

## Q1.

1. In this question, for the given list in the first sub question used .sort() method to find the min and max age. Sort(reverse=False) is the inbuilt function as it gives the ascending order.
2. To add the min age, and max age again, used the inbuilt min() and max() method.
3. As there is no inbuilt function for median, average which is also called mean in the list, hence statistics has been imported and then used statistics.median() and statistics.mean().
4. For range, did the subtraction.

The screenshot shows a Jupyter Notebook interface with the title "Assignment\_1@ML". The notebook has a single cell labeled "Q1" containing Python code. The code imports the "statistics" module and defines a list of ages. It then sorts the list in ascending order using the .sort() method with reverse=False. It calculates the minimum and maximum values by appending them to a list named "minmax". It prints the sorted list, the minimum and maximum values, the median using the statistics.median() function, the mean using the statistics.mean() function, and the range (maximum minus minimum) using a subtraction operation. The output shows the sorted list [19, 19, 19, 19, 20, 20, 20, 20, 21, 21, 21, 21, 22, 22, 22, 22, 23, 23, 23, 23, 24, 24, 24, 24, 25, 25, 25, 25, 26, 26, 26, 26], the minimum and maximum values as [19, 26], the add again min,max values to the list as [19, 19, 19, 19, 20, 20, 20, 20, 21, 21, 21, 21, 22, 22, 22, 22, 23, 23, 23, 23, 24, 24, 24, 24, 25, 25, 25, 25, 26, 26, 26, 26], the median as 24.0, the mean as 22.75, and the range as 7. The notebook interface includes a toolbar with various icons, a file menu, and a Python 3 (ipykernel) kernel indicator.

```
In [1]: import statistics
ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24]
ages.sort(reverse = False)
print('ages=',ages)
minmax = []
minmax.append(min(ages))
minmax.append(max(ages))
print('max_value and min_value =',minmax)
ages.insert(1,19)
ages.insert(12,26)
print('add again min,max values to the list=',ages)
print('median_ages =',statistics.median(ages))
print('mean or avg_ages =',statistics.mean(ages))
range = max(ages)-min(ages)
print('Range =',range)

ages= [19, 19, 20, 20, 22, 22, 24, 24, 24, 24, 25, 25, 25, 26]
max_value and min_value = [19, 26]
add again min,max values to the list= [19, 19, 19, 19, 20, 20, 20, 20, 21, 21, 21, 21, 22, 22, 22, 22, 23, 23, 23, 23, 24, 24, 24, 24, 25, 25, 25, 25, 26, 26, 26, 26]
median_ages = 24.0
mean or avg_ages = 22.75
Range = 7
```

## Q2

1. Created an empty dictionary labeled as 'Dog' as the dict rep as {}.
2. Added the keys and values to the dictionary.
3. Created another dict for student along with the keys and the values.
4. By using len() method, found the len of the dict
5. To know the data type of a key, type() method has been used.

The screenshot shows a Jupyter Notebook window titled "Assignment\_1@ML". The URL in the address bar is "localhost:8888/notebooks/Assignment\_1%40ML.ipynb". The notebook has tabs for Home Page, Assignment\_1, Untitled, ML\_Assign..., Untitled, Untitled, Untitled, Python\_1, and Untitled. The main area displays Python code and its output:

```
median_ages = 24.0
mean or avg_ages = 22.75
Range = 7
```

Below this, a section titled "Q2" contains the following code:

```
In [3]: Dog = {}
Dog = {"Name": "goldenretriever", "legs": 4, "color": "gold", "age": 12}
print(Dog)
Std_dict = {"first_name":'shiny',"last_name":'sherly','gender':'female','age':21,'marital status':'single','skills':'dancin
print(Std_dict)
print('length_dict =',len(Std_dict))
print(type('skills'))
print('skills=',Std_dict["skills"])
```

The output of this code is:

```
{'Name': 'goldenretriever', 'legs': 4, 'color': 'gold', 'age': 12}
{'first_name': 'shiny', 'last_name': 'sherly', 'gender': 'female', 'age': '21', 'marital status': 'single', 'skills': 'danci
ng,cooking,singing', 'country': 'india', 'city': 'vjjw'}
length_dict = 8
<class 'str'>
skills= dancing,cooking,singing
```

Below this, a section titled "Q3" contains the following code:

```
In [4]: bro=('tony','chotu')
```

The system tray at the bottom shows the date and time as 1/22/2023, 8:01 PM.

## Q3

1. Created two tuples by using ()
2. Then created a new variable siblings, and joined both the tuples by used '+' sign.
3. Then by using count(), found the length of the tuple
4. As the question says to add fathers and mothers name to the siblings tuple by creating new variable as family members.
5. Again used the same '+' sign for adding both the tuples.

