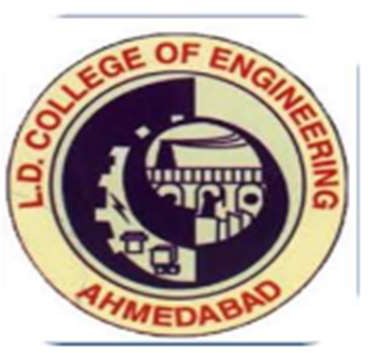


**GUJARAT TECHNICAL UNIVERSITY**

**Academic Year: (2022-23)**



**SUMMER INTERNSHIP REPORT**

***Submitted by***

**KRUNAL JAYESHBHAI MISTRY**

**(190280107070)**

***In partial fulfilment for the award of the degree of***

**BACHELOR OF ENGINEERING**

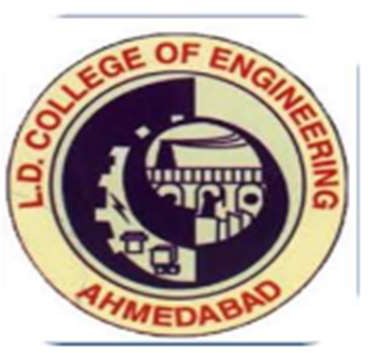
**IN**

**BE Semester – VII Computer Engineering**

**L.D. College of Engineering – Ahmedabad**

Prof. Hetal Gevariya Sagar Jasani

(Internal Guide) (External Guide)



**L.D. COLLEGE OF ENGINEERING**

Ahmedabad

**CERTIFICATE**

This is certify that the summer internship project report submitted along with project Entitled Recommendation System has been carried out by **KRUNAL JAYESHBHAI** **MISTRY** enrolment number **190280107070** under my guidance in partial fulfilment for the degree of Bachelor of Engineering in Computer Engineering, 7th semester of GTU, Ahmedabad during academic year 2022-23

Prof. Hetal Gevariya Dr. Chirag S. Thaker

Internal Guide Head of Department

**Company Profile**





**Company Name:** BrainyBeam Technologies Pvt Ltd

**Address:** 118, Sukan Mall , Science city road, Ahmedabad

**Contact No:** +91 9033237336

**Email Id :** [sagar@brainybeam.com](mailto:sagar@brainybeam.com)

**Website:** [www.brainybeam.com](http://www.brainybeam.com/)

**About Us**

At BrainyBeam, we see Innovation as a clear differentiator. Innovation, along with focus on deep, long-lasting client relationships and strong domain expertise, drives every facet of our day-to-day operations.

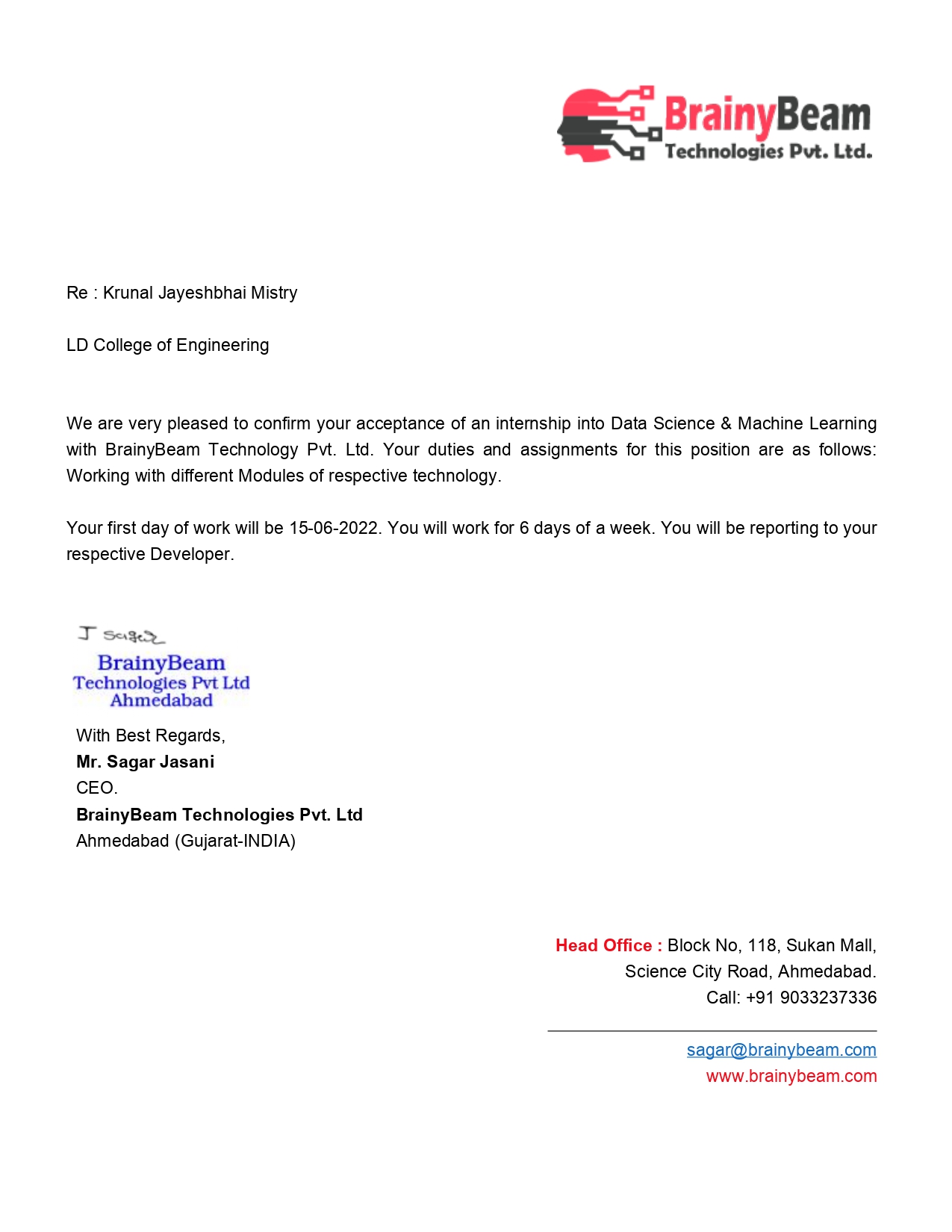
BrainyBeam Technologies was founded with a vision to address growing businesses' needs of reducing the time to market and cost effectiveness required to develop and maintain unique and customized web and mobile solutions. We are uniquely and strategically positioned to partner with startups and leading brands to help them expand their business and offer the most effective and cost-efficient solutions that provide revenues and value to their business needs.

**Vision**

To become the most trusted and preferred offshore IT solutions partner for Startups, SMBs and Enterprises through innovation and technology leadership. Understanding your ambitious vision, honing in on its essence, creating a design strategy, and knowing how to technically execute it is what we do best. Our promise? The integrity of your vision will be maintained and we'll enhance it to best reach your target customers. With our primary focus on creating amazing user experiences, we'll help you understand the tradeoffs, prioritize features, and distill valuable functionality. It's an art form we care about getting right.

**Joining Letter**



****



**Completion Certificate**

****

## **ACKNOWLEDGEMENT**

I would like to express my deepest gratitude to all those who provided me the possibility to the completion of the internship. A special gratitude of thanks I give to our Assistant Professor, Prof. Hetal Gevariya, whose contribution in stimulating suggestions and encouragement, helped me to coordinate the internship especially in drafting this report.

Furthermore, I would also like to acknowledge with much appreciation the crucial role of the Head of Department, Dr. Chirag S. Thaker, who gave the permission to use all required equipment and the necessary material to fulfil the task. Last but not the least, many thanks go to the teachers and my friends and families who have invested their full effort in guiding us in achieving the goal.

Also I appreciate the guidance given by the developer at BrainyBeam, Mr Raj as well as the panels especially for the internship that has advised me and gave guidance at every moment of the internship.

## **Abstract**

Data Science and analysis is playing the most significant role today covering every industry in the market. For e.g., finance, e-commerce, business, education, government.

Now organizations play a 360-degree role to analyse the behaviour and interest of their customers to take decisions in favour of them. Data is analysed through programming language such as python which is one of the most versatile language and helps in doing a lot of things through it.

Netflix is a pure data science project that reached at the top through analysing every single interest of their customers. Key terminology that are used in Data Science are: Data Visualization, Anaconda Jupyter Notebook, Exploratory Data Analysis, Machine Learning, Data wrangling, and Evaluation using scikit library’s surprise module.

Day - 01

**Aim) Enlist Python Operators and Keywords**

**Python Keywords:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| False | await | else | import | pass |
| none | except | break | continue | in |
| raise | True | class | finally | is |
| return | and | for | lambda | try |
| as | def | nonlocal | while | from |
| assert | del | global | if | with |
| async | elif | not | or | yield |

**Python Operators:**

1. **Arithmetic Operator**

|  |  |
| --- | --- |
| Operator | Meaning |
| + | Add two operands or unary plus |
| - | Subtract right operand from the left or unary minus |
| \* | Multiply two operands |
| / | Divide left operand by the right one (always results into float) |
| % | Modulus - remainder of the division of left operand by the right |
| // | Floor division - division that results into whole number adjusted to the left in the number line |
| \*\* | Exponent - left operand raised to the power of right |

1. **Comparison Operators**

|  |  |  |
| --- | --- | --- |
| Operator | Meaning |  |
| > | Greater than - True if left operand is greater than the right |  |
| < | Less than - True if left operand is less than the right |  |
| == | Equal to - True if both operands are equal |  |
| != | Not equal to - True if operands are not equal |  |
| >= | Greater than or equal to - True if left operand is greater than or equal to the right |  |
| <= | Less than or equal to - True if left operand is less than or equal to the right |  |

1. **Logical Operator**

|  |  |
| --- | --- |
| Operator | Meaning |
| and | True if both the operands are true |
| or | True if either of the operands is true |
| not | True if operand is false (complements the operand) |

1. **Identity Operator**

|  |  |
| --- | --- |
| Operator | Meaning |
| is | True if the operands are identical (refer to the same object) |
| is not | True if the operands are not identical (do not refer to the same object) |

1. **Membership Operator**

|  |  |
| --- | --- |
| Operator | Meaning |
| in | True if value/variable is found in the sequence |
| not in | True if value/variable is not found in the sequence |

**Aim) Explain Structured, Semi-Structured and Unstructured data.**

**Structured Data:**

Data with a high degree of organization, typically stored in a spreadsheet-like manner.

