

GitHub: [https://github.com/raimukul/MachineLearning\\_Assignments](https://github.com/raimukul/MachineLearning_Assignments)

Video Link: <https://drive.google.com/file/d/1-x4RSX0YgdisfRdAac4x3awO81cZfa4u/view?usp=sharing>

## 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)

# Sorting the list and find the min and max age

```
ages.sort()
print ("Sorted list =", ages)
```

# min and max-age

```
print("Min age in list =", min(ages))
print("Max age in list =", max(ages))
```

# Add the min age again to the list

```
ages.insert(1, min(ages))
print(ages)
```

# Add the max-age again to the list

```
ages.insert(1, max(ages))
print(ages)
```

# Calculation of Median using statistics library

```
import statistics
med = statistics.median(ages)
print("Median of list is : " + str(med))
```

# Calculation of Average value

```
Average= sum(ages)/len(ages)
print ("Average of list = ", Average)
```

```
In [ ]: ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24]
# Sort the list and find the min and max age
ages.sort()
print("Sorted list =", ages)

Sorted list = [19, 19, 20, 22, 24, 24, 24, 25, 25, 26]
```

```
In [ ]: # min and max age
print("Min age in list =", min(ages))
print("Max age in list =", max(ages))

Min age in list = 19
Max age in list = 26
```

```
In [ ]: # Add the min age again to the list
ages.insert(1, min(ages))

print(ages)

[19, 19, 19, 20, 22, 24, 24, 24, 25, 25, 26]
```

```
In [ ]: # Add the max age again to the list
ages.insert(1, max(ages))
print(ages)

[19, 26, 19, 19, 20, 22, 24, 24, 24, 25, 25, 26]
```

```
In [ ]: import statistics
med = statistics.median(ages)
print("Median of list is : " + str(med))

Median of list is : 24.0
```

```
In [ ]: Average= sum(ages)/len(ages)
print ("Average of list = ", Average)

Average of list = 22.75
```

## Question 2

---

- Create an empty dictionary called a dog.
- Add name, color, breed, legs, and 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

```
# Create an empty dictionary called dog.
```

```
dog = {}
```

```
print ("Dog dictionary as follow :",dog)
```

```
# Add name, color, breed, legs, age to the dog dictionary
```

```
dog.update({'name':'Moti','color':'Grey','breed':'BullDog','legs':'4','age':'2'})
```

```
print ("Dog dictionary having values as follow : " ,dog)
```

```
# Create a student dictionary and add first_name, last_name, gender, age, marital  
status, skills, country, city and address as keys for the dictionary
```

```
student = {'first_name':'Mukul','last_name':'Rai','Gender':'Male','age':'23',  
'marital_status':'Single', 'skills':['Developer', 'Tester'],  
'Country':'India','City':'Gopalganj','Address':'HN 01, Vill+PO - Semranw , PIN: 841436'}  
print ("Student dictionary created as follow : " ,student)
```

```
# Get the length of the student dictionary
```

```
print ("Length of student dictionary:" ,len(student))
```

```
#Get the value of skills and check the data type, it should be a list
```

```
Dskill= student["skills"]
```

```
print(" value of skills: ", Dskill)
```

```
print(" value of skills: ", type(Dskill))
```

```
# Modify the skills values by adding one or two skills
```

```
student = {'skills': ['Python Developer', 'DevOps'],}
```

```
print ("Updated Student dictionary as follow : " ,student)
```

```
#Get the dictionary keys as a list
```

```
keysList = list(student.keys())
```

```
print('Key as list', keysList)
```

```
# Get the dictionary values as a list
```

```
valueList = list(student.values())
```

```
print('Key as list', valueList)
```

```

In [ ]: # Create an empty dictionary called dog.
dog = {}
print ("Dog dictionary as follow :",dog)

Dog dictionary as follow : {}

In [ ]: # Add name, color, breed, legs, age to the dog dictionary
dog.update({'name':'Moti','color':'Grey','breed':'Bulldog','legs':'4','age':'2'})
print ("Dog dictionary having values as follow :",dog)

Dog dictionary having values as follow : {'name': 'Moti', 'color': 'Grey', 'breed': 'Bulldog', 'legs': '4', 'age': '2'}

In [ ]: # Create a student dictionary and add first_name, last_name, gender, age, marital status, skills, country, city and address as keys for the dictionary
student = {'first_name':'Mukul','last_name':'Rai','Gender':'Male','age':'23','marital_status':'Single','skills':['Developer', 'Tester'],'Country':'India'}
print ("Student dictionary created as follow :", student)

Student dictionary created as follow : {'first_name': 'Mukul', 'last_name': 'Rai', 'Gender': 'Male', 'age': '23', 'marital_status': 'Single', 'skills': ['Developer', 'Tester'], 'Country': 'India', 'City': 'Gopalganj', 'Address': 'HN 01, Vill+PO - Semranw , PIN: 841436'}

In [ ]: # Get the length of the student dictionary
print ("Length of student dictionary:",len(student))

Length of student dictionary: 9

In [ ]: #Get the value of skills and check the data type, it should be a list
Dskill= student["skills"]
print(" value of skills: ", Dskill)
print(" value of skills: ", type(Dskill))

value of skills: ['Developer', 'Tester']
value of skills: <class 'list'>

In [ ]: # Modify the skills values by adding one or two skills
student = {'skills': ['Python Developer', 'DevOps'],}
print ("Updated Student dictionary as follow :", student)

Updated Student dictionary as follow : {'skills': ['Python Developer', 'DevOps']}

In [ ]: #Get the dictionary keys as a list
keysList = list(student.keys())
print('Key as list', keysList)

Key as list ['first_name', 'last_name', 'Gender', 'age', 'marital_status', 'skills', 'Country', 'City', 'Address']

In [ ]: # Get the dictionary values as a list
valueList = list(student.values())
print('Key as list', valueList)

Key as list ['Mukul', 'Rai', 'Male', '23', 'Single', ['Developer', 'Tester'], 'India', 'Gopalganj', 'HN 01, Vill+PO - Semranw , PIN: 841436']

