

Question 1:

- The following is a list of 10 students ages: ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24]
- Sort the list and find the min and max age
- Add the min age and the max age again to the list
- Find the median age (one middle item or two middle items divided by two)
- Find the average age (sum of all items divided by their number)
- Find the range of the ages (max minus min)

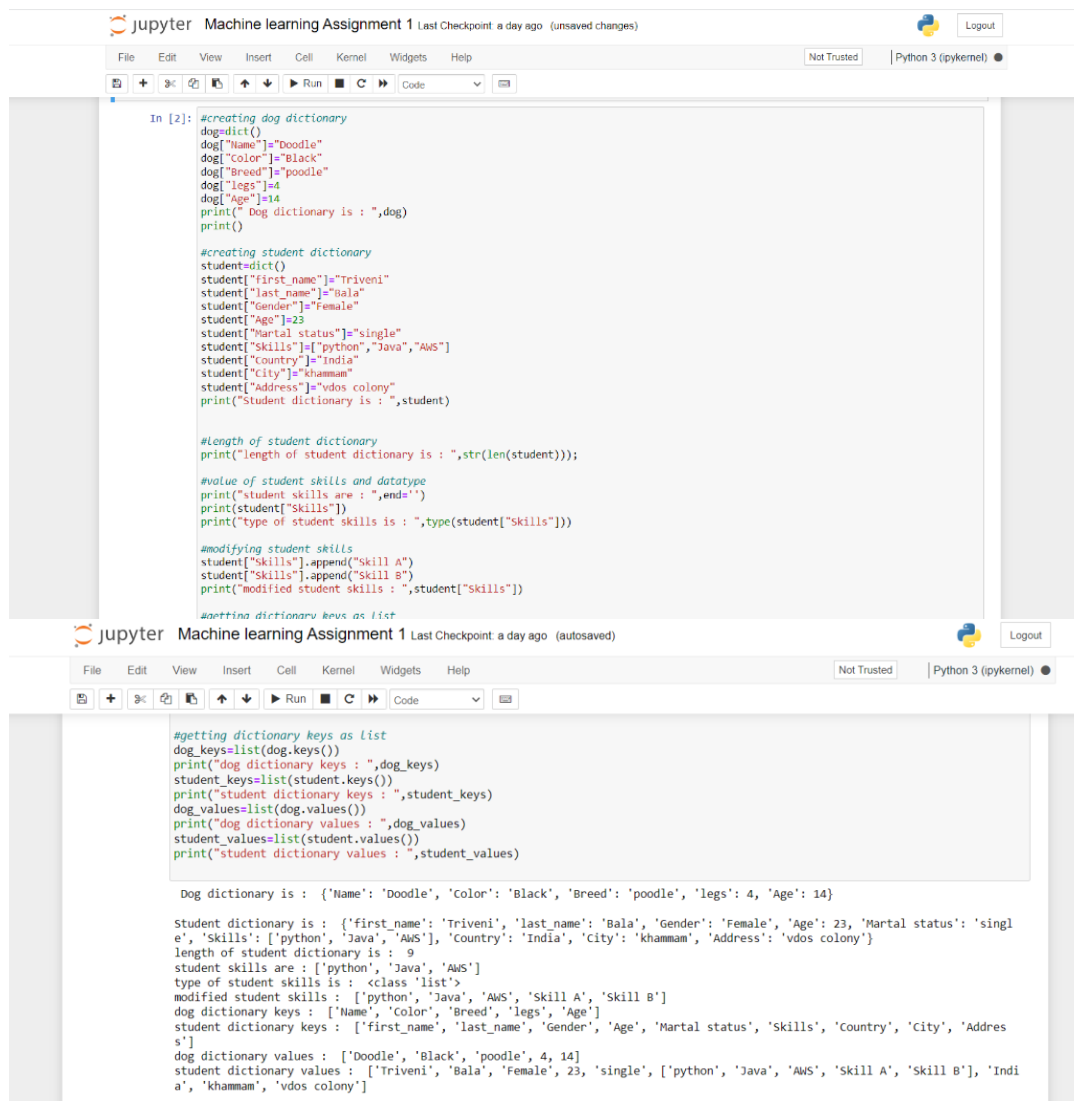
```
In [8]: ages = [19,22,19,24,20,25,26,24,25,24]
ages.sort()
minAge = min(ages)
maxAge = max(ages)
ages.append(minAge)
ages.append(maxAge)
ages.sort()
length = len(ages)
medianAge = ages[ (length+1)//2] if length%2==1 else (ages[length//2]+ages[ (length//2)+1])/2
sumofAges = sum(ages)
averageofAges = sumofAges//length
rangeofAges = maxAge-minAge
print("Sorted list: ", ages)
print("Min age:", minAge)
print("Max age :", maxAge)
print("Median :", medianAge)
print("Average :", averageofAges)
print("Range:", rangeofAges)

Sorted list: [19, 19, 19, 20, 22, 24, 24, 24, 25, 25, 26, 26]
Min age: 19
Max age : 26
Median : 24.0
Average : 22
Range: 7
```