Examples:

* Excel spreadsheets
* Comma-separated value file (.csv)
* Relational database tables

**Unstructured Data:**

Unstructured data is data with no predefined organizational form and no specific format, so essentially everything which is not structured or semi-structured data.

Examples:

* Images such as .jpeg or .png files
* Videos such as .mp4 or m4a files
* Sound files such as .mp3 or .wav files

**Semi-Structured Data:**

Semi-structured data is data with some degree of organization.

Examples:

* Hypertext Markup Language (HTML) files
* JavaScript Object Notation (JSON) files
* Extensible Markup Language (XML) files

**Aim) Build Simple GST calculator with different categories and percentages**

**Code:**

def total(num):

  i = 0

  total = 0

  while i < num:

    price = int(input("Enter price:"))

    total += price

    i += 1

  return total

print("Category 1: Electronics")

print("Category 2: Home Appliances")

print("Category 3: Fashion")

c =  int(input("Choose your catagery"))

if c == 1:

  product = int(input("Enter no of products"))

  total = total(product)

  print(f"GST is {total \* 0.33}")

elif c == 2:

  product = int(input("Enter no of products"))

  total = total(product)

  print(f"GST is {total \* 0.30}")

elif c == 3:

  product = int(input("Enter no of products"))

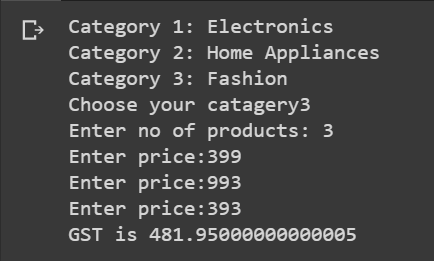
  total = total(product)

  print(f"GST is {total \* 0.27}")

else:

  print("Enter Correct choice")

**Output:**

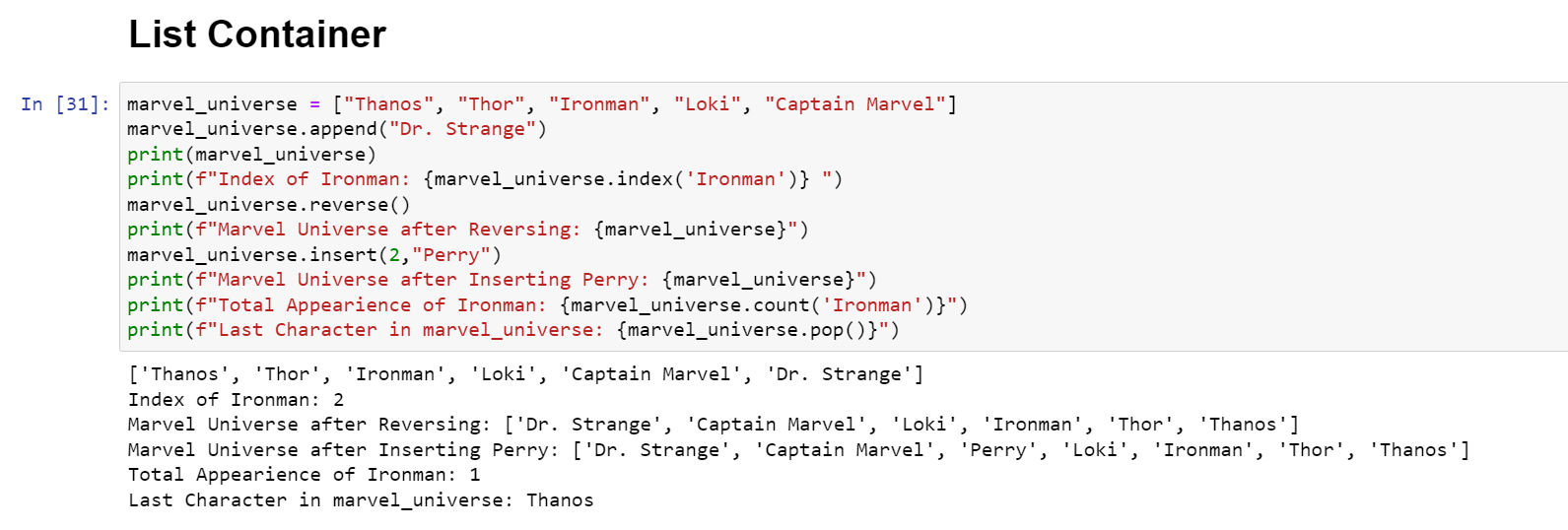
****

Day - 02

**Aim) List out five methods of List, Set and Dictionary with example.**

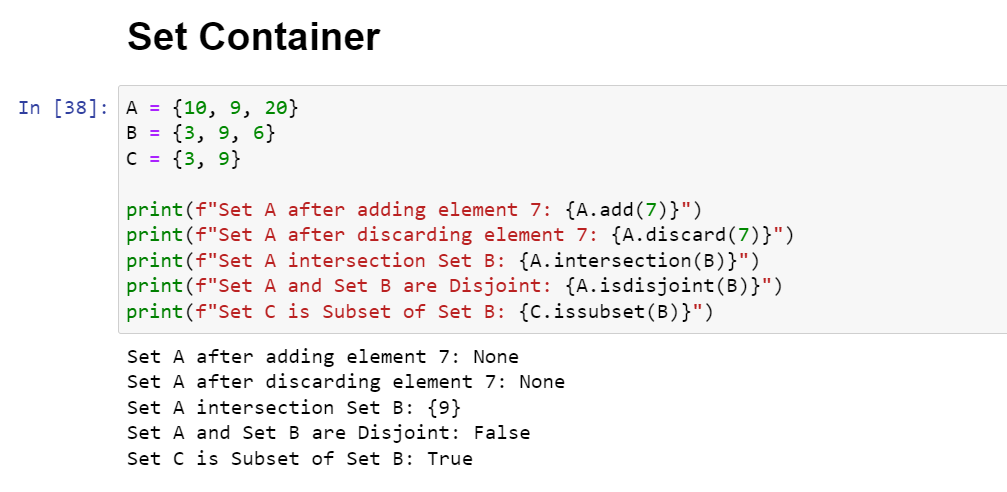
**List Methods:**

1. **append():** Adds an element at the end of the list
2. **count():** Returns the number of elements with the specified value
3. **pop():** Removes the element at the specified position
4. **reverse():** Reverses the order of the list
5. **insert():** Adds an element at the specified position
6. **index():** Returns the index of the first element with the specified value

****

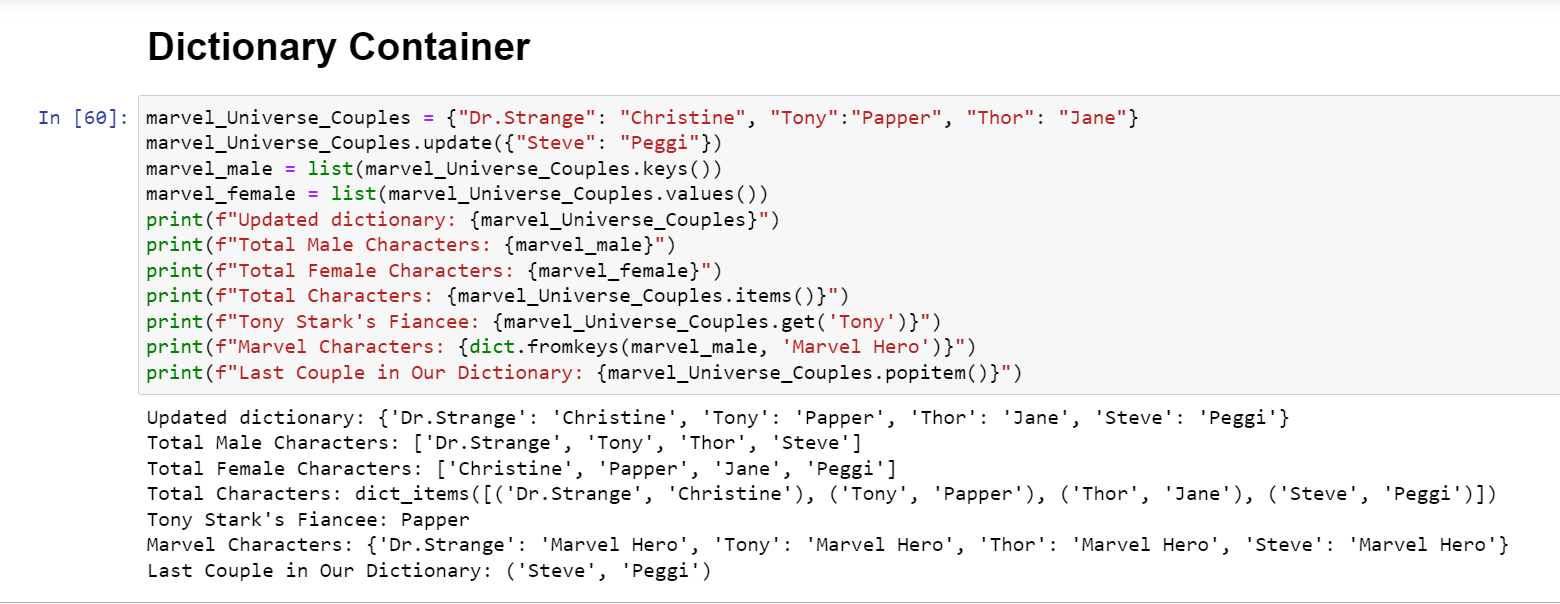
**Set Methods:**

1. **add():** Adds an element to the set
2. **discard():** Remove the specified item
3. **intersection():** Returns a set, that is the intersection of two or more sets
4. **issubset():** Returns whether another set contains this set or not
5. **isdisjoint()** Returns whether two sets have a intersection or not

****

**Dictionary Methods:**

1. **fromkeys():** Returns a dictionary with the specified keys and value
2. **get():** Returns the value of the specified key
3. **items():** Returns a list containing a tuple for each key value pair
4. **keys():** Returns a list containing the dictionary's keys
5. **values():** Returns a list of all the values in the dictionary
6. **update():** Updates the dictionary with the specified key-value pairs
7. **popitem():** Removes the last inserted key-value pair

****

**Aim) Make a program of Student Report card with take input from the user as Subject and input and return marks using function.**

