

Video :

<https://drive.google.com/drive/folders/1CCPgLY40DmIOH7hVmoTLIZre6bltm5Mu?usp=sharing>

**Question 1:**

-The list ages is given. I have been asked to sort the list and find the maximum and minimum age.

I sorted the list using the `sort()` function and used `max()` and `min()` functions in python to find the maximum and minimum age present in the list.

-I have been asked to add the minimum and maximum age to my original list which I did using the `append` function which adds new values to the list.

-Then I have been asked to find the median.

The median means the middle element if the number of elements is odd and the sum of 2 middle elements if the number is even. I used a mod operator the length of the list and then used `if else` to calculate the median for odd and even elements

-Then I have been asked to find the average of the list.

The average is the sum of total elements divided by the total no. of inputs. I have used a `for` loop to calculate the sum of elements and then divided it by the total no. of elements in the list

-Finally I have been asked to get the range of the list. Range is defined as the difference of maximum element and the minimum element in the list. Which I have done in the code. I have attached my screenshot for your reference.

Mac OS menu bar: Finder, File, Edit, View, Go, Window, Help. System status: Thu Sep 1 9:14 PM.

Browser tabs: Inbox - a..., Untitled - x, (1,143 un..., Download, Inbox (66..., Inbox - a..., super30a..., Job Open..., Internal S..., +.

Browser address bar: localhost:8888/notebooks/Downloads/Masters/Machine%20Learning/Assignment1/Untitled.ipynb?kernel\_name=python3

Jupyter Notebook interface:

- Header: jupyter Untitled Last Checkpoint: 32 minutes ago (unsaved changes) Logout
- Menu: File, Edit, View, Insert, Cell, Kernel, Widgets, Help. Trusted Python 3 (ipykernel)
- Toolbar: New, Open, Save, Copy, Paste, Undo, Redo, Run, Stop, Restart, Code, Help.

### Question 1

```
In [4]: age = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24]
age.sort() #sorting the age list as per requirement
print(age) #printing the age in sorted order

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

```
In [9]: maxage = max(age) #finding the maximum age in the list
minage = min(age) #finding the minimum age in the list
print("Max age = ",maxage)
print("Min age = ",minage)

Max age = 26
Min age = 19
```

```
In [10]: age.append(maxage) #adding max age to the list
age.append(minage) #adding min age to the list
print(age)

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

```
In [11]: age.sort() #sorting the age list again to find the median
m = len(age) #getting the length of the list
n = m // 2
if(m % 2 == 0): #if the length is even then will take 2 middle elements
    print("Median = ",(age[n-1] + age[n]) / 2)
else: #if the length is odd, just the middle element
    print("Median = ",age[n])

Median = 24.0
```

Mac OS dock: Finder, Launchpad, Safari, App Store, System Preferences, Visual Studio Code, Mail (3,863), Terminal, Google Chrome, Notes, Teams, Word, Docker Desktop, File Explorer, Task View, Recycle Bin.

```
minage = min(age) #finding the minimum age in the list
print("Max age = ",maxage)
print("Min age = ",minage)

Max age = 26
Min age = 19

In [10]: age.append(maxage) #adding max age to the list
age.append(minage) #adding min age to the list
print(age)

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

In [11]: age.sort() #sorting the age list again to find the median
m = len(age) #getting the length of the list
n = m // 2
if(m % 2 == 0): #if the length is even then will take 2 middle elements
    print("Median = ",(age[n-1] + age[n]) / 2)
else: #if the length is odd, just the middle element
    print("Median = ",age[n])

Median = 24.0

In [13]: sum = 0
for i in age:
    sum = sum + i #Adding all the numbers in age list to get the total sum
print("Average is =", sum/m) #total sum divided by total no of elements gives the average

Average is = 22.75

In [16]: print("Range of list is", maxage-minage) #maxage - minage gives us the range of the list

Range of list is 7
```