The list is sorted using the sort() method. min() and max() are used to calculate the Maximum and minimum values. By using sum() and len() methods we have calculated the median Average values. Append method is using to add minimum and maximum values to the list .The range is difference between the maximum and maximum values.

Question 2 :

- Create an empty dictionary called dog
- Add name, color, breed, legs, age to the dog dictionary
- Create a student dictionary and add first_name, last_name, gender, age, marital status, skills, country, city and address as keys for the dictionary
- Get the length of the student dictionary
- Get the value of skills and check the data type, it should be a list
- Modify the skills values by adding one or two skills
- Get the dictionary keys as a list
- Get the dictionary values as a list



The image displays two screenshots of a Jupyter Notebook interface. The top screenshot shows the initial code for creating and printing two dictionaries. The bottom screenshot shows the same code with additional operations: converting dictionary keys and values to lists, and printing the results.

```
In [2]: #creating dog dictionary
dog=dict()
dog["Name"]="Doodle"
dog["Color"]="Black"
dog["Breed"]="poodle"
dog["legs"]=4
dog["Age"]=14
print(" Dog dictionary is : ",dog)
print()

#creating student dictionary
student=dict()
student["first_name"]="Triveni"
student["last_name"]="Bala"
student["Gender"]="Female"
student["Age"]=23
student["Martial status"]="single"
student["Skills"]=["python","Java","AWS"]
student["Country"]="India"
student["City"]="khammam"
student["Address"]="vdos colony"
print("Student dictionary is : ",student)

#Length of student dictionary
print("length of student dictionary is : ",str(len(student)));

#value of student skills and datatype
print("student skills are : ",end='')
print(student["Skills"])
print("type of student skills is : ",type(student["Skills"]))

#modifying student skills
student["Skills"].append("Skill A")
student["Skills"].append("Skill B")
print("modified student skills : ",student["Skills"])

#getting dictionary keys as list
dog_keys=list(dog.keys())
print("dog dictionary keys : ",dog_keys)
student_keys=list(student.keys())
print("student dictionary keys : ",student_keys)
dog_values=list(dog.values())
print("dog dictionary values : ",dog_values)
student_values=list(student.values())
print("student dictionary values : ",student_values)
```

Dog dictionary is : {'Name': 'Doodle', 'Color': 'Black', 'Breed': 'poodle', 'legs': 4, 'Age': 14}

Student dictionary is : {'first_name': 'Triveni', 'last_name': 'Bala', 'Gender': 'Female', 'Age': 23, 'Martial status': 'single', 'Skills': ['python', 'Java', 'AWS'], 'Country': 'India', 'City': 'khammam', 'Address': 'vdos colony'}

length of student dictionary is : 9

student skills are : ['python', 'Java', 'AWS']

type of student skills is : <class 'list'>

modified student skills : ['python', 'Java', 'AWS', 'Skill A', 'Skill B']

dog dictionary keys : ['Name', 'Color', 'Breed', 'legs', 'Age']

student dictionary keys : ['first_name', 'last_name', 'Gender', 'Age', 'Martial status', 'Skills', 'Country', 'City', 'Address']