```

## Question 3

- Create a tuple containing the 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 sibling's tuple and add the name of your father and mother and assign it to family\_members

#Create a tuple containing names of your sisters and your brothers (imaginary siblings are fine)

```

mySisters = ('Anupriya', 'Arya','Aayat','Shweta')
myBrothers = ('Ankit','Alok','Abhinav','Aryman','Aditya')
print('My sister names are : ', mySisters)
print('My brother names are : ', myBrothers)

```

```
# Join brothers and sisters tuples and assign it to siblings
```

```
siblings = myBrothers + mySisters
```

```
print('All siblings are:', siblings)
```

```
# How many siblings do you have?
```

```
print("Number of Siblings:", len(siblings))
```

```
family_members = siblings + ('Rambilash Rai', 'Geeta Rai')
```

```
print('All family members are:', family_members)
```

```
In [ ]: #Create a tuple containing names of your sisters and your brothers (imaginary siblings are fine)
mySisters = ('Anupriya', 'Arya', 'Aayat', 'Shweta')
myBrothers = ('Ankit', 'Alok', 'Abhinav', 'Aryman', 'Aditya')
print('My sister names are : ', mySisters)
print('My brother names are : ', myBrothers)

My sister names are : ('Anupriya', 'Arya', 'Aayat', 'Shweta')
My brother names are : ('Ankit', 'Alok', 'Abhinav', 'Aryman', 'Aditya')

In [ ]: # Join brothers and sisters tuples and assign it to siblings
siblings = myBrothers + mySisters
print('All siblings are:', siblings)

All siblings are: ('Ankit', 'Alok', 'Abhinav', 'Aryman', 'Aditya', 'Anupriya', 'Arya', 'Aayat', 'Shweta')

In [ ]: # How many siblings do you have?
print("Number of Siblings:", len(siblings))

Number of Siblings: 9

In [ ]: family_members = siblings + ('Rambilash Rai', 'Geeta Rai')
print('All family members are:', family_members)

All family members are: ('Ankit', 'Alok', 'Abhinav', 'Aryman', 'Aditya', 'Anupriya', 'Arya', 'Aayat', 'Shweta', 'Rambilash Rai', 'Geeta Rai')
```