The screenshot shows a Jupyter Notebook window with two code cells. The top cell, labeled Q3, contains Python code to calculate family members based on sibling and parent names. The bottom cell, labeled Q4, contains Python code to demonstrate set operations like union, intersection, and symmetric difference.

```
In [4]: bro='tony','chotu'
sis='chinu','fanny','nivvi'
siblings = bro+sis
print('siblings_names=',siblings)
print('count=',len(siblings))
dm_name = ('purushotham','aruna')
print('father,mother=',dm_name)
family_members = dm_name+siblings
print('familymembers',family_members)

siblings_names= ('tony', 'chotu', 'chinu', 'fanny', 'nivvi')
count= 5
father,mother= ('purushotham', 'aruna')
familymembers= ('purushotham', 'aruna', 'tony', 'chotu', 'chinu', 'fanny', 'nivvi')

In [5]: 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(len(it_companies))
it_companies.add('twitter')
print(it_companies)
```

## Q4

1. For the given set, found the length of the set by using the len() method.
2. To add an element to the set, used the .add() method
3. To add multiple elements into the set, update([]) method has been used.
4. To remove an element, .remove() method has been used.
5. To join or for the union of sets uses ‘|’ sign, for intersection, ‘&’ sign is used.
6. .issubset(), isdisjoint() methods has been used for the subset and disjoint sets.
7. For symmetric difference ‘^’ this sign has been used between the defined variables.
8. Len() to find the length

The screenshot shows a Jupyter Notebook window titled "Assignment\_1@ML". The code in cell [5] is as follows:

```
it_companies = {'Facebook', 'Google', 'Microsoft', 'Apple', 'IBM', 'Oracle', 'Amazon'}
A = {19, 22, 24, 28, 25, 26}
B = {19, 22, 28, 25, 26, 24, 28, 27}
age = [22, 19, 24, 25, 26, 24, 25, 24]
print(len(it_companies))
it_companies.add('twitter')
print(it_companies)
it_companies.update(['capgemini','accenture','cognizant'])
print(it_companies)
it_companies.remove('Oracle')
print(it_companies)
print(A|B)
print(A&B)
print(A.issubset(B))
print(A.isdisjoint(B))
C=A.union(B)
print(C)
print(A|B)
len(age)
```

The output of cell [5] is:

```
7
{'Microsoft', 'Apple', 'Google', 'IBM', 'twitter', 'Oracle', 'Amazon', 'Facebook'}
{'Microsoft', 'cognizant', 'capgemini', 'twitter', 'Oracle', 'Amazon', 'accenture', 'Facebook', 'Apple', 'Google', 'IBM'}
{'Microsoft', 'cognizant', 'capgemini', 'twitter', 'Amazon', 'accenture', 'Facebook', 'Apple', 'Google', 'IBM'}
{19, 20, 22, 24, 25, 26, 27, 28}
{19, 20, 22, 24, 25, 26}
True
False
{19, 20, 22, 24, 25, 26, 27, 28}
{27, 28}
```

The status bar at the bottom shows the date and time: 1/22/2023, 8:13 PM.

## Q5

- As the first sub question and the last sub question says to give 30 as input and to give the used to input the values, the first line of the code satisfies both the cases.
- Secondly, \_area\_of\_circle, and \_circumference\_of\_circle has been defined by the formulas for area and circumference

The screenshot shows a Jupyter Notebook running on a Windows desktop. The browser tab is titled "Assignment\_1@ML". The notebook has two visible cells:

**Q5**

```
In [6]: r_circle = int(input('enter radius:'))  
_area_of_circle = 3.14*r_circle*_circle  
_circumference_of_circle = 2*3.14*_circle  
print('area of circle ',_area_of_circle)  
print('circumference of circle ',_circumference_of_circle)  
  
enter radius:30  
area of circle = 2826.0  
circumference of circle = 188.4
```

**Q6**

```
In [8]: sen = "I am a teacher and I love to inspire and teach people"  
uni = set(sen.split(" "))  
print(uni)  
print('number_unq_words =',len(uni))  
  
{'and', 'teach', 'people', 'teacher', 'love', 'to', 'a', 'inspire', 'I', 'am'}  
number_unq_words = 10
```

The desktop taskbar at the bottom shows various application icons, and the system tray indicates it's 8:18 PM on 1/22/2023.

## Q6

1. For the sentence, if `.split()` method is defined it only splits the words by leaving behind the unique words.
2. Inorder to get the unique words `set(variable.split())` method is used.
3. `Len()` is used to find the length.

The screenshot shows a Jupyter Notebook interface running on a Windows desktop. The browser tab is titled "localhost:8888/notebooks/Assignment\_1%40MLipython". The notebook contains two cells:

**Q6**

```
In [8]: M sen = "I am a teacher and I love to inspire and teach people"
uni = set(sen.split(" "))
print(uni)
print('number_unq_words =' , len(uni))

{'and', 'teach', 'people', 'teacher', 'love', 'to', 'a', 'inspire', 'I', 'am'}
number_unq_words = 10
```

**Q7**

```
In [9]: M qn = "Name"\tAge" "\tCountrycity""\nAsabeneh"\t250"\tFinland Helsinki"
print(qn)

Name      Age      Countrycity
Asabeneh  250      Finland Helsinki
```

The taskbar at the bottom shows various application icons, and the system tray indicates it's 29°F Cloudy, 8:22 PM on 1/22/2023.

