/5 [==============================] - 1s 194ms/step - loss: 0.6404 - accuracy: 0.6434
Epoch 9/10
5/5 [==============================] - 1s 157ms/step - loss: 0.6080 - accuracy: 0.6573
Epoch 10/10
5/5 [==============================] - 1s 168ms/step - loss: 0.5943 - accuracy: 0.6573
```

Fig 10.6

```
In [12]: test_datagen = ImageDataGenerator(rescale=1.0/255.0)

In [13]: class_labels = train_generator.class_indices
         class_labels = {v: k for k, v in class_labels.items()}

In [14]: test_generator = test_datagen.flow_from_directory(
             test_dir,
             target_size=(image_width, image_height),
             batch_size=batch_size,
             class_mode='binary',
             shuffle=False
         )

         Found 91 images belonging to 2 classes.
```

Fig 10.7

```
In [15]: num_samples = len(test_generator)
         for i in range(num_samples):
             image, actual_label = test_generator[i]
             actual_label = int(actual_label[0])  # Convert one-hot encoded label to integer

             predicted_prob = model.predict(image)[0][0]  # Predicted probability
             predicted_label = 1 if predicted_prob > 0.5 else 0  # Convert probability to label

             actual_class = class_labels[actual_label]
             predicted_class = class_labels[predicted_label]

             image_array = np.asarray(image[0])  # Convert image to array
             image_pil = Image.fromarray(np.uint8(image_array*255))

             plt.imshow(image_pil)
             plt.title(f'Actual: {actual_class}, Predicted: {predicted_class}')
             plt.show()
```

Fig 10.8

- The output of the above given code shows the prediction made using CNN.
- The output is as follows:

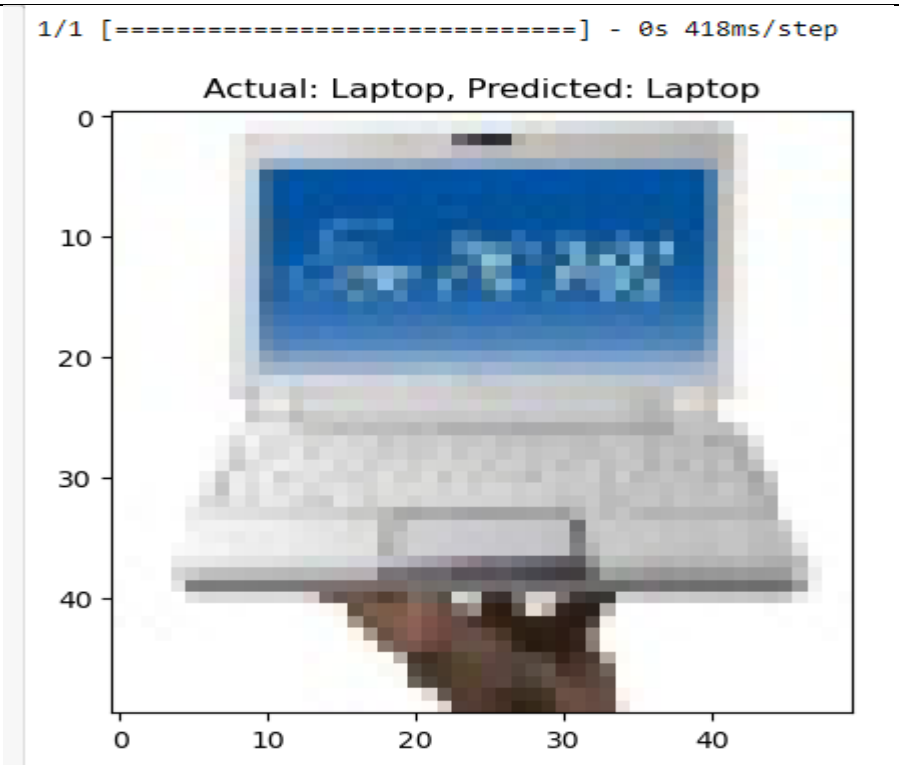


Fig 10.9

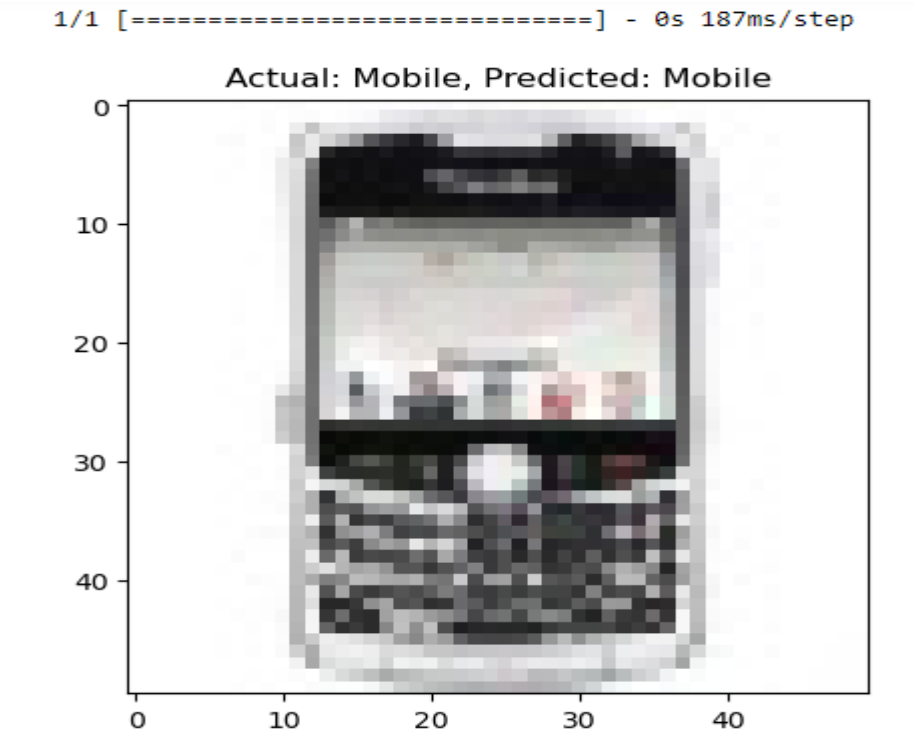


Fig 10.10

CHAPTER: 5

Conclusion

5.1 Conclusion

In conclusion, this internship journey has been very productive and great which aided me to learn new things such as API data handling, creating better data visualization of large and dynamic data like APIs, and the machine learning concepts like regression and its types and machine learning model generation.

This internship assisted me to polish my skills, enhance my performance and ingest new concepts regarding Data Analytics and Machine Learning.

This training provided practical knowledge to Data Analytics techniques, Data Visualization concepts and a few Machine learning concepts.

With the internship drawing to a close, the skills obtained and insights gained establish a precious base for ongoing advancement within the technology sector. The incorporation of theories and practical experience underscores the tangible applicability of technology in real-world contexts.

CHAPTER: 6

Future Enhancement

6.1 Future Enhancement

With this remarkable learning journey behind me and a genuine attraction to Data Analytics and Machine Learning, my future course involves thorough exploration followed by the application of these concepts in project endeavors.

My future aims related to this field will mainly include projects which would solely focus on the Data Analysis concepts and models made using Machine Learning concepts and algorithms.