## Question 2

- I have initialized an empty dictionary named dog. Then later added the attributes of Dog as asked.
- Then I initialized a dictionary named student and entered it's attributes.
- I have been asked to calculate the length of the dictionary which I did by using the built in function `len(student)`. This **len** function gives the length of the dictionary in python.
- Then I have been asked to print the values of the key skills in the student dictionary. I did the same using the keyword `student["skills"]`. Giving the key to the dictionary gives out the value or values present in the key.
- Next I have been asked to specify the data type of the key skills. I used the built in function `type` which gives out the data type of the variable.
- Next I have used `append` keyword to add values to my key skills and then print all the keys in the dictionary. For this I used the `student_keys` command which lists out all the keys in a dictionary.
- Finally I have been asked to give the values of the dictionary as a list for which I used `student_values`.

```
Question 2

In [19]: dog = {} #creating dictionary
dog = {"name": "Tiesto", "color": "Golden", "breed": "Golden retriever", "legs": "4", "age": "2"} #adding attributes
student = {"first_name": "Aayush", "last_name": "Agrawal", "gender": "male", "age": "28", "marital status": "Single",
"skills": ["Growth Mindset", "Adaptability"], "country": "Nepal", "city": "Kathmandu", "address": "Baafal"} #Creating student dictionary
print("Length of student dictionary is :", len(student))

Length of student dictionary is : 9

In [24]: print("Values of skills are:", student["skills"]) #Printing the values of skills key in student dictionary

Values of skills are: ['Growth Mindset', 'Adaptability']

In [26]: print("Data type of skills is:", type(student.get("skills"))) ##Getting the data type

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

In [27]: student["skills"].append("Team Player")
student_keys = student.keys()
print(student_keys) #Getting the keys of student dictionary

dict_keys(['first_name', 'last_name', 'gender', 'age', 'marital status', 'skills', 'country', 'city', 'address'])

In [28]: student_values = student.values()
print(student_values) #Printing student values

dict_values(['Aayush', 'Agrawal', 'male', '28', 'Single', ['Growth Mindset', 'Adaptability', 'Team Player'], 'Nepal', 'Kathmandu', 'Baafal'])
```

### Question 3

- First I have been asked to create a tuple of brothers and sisters which I did. A tuple creation in python is done by using small brackets().
- Next I added values to it.
- Next, I have been asked to join the two tuples and name it siblings. Joining the two tuples is simple using the + sign as if 2 data types are same, we can join the two data structures in python.
- I have been asked to give the total no. of siblings which I gave out by using function len.
- Next I have been asked to create a new tuple family members which has the data of siblings and then add my father's and mother's name in it. I did this by using the + sign again.

### Question 3

```
In [29]: brothers = ("Abhi", "Ankur") #Creating a brother tuple
sisters = ("Deevoo", "Avantika", "Shaal") #creating a sister tuple
siblings = brothers + sisters #Adding 2 tuples
print("Total siblings that I have is :", len(siblings))

Total siblings that I have is : 5

In [30]: family_members = siblings + ("Ajesh", "Sneha")
print("Members of my family", family_members)

Members of my family ('Abhi', 'Ankur', 'Deevoo', 'Avantika', 'Shaal', 'Ajesh', 'Sneha')
```

## Question 4

Here, I have initialized it companies, A, B and age as given.

-I have calculated the length using the len keyword

-Added a new it company using the add function.

-Added multiple companies at the same time using the update function.

-Then removed one company using the remove keyword.

- Discard function just deletes the element from the set if it is present and if it is not, it does nothing whereas if we use remove and if the element is not present in the set, then it throws an error