**Code:**

def marksEntry(num):

    report = []

    total = 0

    subject\_range = int(input(f"Enter the number of subjects for Student {num + 1}: "))

    for number in range(subject\_range):

        subject = input("Enter Subject Name: ")

        marks = float(input("Enter Marks: "))

        total += marks

        report.append([subject,marks])

    report.append(["Total", total])

    return report

def report(num):

    student\_report = []

    for i in range(num):

        student\_report.append(marksEntry(i))

    return student\_report

number = int(input("Enter Number of students: "))

report\_list = report(number)

stu\_num = 0

for info in report\_list:

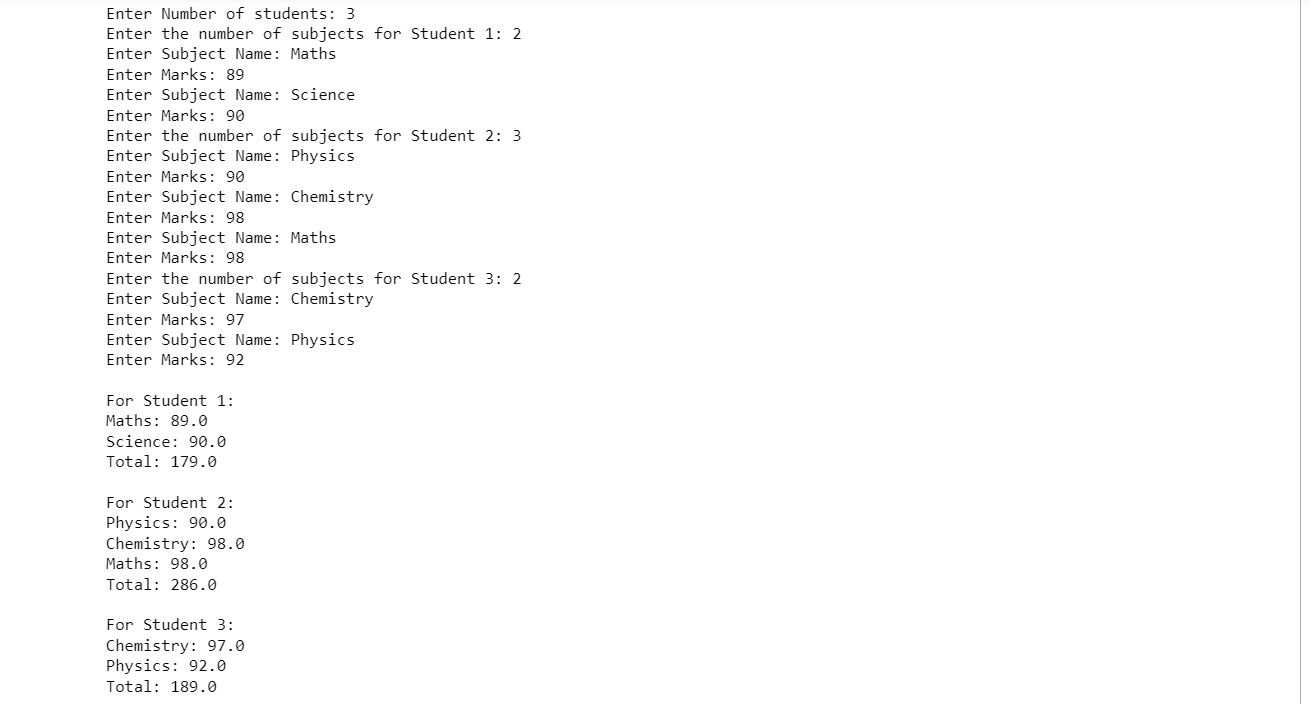
    stu\_num += 1

    print(f"\nFor Student {stu\_num}:")

    for num in info:

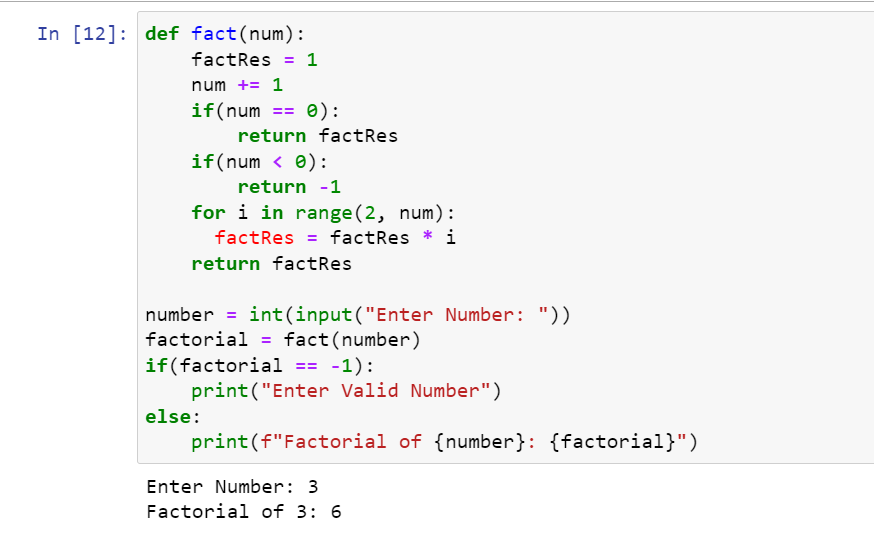
        print(f"{num[0]}: {num[1]}")

**Output:**

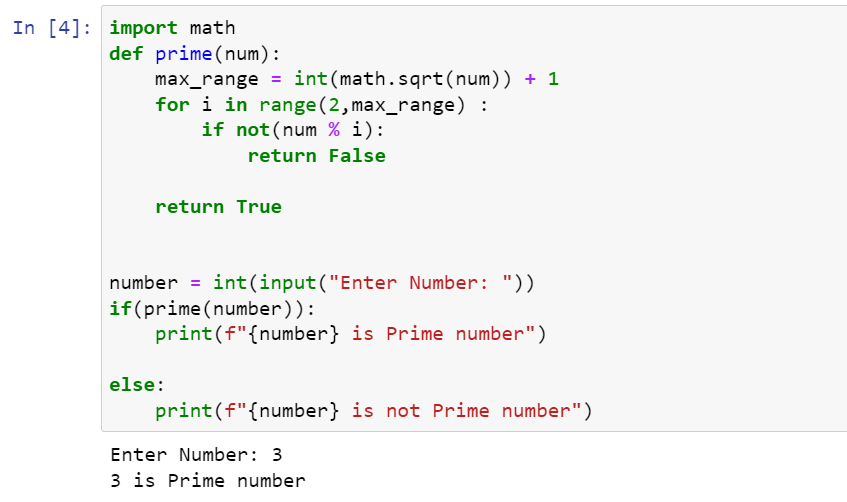
****

**Aim) Build a program to find factorial, even-odd number and prime number using function.**

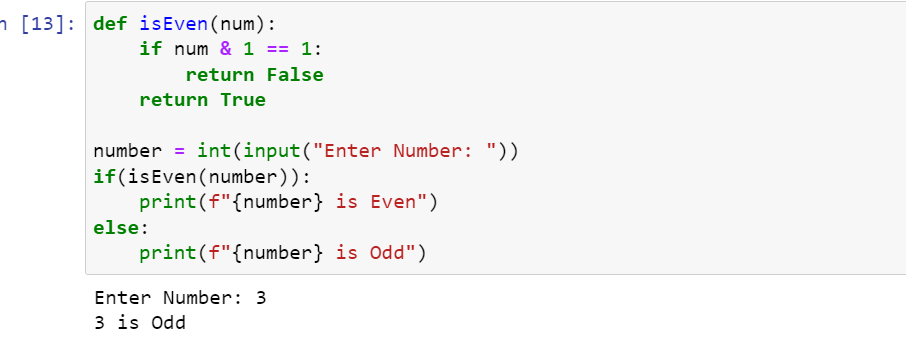
**Factorial:**

****

**Prime Number:**

****

**Even – Odd Number:**

****

Day - 03

**Aim) Built an ATM program containing Deposit and Withdraw functions with validations and loop using classes**

**Code:**

class ATM():

    def \_\_init\_\_(self, money):

        self.acc\_amount = money

    def enquiry(self):

        print(f"Account Balance: {self.acc\_amount}")

    def withDraw(self, amount):

        amount = int(amount)

        if(self.acc\_amount < amount):

            print("Insuffeicient Balance")

        else:

            self.acc\_amount -= amount

        self.enquiry()

    def deposit(self,amount):

        amount = int(amount)

        if(amount < 1):

            print("Enter Valid amount")

        else:

            self.acc\_amount += amount

            self.enquiry()

print("For ATM Card Facility you have to deposit atleast 5000 INR")

amount = int(input("Enter the Opeaning Account Amount: "))

if (amount >= 5000):

    Acc1 = ATM(amount)

    print("Enter 1 for Deposit the money")

    print("Enter 2 for Withdraw the money")

    while True:

        choice = int(input("Choice: "))

        if(choice == 1):

            amount = int(input("Deposit Amount: "))

            Acc1.deposit(amount)

        if(choice == 2):

            amount = int(input("Withdraw Amount: "))

            Acc1.withDraw(amount)

        else:

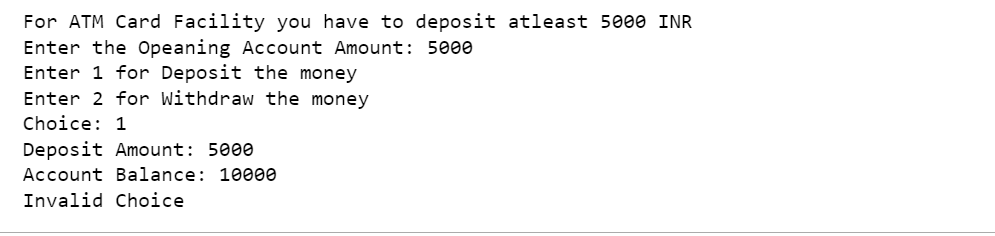
            print("Invalid Choice")

            break

else:

    print("Enter Amount above 5000")