## 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

#Find the length of the set it\_companies

```
it_companies = {'Facebook', 'Google', 'Microsoft', 'Apple', 'IBM', 'Oracle', 'Amazon'}
print("Length of IT Companies:", len(it_companies))
```

#Add 'Twitter' to it\_companies

```
it_companies.add('Twitter')
print("After adding Twitter", it_companies)
```

#Insert multiple IT companies at once to the set it\_companies

```
it_companies.update({'AIS', 'Infosys', 'Capgemini', 'Wipro', 'TCS'})
print("After adding multiple companies:", it_companies)
```

#Remove one of the companies from the set it\_companies

```
it_companies.remove('Capgemini')
print("After removing one company:", it_companies)
```

#What is the difference between remove and discard

#Discard doesn't raise any error if any item is not present in the set

```
it_companies.discard('TCS')
print("After discarding company:", it_companies)
```

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]

#Join A and B

```
print("Join A and B:", A.union(B))
```

#Find A intersection B

```
print("Intersection of A and B:", A.intersection(B))
```

#Is A subset of B

```
print("Is A subset of B:", A.issubset(B))
```

```
#Join A with B and B with A
```

```
#Join A and B  
print("Join A and B:", A.union(B))
```

```
#Join B and A  
print("Join B and A:", B.union(A))
```

```
#Disjoint  
print("Disjoint:", A.isdisjoint(B))
```

```
#What is the symmetric difference between A and B  
symmetry = A.symmetric_difference(B)  
print('The symmetry between A and B is=', symmetry)
```

```
#Delete the sets completely  
A.clear()  
B.clear()  
print('Value of A:', A)  
print('Value of B:', B)
```

```
#Convert the ages to a set and compare the length of the list and the set  
age = [22, 19, 24, 25, 26, 24, 25, 24]  
s = set(age)  
print(s)  
print('Type:', type(s) )  
print('length of list:', len(age) )  
print('length of set:', len(s) )
```

```
In [1]: #Find the length of the set it_companies  
it_companies = {'Facebook', 'Google', 'Microsoft', 'Apple', 'IBM', 'Oracle', 'Amazon'}  
print("Length of IT Companies:", len(it_companies))
```

Length of IT Companies: 7

```
In [2]: #Add 'Twitter' to it_companies  
it_companies.add('Twitter')  
print("After adding Twitter", it_companies)
```

After adding Twitter {'IBM', 'Facebook', 'Apple', 'Oracle', 'Amazon', 'Google', 'Twitter', 'Microsoft'}

```
In [6]: #Insert multiple IT companies at once to the set it_companies  
it_companies.update({'AIS', 'Infosys', 'Capgemini', 'Wipro', 'TCS'})  
print("After adding multiple companies:", it_companies)
```

After adding multiple companies: {'AIS', 'IBM', 'TCS', 'Facebook', 'Capgemini', 'Apple', 'Oracle', 'Amazon', 'Google', 'Twitter', 'Microsoft', 'Wipro', 'Infosys'}

```
In [7]: #Remove one of the companies from the set it_companies  
it_companies.remove('Capgemini')  
print("After removing one company:", it_companies)
```

After removing one company: {'AIS', 'IBM', 'TCS', 'Facebook', 'Apple', 'Oracle', 'Amazon', 'Google', 'Twitter', 'Microsoft', 'Wipro', 'Infosys'}

```
In [8]: #What is the difference between remove and discard  
##Discard doesn't raise any error if any item is not present in the set  
it_companies.discard('TCS')  
print("After discarding company:", it_companies)
```