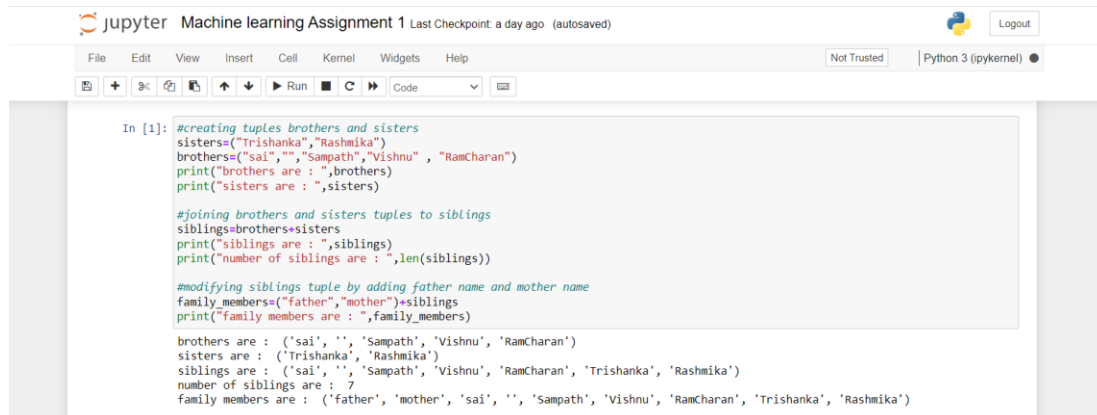
dog dictionary values : ['Doodle', 'Black', 'poodle', 4, 14]

student dictionary values : ['Triveni', 'Bala', 'Female', 23, 'single', ['python', 'Java', 'AWS', 'Skill A', 'Skill B'], 'India', 'khammam', 'vdos colony']

Empty dictionaries 'dog' and 'student' are created using dict() .data is initialized using key and value The len() method is used to find the length of the student dictionary. The value of skills is accessed using indexing and the type() method is used to find its datatype. Keys and values methods are used to find the keys and values of the student dictionary.

Question 3:

- Create a tuple containing names of your sisters and your brothers (imaginary siblings are fine)
- Join brothers and sisters tuples and assign it to siblings
- How many siblings do you have?
- Modify the siblings tuple and add the name of your father and mother and assign it to family_members
-



The screenshot shows a Jupyter Notebook titled "Machine learning Assignment 1" with a "Last Checkpoint: a day ago (autosaved)" status. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and code execution. The code cell contains the following Python code:

```
In [1]: #creating tuples brothers and sisters
sisters=('Trishanka','Rashmika')
brothers=('sai','','Sampath','Vishnu', 'RamCharan')
print("brothers are :",brothers)
print("sisters are :",sisters)

#joining brothers and sisters tuples to siblings
siblings=brothers+sisters
print("siblings are :",siblings)
print("number of siblings are :",len(siblings))

#modifying siblings tuple by adding father name and mother name
family_members=('father','mother')+siblings
print("family members are :",family_members)

brothers are : ('sai', '', 'Sampath', 'Vishnu', 'RamCharan')
sisters are : ('Trishanka', 'Rashmika')
siblings are : ('sai', '', 'Sampath', 'Vishnu', 'RamCharan', 'Trishanka', 'Rashmika')
number of siblings are : 7
family members are : ('father', 'mother', 'sai', '', 'Sampath', 'Vishnu', 'RamCharan', 'Trishanka', 'Rashmika')
```

Two tuples are created naming sisters and brothers. Both tuples are joined together using a + sign and stored in the siblings. Len() method is used to find the number of siblings are there. A new tuple with 'parents' names is created and combined with siblings and stored in the family_members tuple.

Question 4:

it_companies = {'Facebook', 'Google', 'Microsoft', 'Apple', 'IBM', 'Oracle', 'Amazon'} A = {19, 22, 24, 20, 25, 26} B = {19, 22, 20, 25, 26, 24, 28, 27} age = [22, 19, 24, 25, 26, 24, 25, 24]