**Output:**

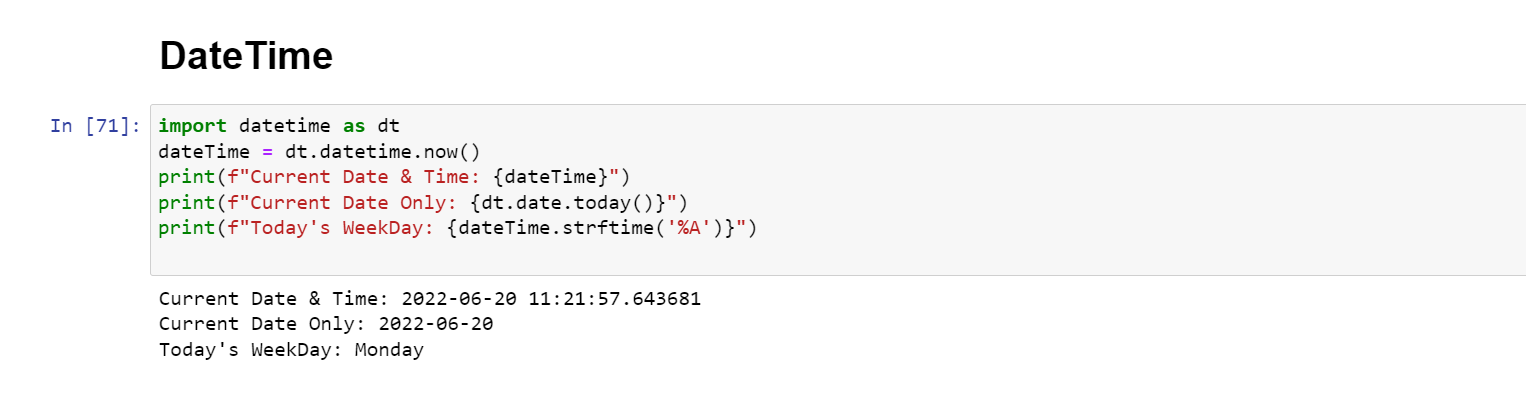
****

**Aim) List out 5 different inbuilt python libraries and use their 3 methods with example.**

1. **Math**

****

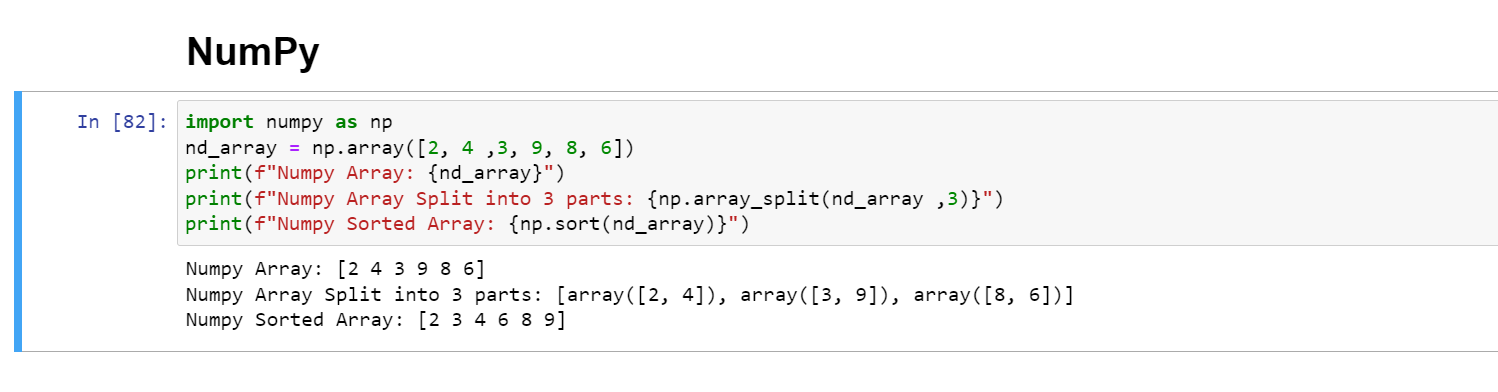
1. **DateTime**

****

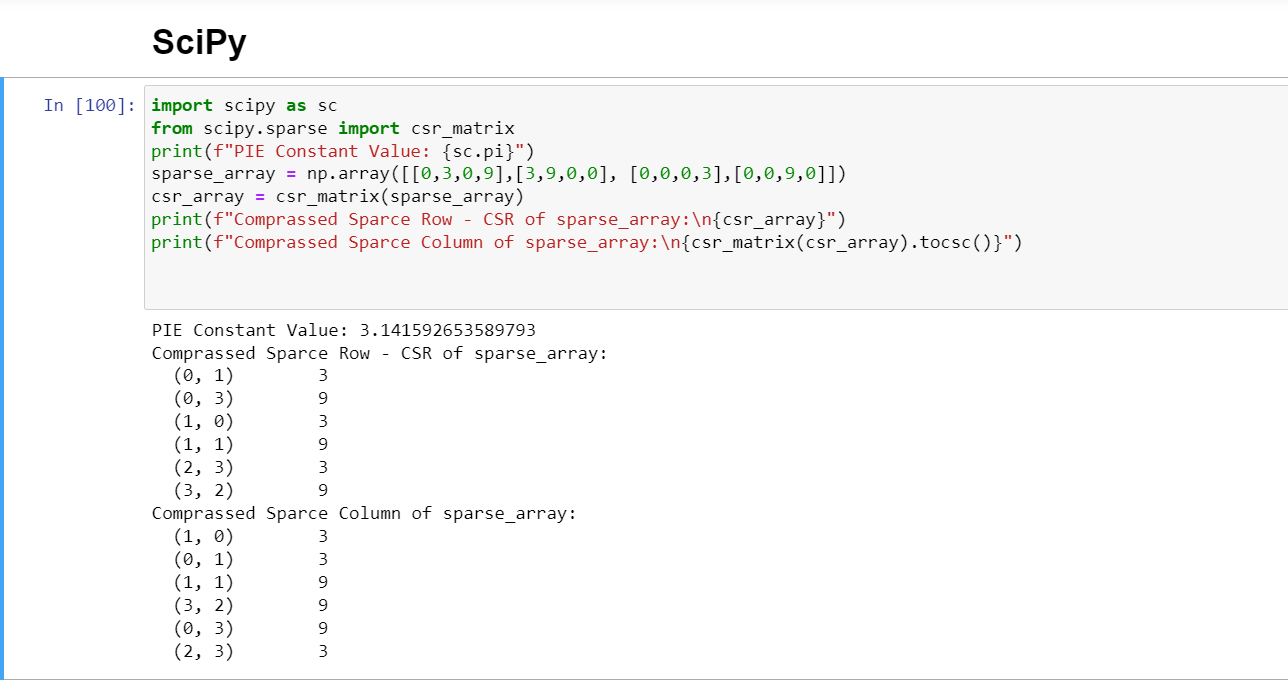
1. **Json**

****

1. **NumPy**

****

1. **SciPy**

****

**Aim) List out atleast 10 meta characters and use them in pattern for email and phone number validation**

**Meta Characters: These are characters with special meaning.**

1. **[ ] :** A set of Characters
2. **\ :** Signals a special sequence
3. **. :** Any character (except newline character)
4. **^ :** Starts with
5. **$ :** Ends with
6. **\* :** 0 or more occurrences
7. **+ :** 1 or more occurrences
8. **? :** 0 or 1 occurrences
9. **{ } :** Exactly the specified number of occurrences
10. **| :** Either or

**Email and Phone Validation using Regex:**

**Code:**

import re

#Email Regex Object made with re.compile() method

email\_regex = re.compile(r'([A-Za-z0-9]+[.-\_])\*[a-zA-Z0-9-]+(\.[a-z/A-Z]{2,3})+')

phone\_regex = re.compile(r'^[6-9]\d{9}$')

def isValidEmail(email):

    if re.fullmatch(email\_regex, email):

        print(f"Valid Email: {email}")

    else:

        print(f"Invalid Email: {email}")

def isValidPhone(phone):

    if re.fullmatch(phone\_regex, phone):

        print(f"Valid Phone Number: {phone}")

    else:

        print(f"Invalid Phone Number: {phone}")

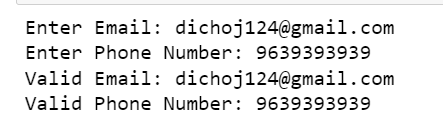
email = input("Enter Email: ")

phone = input("Enter Phone Number: ")

isValidEmail(email)

isValidPhone(phone)