After discarding company: {'AIS', 'IBM', 'Facebook', 'Apple', 'Oracle', 'Amazon', 'Google', 'Twitter', 'Microsoft', 'Wipro', 'Infosys'}



```
In [9]: 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]
```

```
In [10]: #Join A and B
         print("Join A and B:", A.union(B))

Join A and B: {19, 20, 22, 24, 25, 26, 27, 28}
```

```
In [11]: #Find A intersection B
         print("Intersection of A and B:", A.intersection(B))

Intersection of A and B: {19, 20, 22, 24, 25, 26}
```

```
In [13]: #Is A subset of B
         print("Is A subset of B:", A.issubset(B))

Is A subset of B: True
```

```
In [14]: #Join A with B and B with A
         #Join A and B
         print("Join A and B:", A.union(B))
         #Join B and A
         print("Join B and A:", B.union(A))

Join A and B: {19, 20, 22, 24, 25, 26, 27, 28}
Join B and A: {19, 20, 22, 24, 25, 26, 27, 28}
```

```
In [15]: #Disjoint
         print("Disjoint:", A.isdisjoint(B))

Disjoint: False
```

```
In [16]: #What is the symmetric difference between A and B  
symmetry = A.symmetric_difference(B)  
print('The symmetry between A and B is=', symmetry)
```

The symmetry between A and B is= {27, 28}

```
In [17]: #Delete the sets completely  
  
A.clear()  
B.clear()  
print('Value of A:', A)  
print('Value of B:', B)
```

Value of A: set()  
Value of B: set()

```
In [18]: #Convert the ages to a set and compare the length of the list and the set  
age = [22, 19, 24, 25, 26, 24, 25, 24]  
s = set(age)  
print(s)  
print('Type:', type(s) )
```

{19, 22, 24, 25, 26}  
Type: <class 'set'>

```
In [19]: print('length of list:', len(age) )  
print('length of set:', len(s) )
```

length of list: 8  
length of set: 5

## 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.

*#Calculate the area of a circle and assign the value to a variable name of area\_of\_circle*  
from math import pi

r = 30

print ("The area of the circle with radius " + str(r) + " is: " + str(pi \* r\*\*2))

*#Calculate the circumference of a circle and assign the value to a variable name of circum\_of\_circle*

```
print ("The circumference of the circle with radius " + str(r) + " is: " + str(2 * pi * r))
```

#Take radius as user input and calculate the area.

```
from math import pi
```

```
r = float(input ("Input the radius of the circle : "))
```

```
print ("The area of the circle with radius " + str(r) + " is: " + str(pi * r**2))
```

```
In [20]: #Calculate the area of a circle and assign the value to a variable name of area_of_circle
from math import pi
r = 30
print ("The area of the circle with radius " + str(r) + " is: " + str(pi * r**2))
```

The area of the circle with radius 30 is: 2827.4333882308138

```
In [21]: #Calculate the circumference of a circle and assign the value to a variable name of circum_of_circle
print ("The circumference of the circle with radius " + str(r) + " is: " + str(2 * pi * r))
```

The circumference of the circle with radius 30 is: 188.49555921538757

```
In [22]: #Take radius as user input and calculate the area.
from math import pi
r = float(input ("Input the radius of the circle : "))
print ("The area of the circle with radius " + str(r) + " is: " + str(pi * r**2))
```

Input the radius of the circle : 28

The area of the circle with radius 28.0 is: 2463.0086404143976

## 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.

#Unique word

```
sentence = "I am a teacher and I love to inspire and teach people".split()
```

```
unique = [uniq for uniq in sentence if uniq not in sentence[sentence.index(uniq)+1:]]
```

```
print('Unique words:', unique)
```

```
print('Number of Unique words:', len(unique))
```

```
In [26]: #Unique word
sentence = "I am a teacher and I love to inspire and teach people".split()
unique = [uniq for uniq in sentence if uniq not in sentence[sentence.index(uniq)+1:]]
print('Unique words:', unique)
print('Number of Unique words:', len(unique))
```

Unique words: ['am', 'a', 'teacher', 'love', 'to', 'inspire', 'teach', 'people']

Number of Unique words: 8

## Question 7

---

Use a tab escape sequence to get the following lines.