- Find the length of the set it_companies
- Add 'Twitter' to it_companies
- Insert multiple IT companies at once to the set it_companies
- Remove one of the companies from the set it_companies
- What is the difference between remove and discard
- Join A and B
- Find A intersection B
 - Is A subset of B
- Are A and B disjoint sets
- Join A with B and B with A
- What is the symmetric difference between A and B
- Delete the sets completely
- Convert the ages to a set and compare the length of the list and the set.

```
jupyter Machine learning Assignment 1 Last Checkpoint a day ago (autosaved) Logout
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

In [2]: # given list of IT companies
it_companies = {"facebook", "Google", "Microsoft", "Apple", "IBM", "Oracle", "Amazon"}
A = {19, 22, 24, 20, 25, 26}
B = {19, 22, 20, 25, 26, 24, 28, 27}
age = [22, 19, 24, 25, 26, 24, 25, 24]
print("length of it_companies : ", len(it_companies))

# adding twitter to it_companies
it_companies.add("Twitter")
print("it_companies after appending twitter :", it_companies)

# inserting multiple it companies
companies = ["NCR", "Wipro", "TCS"]
it_companies.update(companies)
print("it_companies after appending multiple companies :", it_companies)

# removing one company
it_companies.remove("Twitter")
print("it_companies after removing one company : ", it_companies)

# Difference between remove and discard method
# The remove() method will raise an error if the specified item does not exist, and the discard() method will not raise error.

# joining A and B
C = A.union(B)
print("joining A and B gives :", C)

# finding A intersection B
I = A.intersection(B)
print("Intersection of A and B is:", I)

# checking is A subset of B
check = A.issubset(B)
if check:
    print("A is subset of B")
else:
    print("A is not a subset of B")

# checking are A and B are disjoint sets
check1 = A.isdisjoint(B)
if check1:
    print("A and B are disjoint sets")
else:
    print("A and B are not disjoint sets")

# joining A with B and B with A
A_join_B = A.union(B)
B_join_A = B.union(A)
print("A join B is :", A_join_B)
print("B join A is :", B_join_A)

# symmetric difference between A and B
D = A.symmetric_difference(B)
print("symmetric difference between A and B is :", D)

# deleting all the sets
it_companies.clear()
A.clear()
B.clear()

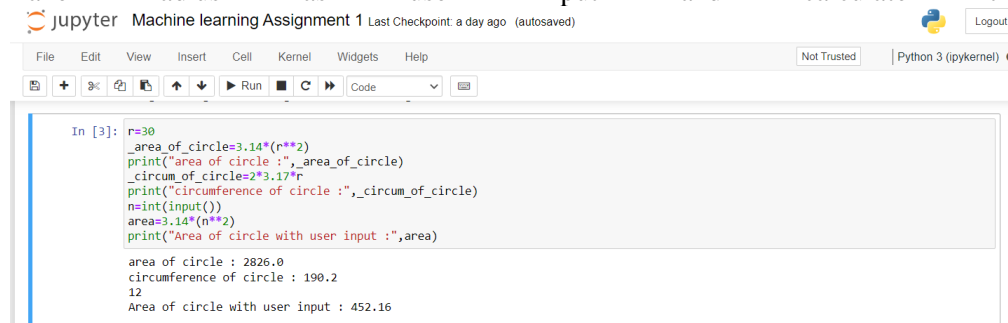
# converting ages list to set
age_set = set(age)
print("length of ages list is : ", len(age))
print("length of ages set is : ", len(age_set))
print("length of ages list is greater than the ages set")

length of it_companies : 7
it_companies after appending twitter : {'Google', 'Twitter', 'Facebook', 'Oracle', 'Apple', 'Microsoft', 'Amazon', 'IBM'}
it_companies after appending multiple companies : {'Oracle', 'Wipro', 'Amazon', 'Apple', 'TCS', 'Google', 'Twitter', 'Facebook', 'IBM', 'NCR', 'Microsoft'}
it_companies after removing one company : {'Oracle', 'Wipro', 'Amazon', 'Apple', 'TCS', 'Google', 'Facebook', 'IBM', 'NCR', 'Microsoft'}
joining A and B gives : {19, 20, 22, 24, 25, 26, 27, 28}
Intersection of A and B is: {19, 20, 22, 24, 25, 26}
A is subset of B
A and B are not disjoint sets
A join B is : {19, 20, 22, 24, 25, 26, 27, 28}
B join A is : {19, 20, 22, 24, 25, 26, 27, 28}
symmetric difference between A and B is : {27, 28}
length of ages list is : 8
length of ages set is : 5
length of ages list is greater than the ages set
```

Length of the set is calculated using the len() method. A value is added to the set it_companies using add() method and multiple companies are added using the update() method. An item is removed using the remove() method. Upon trying to remove the deleted item using the remove method again raises an error while using the discard method does not raise an error. Methods of sets like update, intersection, issubset, isdisjoint, union, symmetric_difference are used to perform operations on sets. The clear method is used to remove the set completely. The list ages is converted to a set using the set() method and the len() method is used to compare the lengths of the list and set.