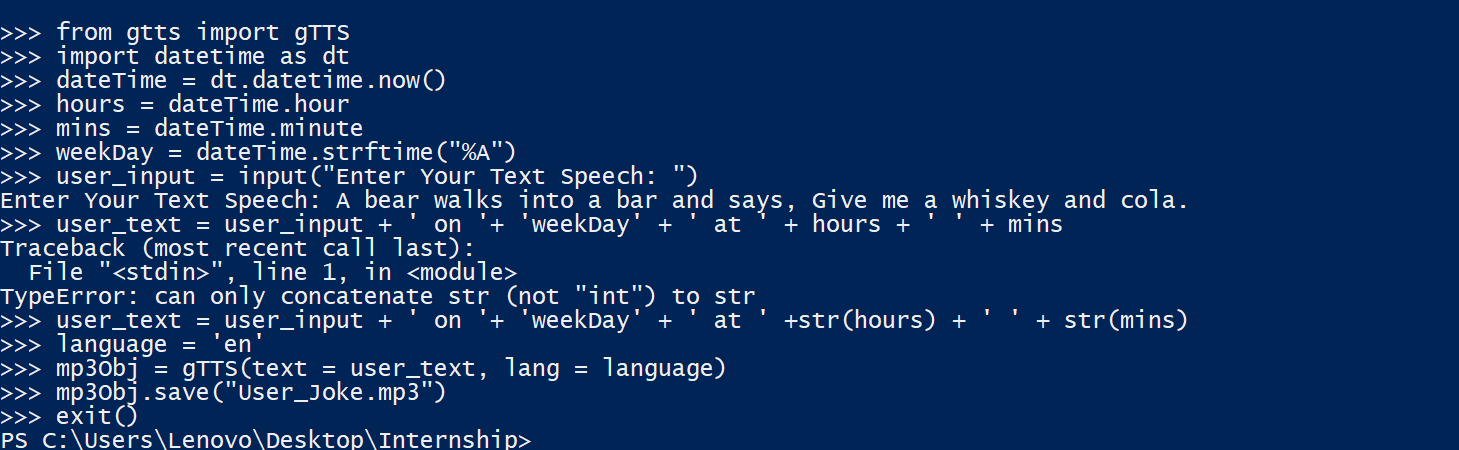
**Output:**

****

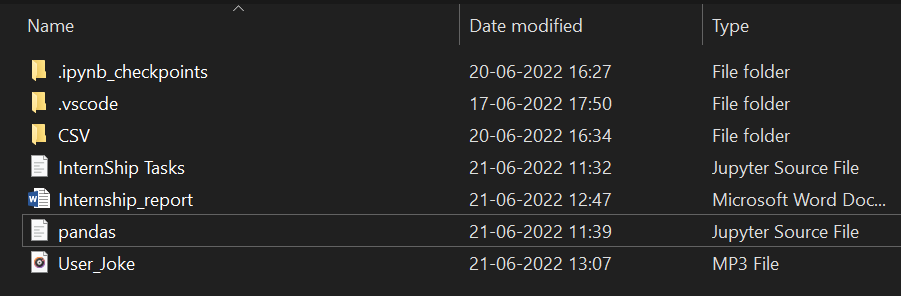
Day - 04

**Aim) Use and External library and In-built library in one python program with user input.**

**Code:**

****

**Output:**

****

**Aim) Explain PANDAS application and List out at least 5 methods of it with example**

**Applications of Pandas:**

* **Economics:**
* Analyzing data to form patterns and understanding trends about how the economy in various sector is growing, is something essential for economist.
* Pandas provide tools for analyzing large datasets like datasets and file handling.
* These tools help economists to immensely in accessing and manipulating data to get desired output.
* **Recommendation Systems:**
* Spotify or Netflix provides brilliant recommendations with the help of the DEEP LEARNING and it is one of the most important application of PANDAS.
* As the Recommending Systems is possible only by learning and handling huge datasets, functions like groupBy and mapping help tremendously in making these system possible.
* **Big Data:**
* PANDAS has a good connection with HADOOP and SPARK, allowing PANDAS to have access Big Data.
* **Data Science:**
* PANDAS and Data Science are almost synonymous as almost all application of PANDAS falls under the scope of Data Science.
* **NLP – Natural Language Processing:**
* NLP has taken the word by a storm and it is creating a lot of buzzes.
* PANDAS and SCIKIT-LEARN provides functionalities to easily create NLP models which can be improved continuously.
* **Neuroscience:**
* Data manipulation capabilities of PANDAS have played major role in compiling a huge amount of data which has helped neuroscientists in understanding trends that are followed inside our bodies and the effects of various things on our entire nervous system.

**PANDAS Methods:**

**1) read\_csv(url):**

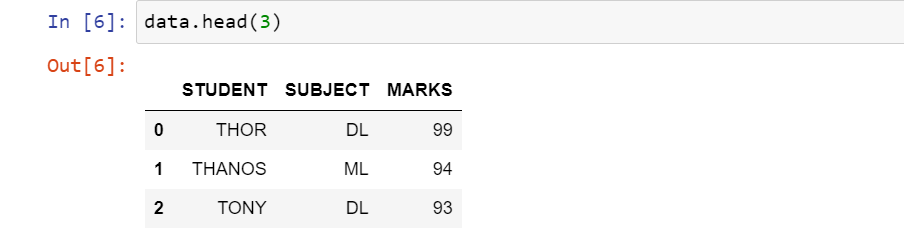
This function helps to read Comma Separated Values (CSV) files into a PANDAS DataFrame object.



Function to\_csv(name) works exactly opposite of it. It converts PANDAS DataFrame or Series to a CSV files.

**2) head() or tail():**

Function head(number) is used to return the first number rows of a dataset. By default .head() returns first 5 rows of the DataFrame.

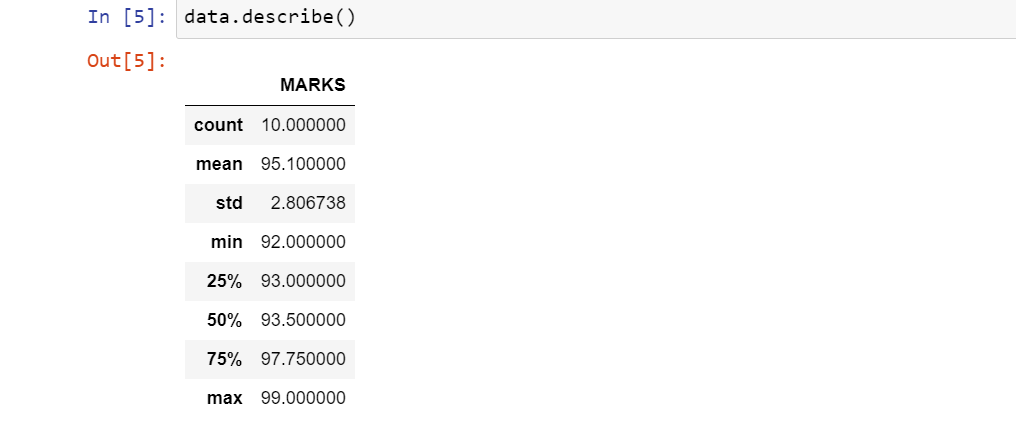


Function tail(number) is used to return the last number rows of a dataset

**3) describe():**

Function describe() is used to generate descriptive statistics of the data in a PANDAS DataFrame or Series.

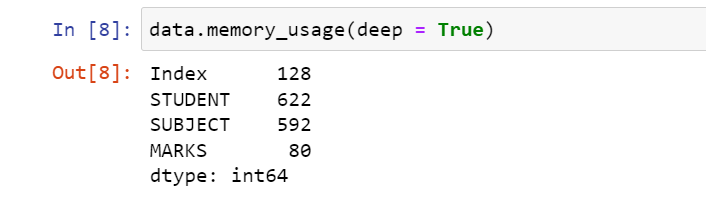
It summarizes central tendency and dispersion of the dataset.



**4) memory\_usage()**

Function memory\_usage() returns PANDAS Series having the memory usage of each column in a PANDAS DataFrame in bytes.

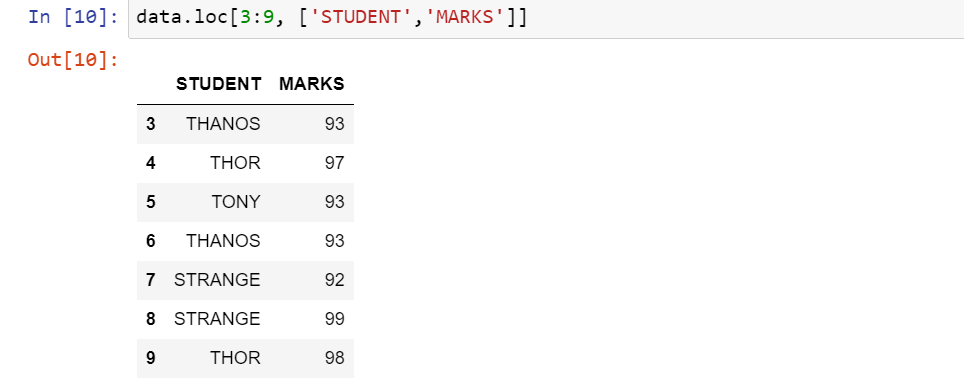
By specifying the deep attribute as True, we can get to know the actual space being taken by each column.



**5) loc[ : ] or iloc[ : ]**

Function loc[:] helps to access a group of rows and columns in a dataset by slicing the dataset.

We can access rows and columns based on labels.



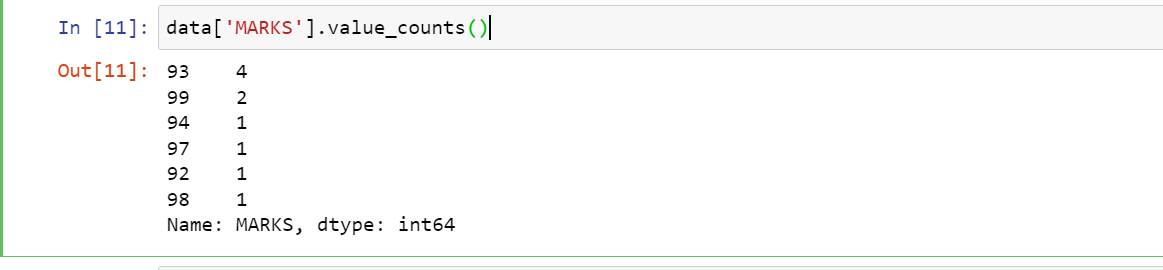
Here we select rows labelled 3 to 9 and the column labelled STUDENT and MARKS.

Function iloc[ : ] is used in same manner but with indexes and it is not inclusive on last value.

**6) value\_counts()**

Function value\_counts() returns PANDAS Series containing the counts of unique values.

It is used to identify the number of occurrences of each unique value in a Series.