Name	Age	Country	City
------	-----	---------	------

Asabeneh	250	Finland	Helsinki
----------	-----	---------	----------

```
lines= "Name\t Age\tCountry\tCity\t\nAsabeneh 250\tFinland\tHelsinki"
print(lines)
```

```
In [27]: lines= "Name\t Age\tCountry\tCity\t\nAsabeneh 250\tFinland\tHelsinki"
         print(lines)
```

Name	Age	Country	City
Asabeneh	250	Finland	Helsinki

## 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.”

```
r=10
```

```
print(f'radius = 10')
```

```
print(f'area = 3.14*radius**2')
```

```
print(f"The area of circle with radius {r} is {3.14*r*r} meters square")
```

```
In [29]: r=10
         print(f'radius = 10')
         print(f'area = 3.14*radius**2')
         print(f"The area of circle with radius {r} is {3.14*r*r} meters square")
```

```
radius = 10
area = 3.14*radius**2
"The area of circle with radius 10 is 314.0 meters square"
```

## 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]

```
L1=[int(num) for num in input().split(" ")]
```

```
#Creating another list called Weight_kg  
Weight_kg=[]
```

```
#Using for loop to iterate the values and appending the list
```

```
for i in L1:  
    Weight_kg.append(round(i/2.205,2))
```

```
#Displaying the values in kgs after conversion  
print ("Values are:",Weight_kg)
```

```
In [31]: L1=[int(num) for num in input().split(" ")]  
          #Creating another list called Weight_kg  
          Weight_kg=[]  
          #Using for loop to iterate the values and appending the list  
          for i in L1:  
              Weight_kg.append(round(i/2.205,2))  
          #Displaying the values in kgs after conversion  
          print ("Values are:",Weight_kg)
```

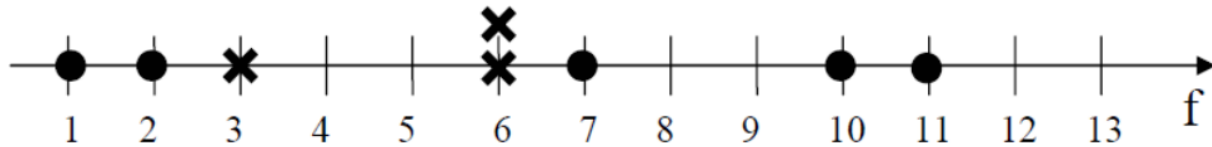
```
10 30 45 69 100
```

```
Values are: [4.54, 13.61, 20.41, 31.29, 45.35]
```

### Question 10

---

The diagram below shows a dataset with 2 classes and 8 data points, each with only one feature value, labeled  $f$ . Note that there are two data points with the same feature value of 6. These are shown as two x's one above the other. Provide stepwise mathematical solution, do not write code for it.



1. Divide this data equally into two parts. Use first part as training and second part as testing. Using KNN classifier, for  $K=3$ , what would be the predicted outputs for the test samples? Show how you arrived at your answer.
2. Compute the confusion matrix for this and calculate accuracy, sensitivity and specificity values.

10.

F	1	2	3	6	6	7	10	11
label	1	1	0	0	0	1	1	1

testing set

Training set

① Using KNN classifier  
K = 3

$$d = \sqrt{(x_0 - x_1)^2}$$

points need to be calculated  
are

$$\begin{matrix} (6, 6) & (6, 3) & (6, 2) & (6, 1) \\ \uparrow & \uparrow & & \\ x_0 & x_1 & & \end{matrix}$$

$$(6, 6) \rightarrow d = \sqrt{(6-6)^2} = 0$$

$$(6, 3) \rightarrow d = \sqrt{(6-3)^2} = 3$$

$$(6, 2) \rightarrow d = \sqrt{(6-2)^2} = 4$$

} nearest

$$(6,1) \rightarrow d = \sqrt{(6-1)^2} = 5$$

i.e.  $(0,0,1)$

max = 0 (O/P is also 0)

Calculate for rest points which are also 0 (Predicted)

## ② Confusion Matrix

$$\text{accuracy} = (TP + TN) / (TN + FP + FN + TP)$$

$$\text{sensitivity} = TP / (TP + FN)$$

$$\text{specificity} = TN / (FP + TN)$$

	0	1
0	TN = 1	FP = 0
1	FN = 3	TP = 0



$$A = (0+1) / (1+0+3+0)$$

$$= \frac{1}{4}$$

$$= 25\%$$

$$S = 0 / (0+3)$$

$$= 0$$

$$SP = 1 / 0+1$$

$$= 1$$