Question 5:

- The radius of a circle is 30 meters.
- Calculate the area of a circle and assign the value to a variable name of `_area_of_circle_`
- Calculate the circumference of a circle and assign the value to a variable name of `_circum_of_circle_`.
- Take radius as user input and calculate the area.

A screenshot of a Jupyter Notebook interface. The title bar says "Machine learning Assignment 1" and "Last Checkpoint: a day ago (autosaved)". The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The toolbar shows icons for saving, undo, redo, and running code. The code cell contains the following Python code:

```
In [3]: r=30
        _area_of_circle=3.14*(r**2)
        print("area of circle :",_area_of_circle)
        _circum_of_circle=2*3.17*r
        print("circumference of circle :",_circum_of_circle)
        n=int(input())
        area=3.14*(n**2)
        print("Area of circle with user input :",area)

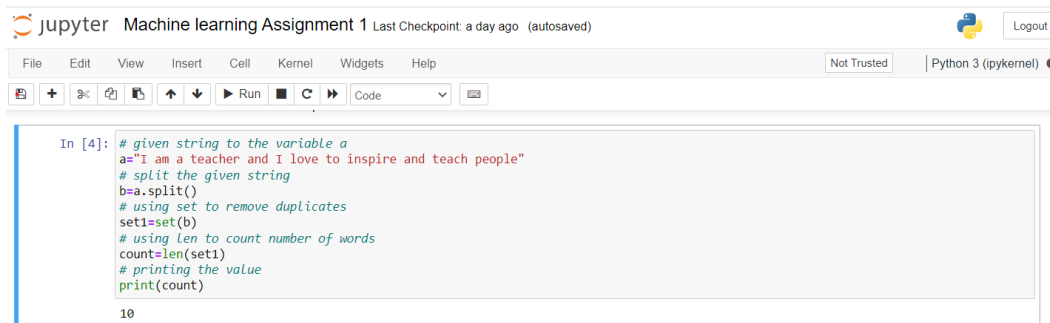
area of circle : 2826.0
circumference of circle : 190.2
12
Area of circle with user input : 452.16
```

The area and circumference of the circle are calculated using their standard formulae and stored in the variables `_area_of_circle_` and `_circum_of_circle_` respectively. Radius is taken as user input using the `input()` method and stored in the 'n' variable and area is calculated using πr^2 .

Question 6:

“I am a teacher and I love to inspire and teach people”

- How many unique words have been used in the sentence? Use the split methods and set to get the unique words

A screenshot of a Jupyter Notebook interface. The title bar says "Machine learning Assignment 1" and "Last Checkpoint: a day ago (autosaved)". The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The toolbar shows icons for saving, undo, redo, and running code. The code cell contains the following Python code:

```
In [4]: # given string to the variable a
a="I am a teacher and I love to inspire and teach people"
# split the given string
b=a.split()
# using set to remove duplicates
set1=set(b)
# using len to count number of words
count=len(set1)
# printing the value
print(count)

10
```

The given string is stored in a variable called 'a'. The `split()` method is used to get the list of words in the statement and the `set()` method is used to remove the duplicates. The `len()` method is used to count the unique words in the string.

Question 7:

Use a tab escape sequence to get the following lines.

Name	Age	Country City
Asabeneh	250	Finland Helsinki

```

In [5]: #tab escape
# declaring a string to the txt variable
txt = "Name\tAge\tCountry\tCity\nAsabeneh\t250\tFinland\tHelsinki"
# print the required output
print(txt)

```

Name	Age	Country	City
Asabeneh	250	Finland	Helsinki

The tab escape sequence is used in order to print the required output.