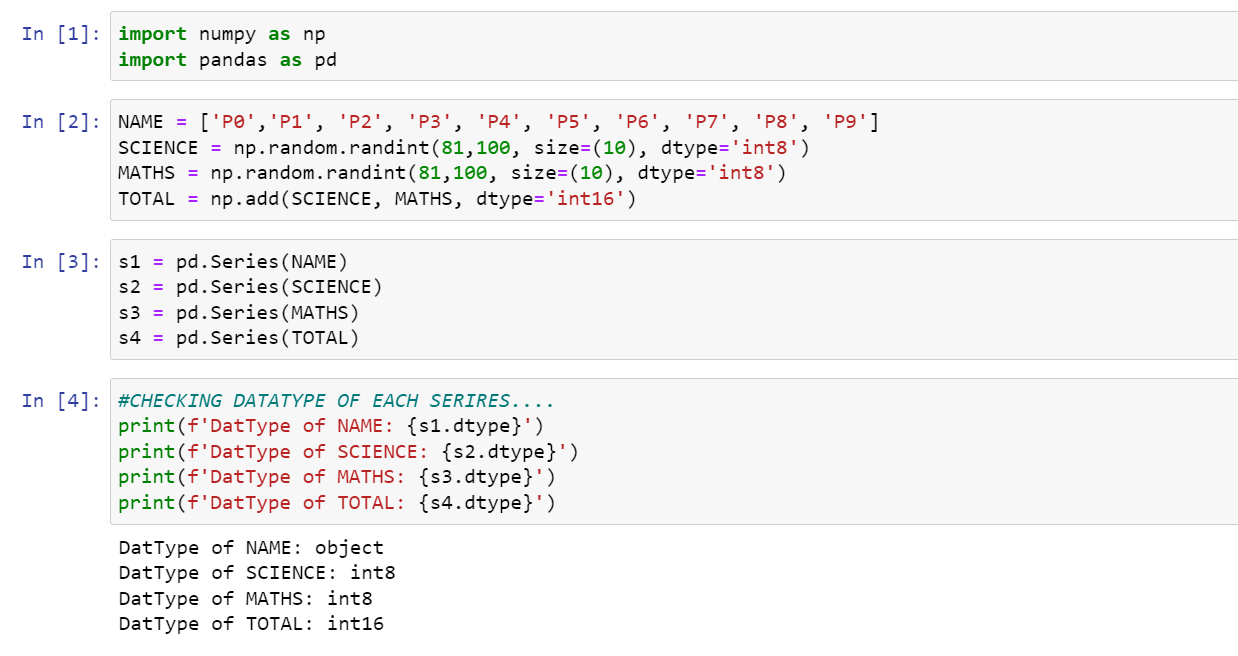


Day - 05

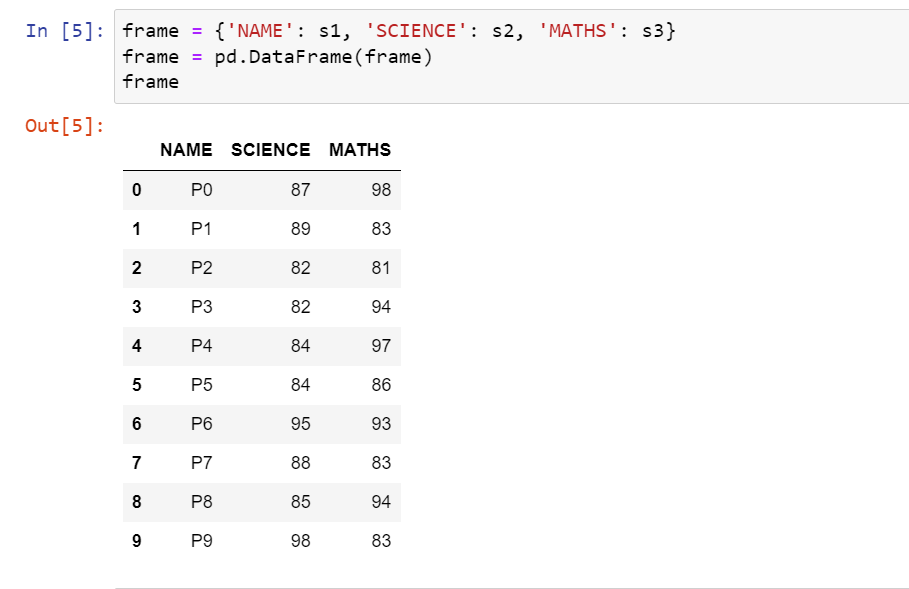
**Aim) Convert Multiple Series into PANDAS DataFrame, and find the Shape and data types of each column.**

**Code:**

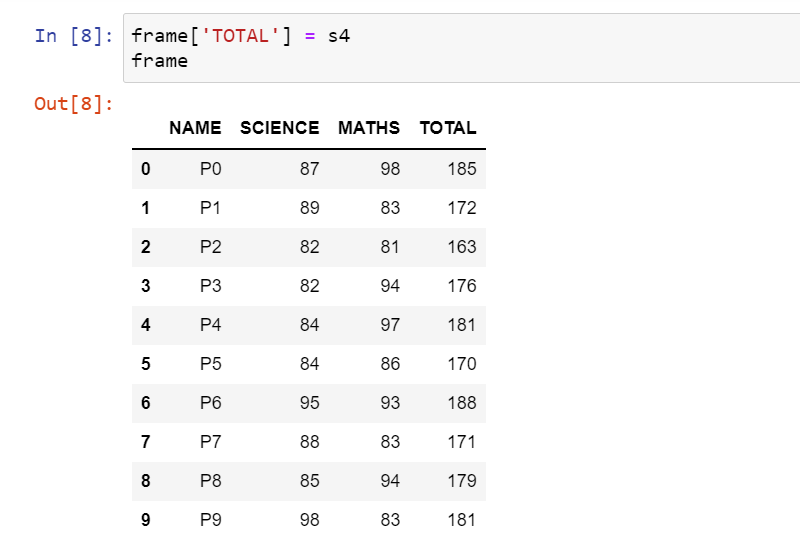
1. Creating Series and checking its Data Type



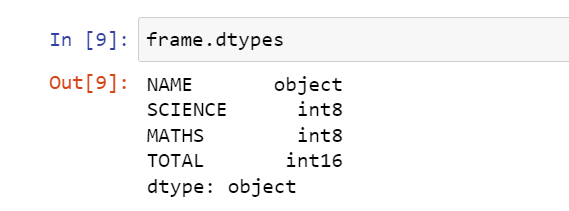
1. Create DataFrame using Multiple Series



1. Adding one Series into existing DataFrame



1. Checking data types of the columns of DataFrame



Day - 06

**Aim) List out and explain atleast five functionalities of matplotlib and numpy with example.**

Numpy:

1) Powerful N – Dimensional Arrays

Fast and versetile, the Numpy vectorization, indexing, and broadcasting concepts are the defacto standards of array computing today.

2) Numerical Computing Tools

Numpy offers comprehensive mathematical functions, random number generators, linear algebra and more.

3) Interoperable

Numpy supports a wide range of hardware and computing platforms, and plays well with distributed, GPU and sparse array libraries.

4) Performant

The core of Numpy is well optimized C code. We enjoy the flexibility of Python with the speed of compiled code.

5) Easy to use

Numpy’s high level syntax makes it accessible and productive for programmers from any background or experience level.

Matplotlib:

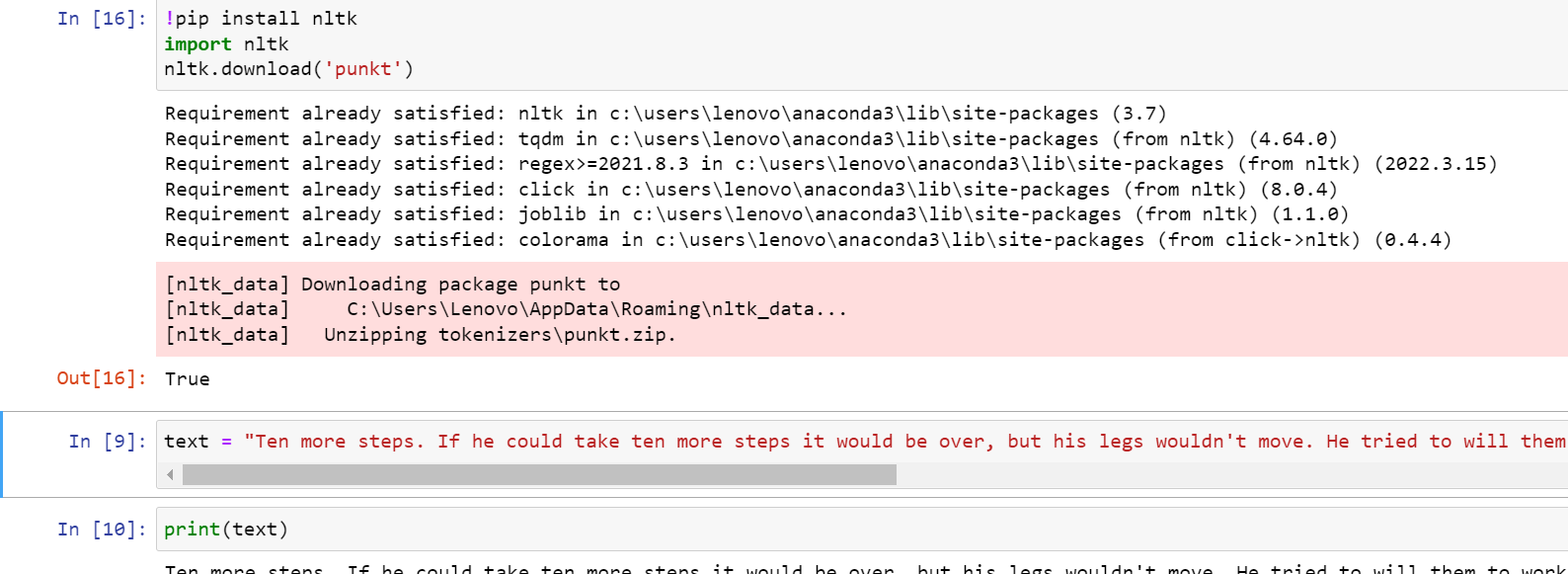
1) Matplotlib is a cross platform, data visualization and graphical plotting library for python and it’s numerical extension Numpy.

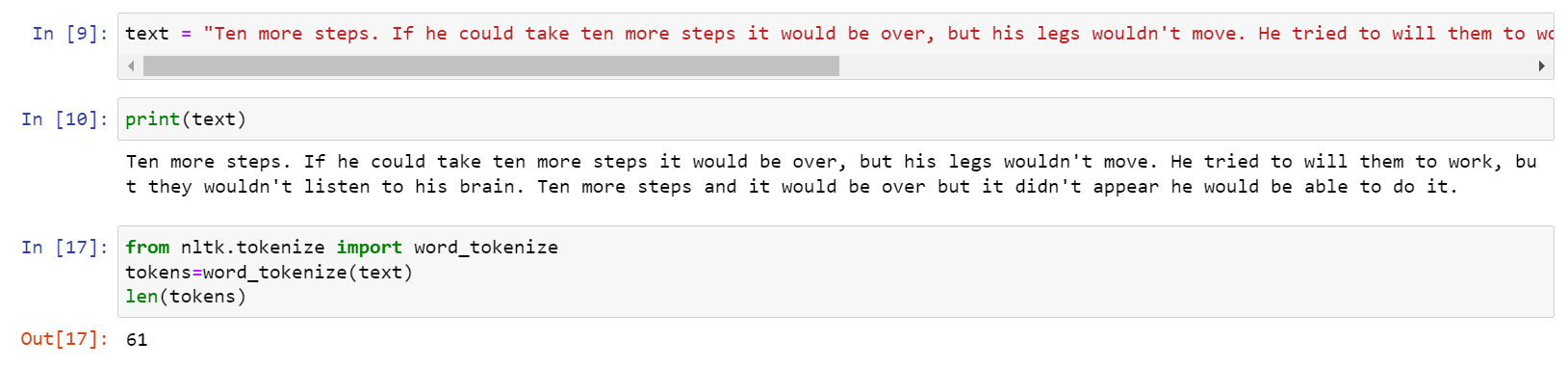
2) It offers a viable open source alternative to MATLAB.