## Q7.

1. Used variable 'qn' to print the output in table format, I have used "\t" to create the required spacing between the header names
2. Used "\n" to print the data in the next line
3. Printed 'qn' to see the required output as shown below

The screenshot shows a Jupyter Notebook interface running on a Windows desktop. The browser tab is titled "localhost:8888/notebooks/Assignment\_1%40ML.ipynb". The notebook contains three cells:

**Q7**

```
In [9]: qn = "Name\tAge\tCountrycity\nAsabeneh\t250\tFinland Helsinki"
print(qn)
```

Name Age Countrycity  
Asabeneh 250 Finland Helsinki

**Q8**

```
In [10]: radius = 10
area = 3.14*radius**2

print("radius = {}".format(radius))
print("area = 3.14*radius**2")
print('The area of a circle with radius {} is {} meters square'.format(radius,area))

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

**Q9**

Windows taskbar icons include: Cloudy (28°F), Start, Search, File Explorer, Task View, Control Panel, Dell logo, Taskbar settings, and several pinned apps like Microsoft Edge, Google Chrome, and Microsoft Word.

## Q8.

1. Created a variable 'radius' to store the radius of a circle as 10
2. Created a variable 'area' with a given formula to calculate area
3. Printed the area value using string format method as shown below

The screenshot shows a Jupyter Notebook interface running on a Windows desktop. The browser tab is titled "localhost:8888/notebooks/Assignment\_1%40ML.ipynb". The notebook contains three cells:

**Q7**

```
In [9]: qn = "Name"\tAge" "\tCountrycity""\nAsabeneh""\t250""\tFinland Helsinki"
print(qn)
Name      Age      Countrycity
Asabeneh  250     Finland Helsinki
```

**Q8**

```
In [10]: radius = 10
area = 3.14*radius**2

print("radius = {}".format(radius))
print("area = 3.14*radius**2")
print('The area of a circle with radius {} is {} meters square'.format(radius,area))

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

**Q9**

The taskbar at the bottom of the screen shows various application icons, including File Explorer, Task View, Microsoft Edge, and Google Chrome. The system tray indicates the date and time as 1/22/2023, 8:37 PM.

## Q9.

1. Created a list which takes weights (lbs.) as input
2. Used Split method to split the input given in 'weights' variable
3. Used for loop to take each number in the list, added the required formula in for loop to convert the weights into Kilometre and added them into new list 'kgs'
4. Printed the list 'kgs' as below

The screenshot shows a Jupyter Notebook running in a web browser. The title bar indicates the URL is `localhost:8888/notebooks/Untitled8.ipynb?kernel_name=python3`. The notebook interface includes a toolbar with various icons for file operations, a menu bar with File, Edit, View, Insert, Cell, Kernel, Widgets, and Help, and a status bar showing "Trusted" and "Python 3 (ipykernel)". The main workspace displays a code cell labeled "In [1]:" containing Python code to calculate weights from input values. The output of the cell shows the calculated weights: 140 and 155, resulting in kgs = [63.0, 69.75]. Below this cell is another empty cell labeled "In [ ]:". The browser's address bar shows "localhost:8888/notebooks/Untitled8.ipynb?kernel\_name=python3". The taskbar at the bottom of the screen shows various application icons, including Microsoft Office and system status indicators like battery level and signal strength.

```
In [1]: weights=input()
L1=weights.split()
#print('weights=',L1)
kgs=[]
for i in L1:
    kgs.append((int(i)*0.45))
print('kgs=',kgs)

140 155
kgs= [63.0, 69.75]
```

Test set

Q.10

f 1 2 3 6 6 7 10 11

Label 1 1 0 0 0 1 1 1

Train set

As the question mentioned half Test & half Train.

i. Using KNN classifier with  $K=3$ .

$$d = \sqrt{(x_2 - x_1)^2}$$

Points to

$(6, 6)$      $(6, 3)$      $(6, 2)$      $(6, 1)$   $\rightarrow$  be calculated  
 $x_2$      $x_1$      $x_2$      $x_1$      $x_2$      $x_1$

i.e.,

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

$$d = \sqrt{(6-3)^2} = \sqrt{9} = 3 \quad (6, 3)$$

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

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

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

nearest  
as  
 $K=3$

## ii) Confusion Matrix:

$$\text{Accuracy} = \frac{TP + TN}{(TP + FP + FN + TN)}$$

TP = True +ve

TN = True -ve

FP = False +ve

FN = False -ve

$$\text{Sensitivity} = \frac{TP}{(TP + FN)}$$

$$\text{Specificity} = \frac{TN}{(FP + TN)}$$

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

$$A = \frac{(0+1)}{(1+0+3+0)}$$

$$= \frac{1}{4} = 25\%$$

Hence Accuracy is 25%.

— / — / —

$$\text{Sensitivity}(s) = \frac{0}{(0+3)} = 0$$

$$\text{Specificity } (sp) = \frac{1}{0+1} \\ = 1.$$