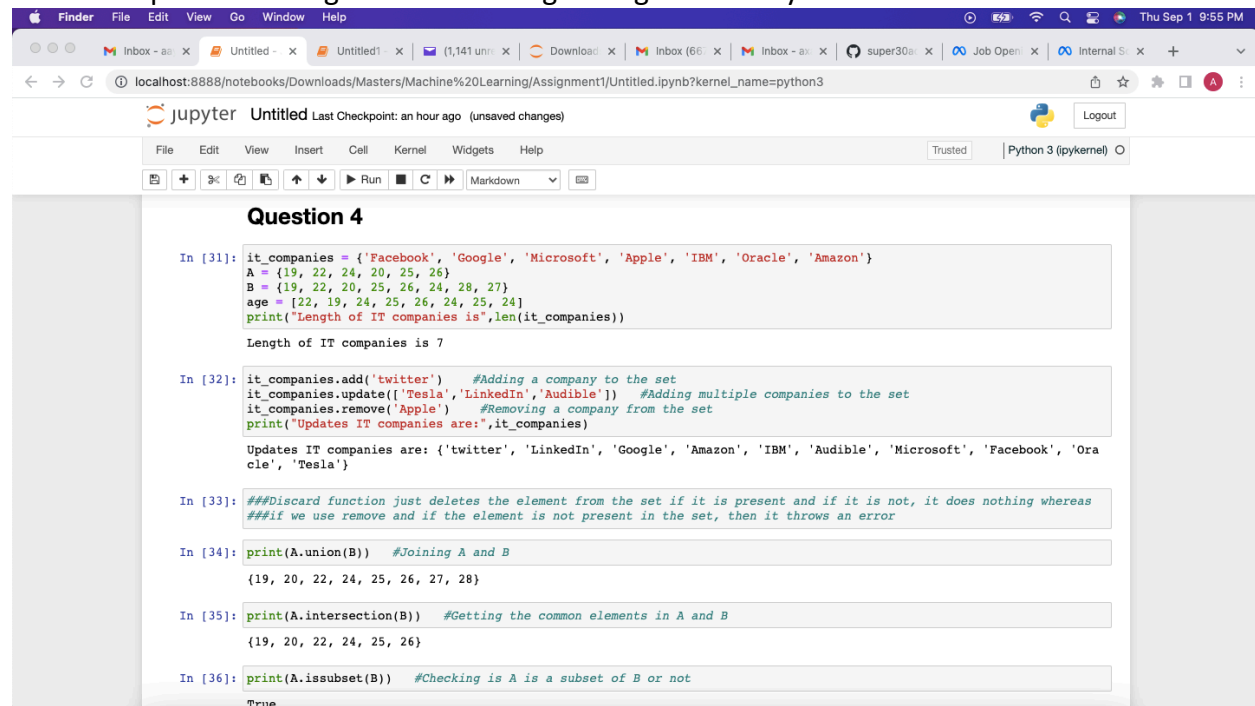
-I have found the union, intersection ,subset and disjoint using the same keywords.

-Also calculated the symmetric difference using the symmetric\_difference function.

-Deleted A and B using the del keyword.

-Type casted to change the data type.

-Then compared the length of both the age using the len keyword.



```
In [31]: 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 is",len(it_companies))

Length of IT companies is 7

In [32]: it_companies.add('twitter')      #Adding a company to the set
it_companies.update(['Tesla','LinkedIn','Audible'])  #Adding multiple companies to the set
it_companies.remove('Apple')      #Removing a company from the set
print("Updates IT companies are:",it_companies)

Updates IT companies are: {'twitter', 'LinkedIn', 'Google', 'Amazon', 'IBM', 'Audible', 'Microsoft', 'Facebook', 'Oracle', 'Tesla'}

In [33]: ###Discard function just deletes the element from the set if it is present and if it is not, it does nothing whereas
###if we use remove and if the element is not present in the set, then it throws an error

In [34]: print(A.union(B))      #Joining A and B

{19, 20, 22, 24, 25, 26, 27, 28}

In [35]: print(A.intersection(B))      #Getting the common elements in A and B

{19, 20, 22, 24, 25, 26}

In [36]: print(A.issubset(B))      #Checking is A is a subset of B or not

True
```

The screenshot shows a Jupyter Notebook with the following code and output:

```
In [36]: print(A.issubset(B)) #Checking is A is a subset of B or not
True

In [37]: print(A.isdisjoint(B)) #Checking if they are disjoint or not
False

In [39]: print(A.union(B))
print(B.union(A)) #Joining A with B then B with A
{19, 20, 22, 24, 25, 26, 27, 28}
{19, 20, 22, 24, 25, 26, 27, 28}

In [40]: print(A.symmetric_difference(B)) #Getting the symmetric difference
{27, 28}

In [41]: del A
del B #Deleting A and B

In [42]: set(age) #Converting age to a set
Out[42]: {19, 22, 24, 25, 26}

In [43]: print(len(set(age)) == len(age)) #Comparing the length of 2 sets
False
```

### Question 5

-The radius of the circle is given, so Calculating the area and circumference using the simple formulas.

-Also used the input keyword to get the input from user and used the same input to calculate the area.