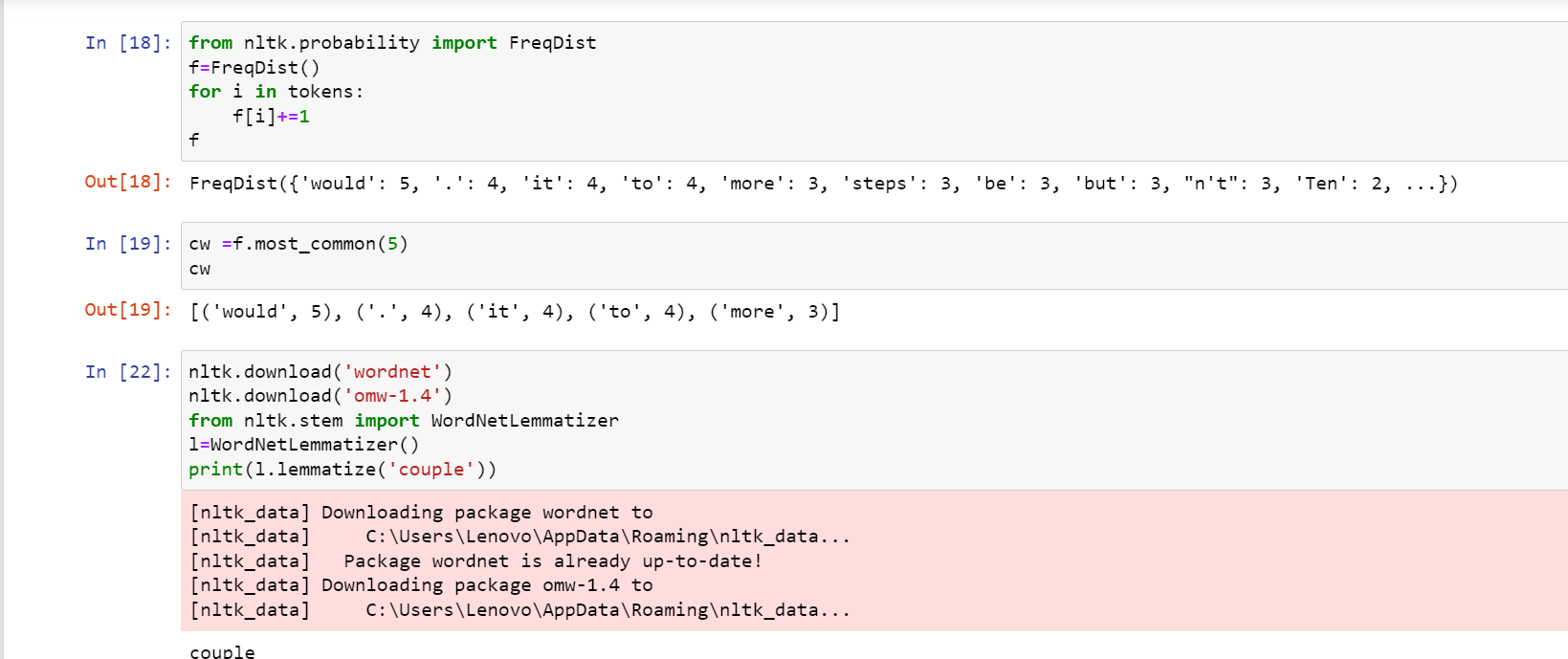
3) Developers use matplotlib’s APIs to embed plots in GUI applications.

Day - 07

**Aim) Download and use the nltk packages and corpus data with example.**









Day - 08

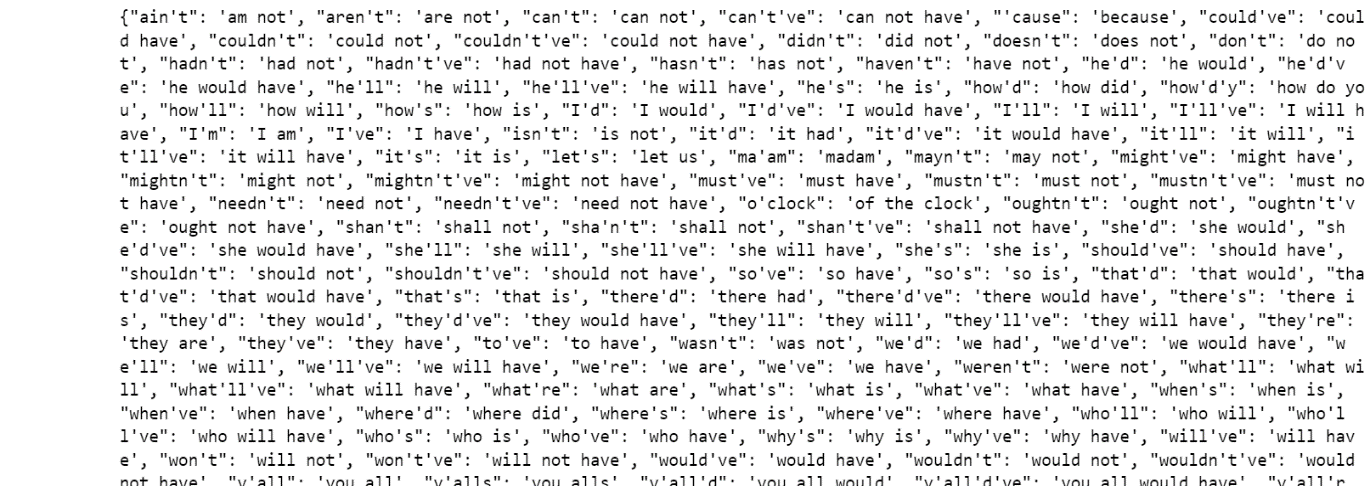
**Aim) Load data from JSON file and find total word count and sentences count from that.**

import json

with open('contractions.json') as file:

    data = json.load(file)

print(data)



Total Number of word and sentences.

words= nltk.word\_tokenize(str(data))

len(words)

sentences= data.keys()

len(sentences)



Day - 09

**Aim) Use label encoding on userId and productId in dataset**

**Code:**

from sklearn.preprocessing import LabelEncoder

label = LabelEncoder()

data['UserID'] = label.fit\_transform(data['UserID'])

data['ProductID'] = label.fit\_transform(data['ProductID'])

data

**Output:**

****

**Aim) Generate a new score by merging the rating and sentimental score.**

Code:

from textblob import TextBlob

def sentiment(text):

    try:

        return TextBlob(str(text)).sentiment.polarity

    except:

        return None

data['Sentiment'] = data['Text'].apply(sentiment)

data['Updated\_Score'] = data['Rating'] \* data['Sentiment']

Output:

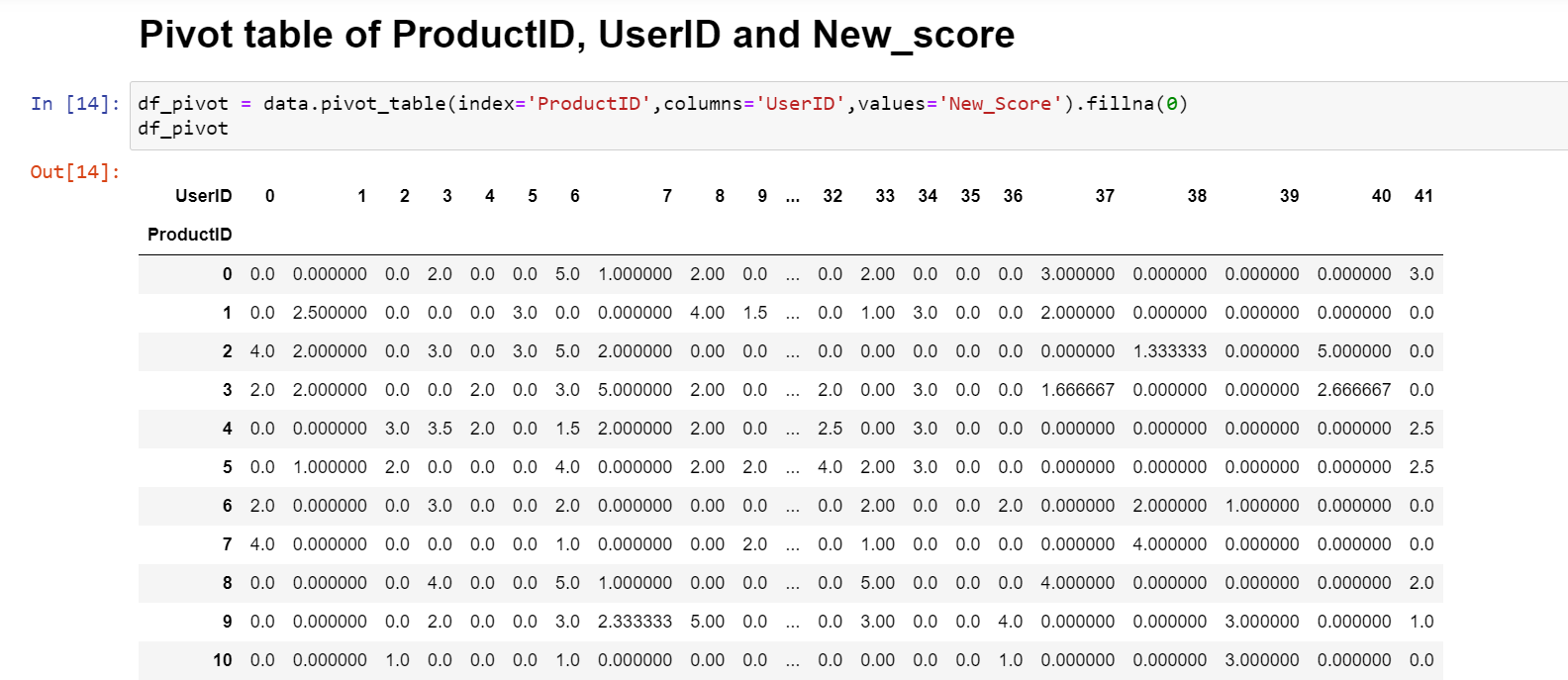


Day - 10

**Aim: Find the cosine similarity between products**

df\_pivot = data.pivot\_table(index='ProductID',columns='UserID',values='New\_Score').fillna(0)