Question 8:

Use the string formatting method to display the following:

radius = 10

area = 3.14 * radius ** 2

“The area of a circle with radius 10 is 314 meters square.”

```

In [6]: #Assigned variable radius
radius=10
#Assigned area of circle formula
area=3.14 * radius ** 2
# print the output using string format
print("The area of circle with radius {} is {} meters square".format(radius,area))

```

The area of circle with radius 10 is 314.0 meters square

The string formatting method is done using escape sequences and the actual string to be printed is enclosed in quotes and the variables are written wherever the values are needed to be printed.

Question 9:

Write a program, which reads weights (lbs.) of N students into a list and convert these weights to kilograms in a separate list using Loop. N: No of students (Read input from user)

Ex: L1: [150, 155, 145, 148]

Output: [68.03, 70.3, 65.77, 67.13]

```

In [9]: n=int(input("Enter number of student's weight to be calculated"))
weights_in_lbs=[]
weights_in_kg=[]
#appending the elements into the List
for i in range(n):
    weights_in_lbs.append(int(input("weight {} \n".format(i+1))))
print(weights_in_lbs)
#converting lbs to kilogram with exactly 2 decimal places
for i in range(len(weights_in_lbs)):
    lbs=0.45359237 #1lbs= 0.45359237kg
    temp=round(weights_in_lbs[i]*lbs,2)
    weights_in_kg.append(temp)
    temp=0
print(weights_in_kg)

```

Enter number of student's weight to be calculated5

weight 1
54
weight 2
76
weight 3
45
weight 4
87
weight 5
49

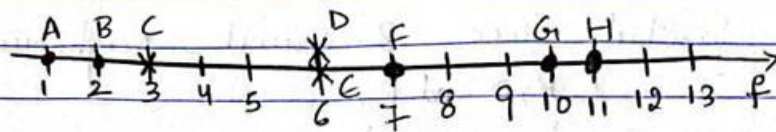
[54, 76, 45, 87, 49]
[24.49, 34.47, 20.41, 39.46, 22.23]

n is taken as input from user and weights are taken as input .add the elements into the list using append() each weight in lbs converted to kilograms and appended to weights_in_kg

Question 10:

Question 10:-

Divide the data equally into 2 Parts use the first Part as training and 2nd Part as testing.



Training = [B, D, H, F]

Testing = [A, C, E, G]

$k=3$

Consider A data Point.

Distance between A and B = $|1-2| = 1$

Distance between A and D = $|1-6| = 5$

Distance between A and F = $|1-7| = 6$

Distance between A and H = $|1-11| = 10$

3-nearest neighbours for A are B, D, F out of which two are of 'circle' type. Therefore A is also circle.

Consider C data Point;

Distance b/w C and B = $|3-2| = 1$

Distance b/w C and D = $|3-6| = 3$

Distance b/w C and F = $|3-7| = 4$

3-nearest neighbours are B, D, F out of which two are of 'circle' type. Therefore C is also circle.

Consider E data Point,

Distance b/w B and E = $|6-2| = 4$

Distance b/w D and E = $|6-6| = 0$

Distance b/w F and E = $|7-6| = 1$

Distance b/w H and E = $|11-6| = 5$

Similarly, Since 3-nearest neighbours are B, D and F, E is also a circle

Consider 'G' data point,

Distance b/w B and G = $|10-2| = 8$

Distance b/w D and G = $|10-6| = 4$

Distance b/w F and G = $|10-7| = 3$

Distance b/w H and G = $|10-11| = 1$

3-nearest neighbours are D, F, H out of which two are of 'circle' type. Therefore G is also circle.

2.

Data points	A	B	C	D	E	F	G	H
Actual Classification	•	•	X	X	X	•	•	•
Predicted Classification	•	•	•	X	•	•	•	•
	TP	TP	FP	TN	FP	TP	TP	TP

Prediction

	0	1
0	TN(1)	FP(2)
1	FN(0)	TP(2)

$$\text{Accuracy} = \frac{TP+TN}{P+N} = \frac{5+1}{5+3} = \frac{6}{8} = \frac{3}{4} = \underline{\underline{0.75}}$$

$$\text{Sensitivity} = \frac{TP}{TP+FN} = \frac{5}{5+0} = \underline{\underline{1}}$$

$$\text{Specificity} = \frac{TN}{FP+TN} = \frac{1}{2+1} = \frac{1}{3} = \underline{\underline{0.33}}$$

Github link : <https://github.com/TriveniBala/TriveniBala>