#### Question 5

```
In [44]: r = 30
         _area_of_circle = 3.14*r*r
         print("Area of the circle is:", _area_of_circle) #Calculating the area of the circle and printing it
Area of the circle is: 2826.0

In [45]: _circum_of_circle = 2*3.14*r
         print("Circumference of the circle is:", circum_of_circle) #Calculating the circumference of the circle and printing
Circumference of the circle is: 188.4

In [46]: rad = int(input("Please enter the radius of the circle:")) #This line takes input from the user
         new_area = 3.14*rad*rad
         print("The new area of the circle is:", new_area) ##Calculating the area based on input from user
Please enter the radius of the circle:20
The new area of the circle is: 1256.0
```

### Question 6

-I used 3 different functions to calculate the unique words in the string. First I used split function to split all the words in the string. Then I put the words in a set. Since set can contain unique values, hence all the duplicates were abandoned. Then I used len function which gave the length of unique words in the string.

#### Question 6

```
In [47]: sentence = "I am a teacher and I love to inspire and teach people"
         print("Unique words in the sentence above is:", len(set(sentence.split()))) #Gives the unique words in the string
Unique words in the sentence above is: 10
```

## Question 7

-I used the \t to print the strings in the give format.

### Question 7

```
In [48]: print("Name\tAge\tCountry\tCity\nAsabeneh\t250\tFinland\tHelsinki") #Using tab escape to print the format as given
```

Name	Age	Country	City
Asabeneh	250	Finland	Helsinki

## -Question 8

-I used the f keyword for string formatting. In python 3, we can use f as well as format keyword to format a string along with numbers.

### Question 8

```
In [49]: print(f'\nradius = {10}') #Using String formatting to display the sentences
print(f'area = {3.14} * radius ** {2}')
print(f'The area of a circle with radius {10} is {314} meters square.')
```

```
radius = 10
area = 3.14 * radius ** 2
The area of a circle with radius 10 is 314 meters square.
```

## Question 9

-I asked the input of 5 different weights from the user and asked him/her to enter in pounds.

-After that, I stored the same in the list.

-Then I multiplied each weight by 0.45 to convert into kgs and display the same.

### Question 9

```
In [51]: weights = []
print("Please enter weights of 5 different individuals in pounds")
for i in range(0,5):
    weight = float(input()) #Getting 5 different inputs from users
    weights.append(weight) #Adding it to the list

weights_kg = [] #Declaring a new list which will contain the weight in kgs
for i in range(0,5):
    new_weight = weights[i] * 0.453592 #Going through thte list in pounds and converting it into kgs
    weights_kg.append(new_weight) #Adding it to the new list
print(weights_kg)
```

Please enter weights of 5 different individuals in pounds  
10  
20  
30  
40  
50  
[4.53592, 9.07184, 13.607759999999999, 18.14368, 22.6796]

### Question 10

Here 2 classes are given. So I will use the 1<sup>st</sup> class for training and 2<sup>nd</sup> class for testing my model.

Training Data:

1	O
2	O
3	X
6	X

KNN is given 3, Therefore I must find the 3 nearest neighbors of it.

For Point 6:

Nearest neighbors are 2=O, 3=X and 6=X.

Hence the prediction in testing set is X

For Point 7:

Nearest neighbors are 3=O, 6=X and 6=X.

Hence the prediction in testing set is X

For Point 10:

Nearest neighbors are 6=X, 6=X and 7=X.

Hence the prediction in testing set is X

For Point 11:

Nearest neighbors are 6=X, 7=X and 10=X.

Hence the prediction in testing set is X

Here, Let

O be positive and X be negative

The testing set or the confusion matrix is depicted as follows:

Data Set	Actual Output	Predicted Output	TP/TN/FP/FN
6	X	X	TN
7	O	X	FN
10	O	X	FN
11	O	X	FN

Therefore,

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



Now I have been asked to calculate the accuracy. For this I have used the formula,

$$\begin{aligned}\text{Accuracy} &= (\text{TP}/\text{TN}) / (\text{P}+\text{N}) \\ &= 0 + 1 / 3 + 1 \\ &= 1/4\end{aligned}$$

Sensitivity is calculated by,

$$\begin{aligned}\text{Sensitivity} &= \text{TP}/\text{P} \\ &= 0\end{aligned}$$

Specificity is calculated by the formula,

$$\begin{aligned}\text{Specificity} &= \text{TN}/\text{N} \\ &= 1\end{aligned}$$