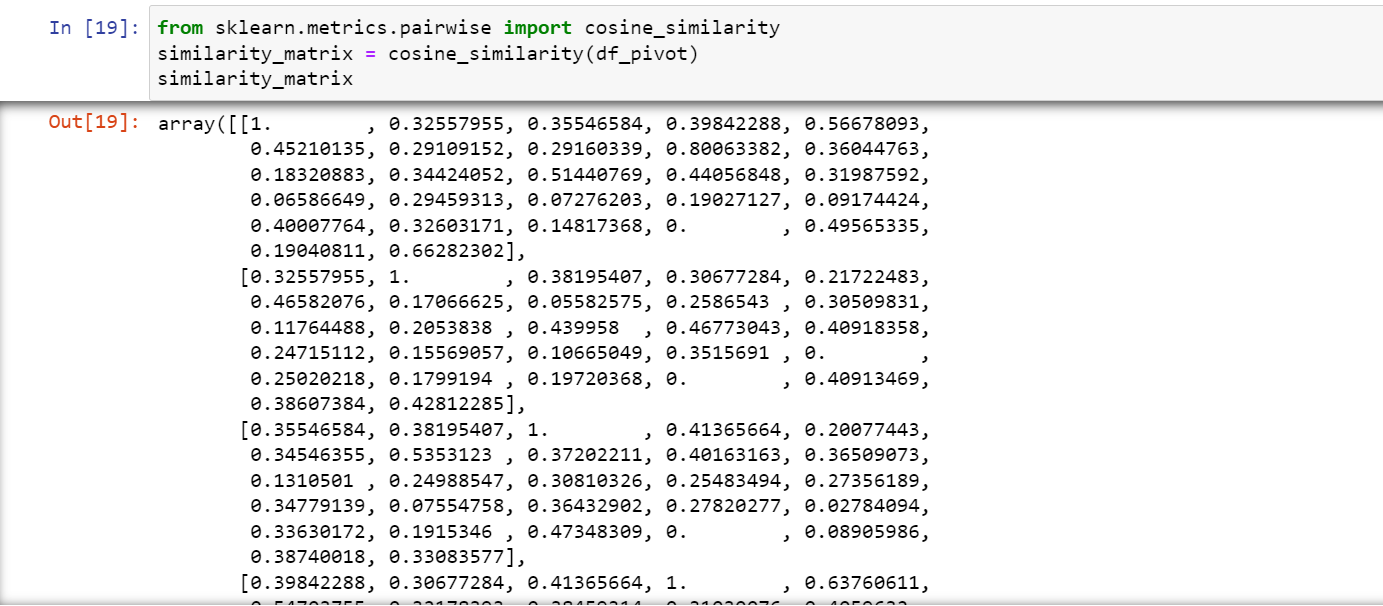
df\_pivot

****

from scipy.sparse import csr\_matrix

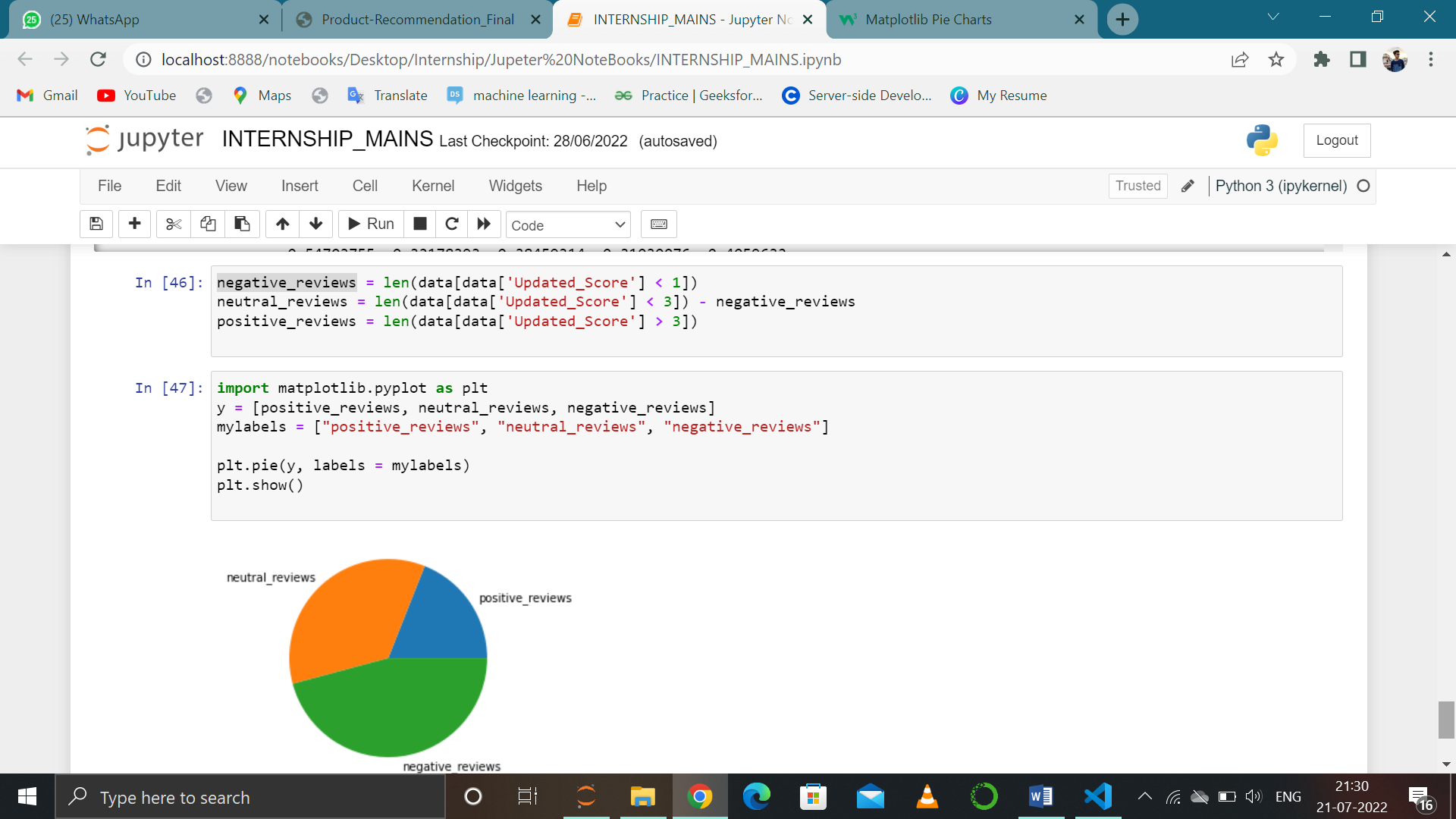
df\_pivot\_matrix = csr\_matrix(df\_pivot.values)

print(df\_pivot\_matrix)



Day - 11

**Aim: Build chart of sentimental analysis containing percentage of positive, negative and neutral reviews.**



Day - 12

**Aim: Convert Recommandations in DataFrame containing productand their distance.**



Day - 14

**Aim: Explain radius and n\_neighbours of KNN.**

The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. While it can be used for either regression or classification problems, it is typically used as a classification algorithm, working off the assumption that similar points can be found near one another.

To recap, the goal of the k-nearest neighbor algorithm is to identify the nearest neighbors of a given query point, so that we can assign a class label to that point. In order to do this, KNN has a few requirements:

Determine your distance metrics

In order to determine which data points are closest to a given query point, the distance between the query point and the other data points will need to be calculated. These distance metrics help to form decision boundaries, which partitions query points into different regions.

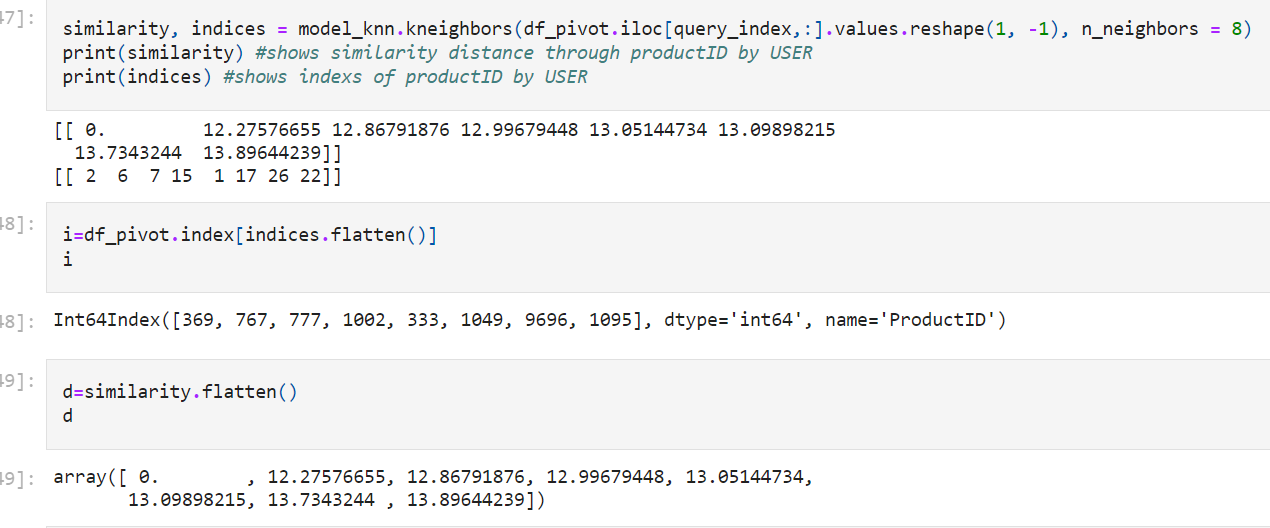
Euclidean distance (p=2):

This is the most commonly used distance measure, and it is limited to real-valued vectors. Using the below formula, it measures a straight line between the query point and the other point being measured.

Day - 15

**Aim: Convert the recommendation into dataframe containing product and their distances.**

****

****

