

Machine Learning - Assignment1

Name: Idavalapati Vijay Taraka Ramarao

ID: **700742485**

CRN: **13428**

Question1:

Given a list of “ages”, and perform sort operation, and get minimum and maximum element from the list and add the max and min back to the list.

```
File Edit View Navigate Code Refactor Run Tools Git Window Help ML - Assignment.py - Administrator
ML Assignment.py
Project READMEmd Assignment.py Test.py
Q- 0 results
1 print("\n")
2 print("Question 1")
3 ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24]
4
5 # Sort the list and find the min and max age
6 ages.sort()
7 print("Sorted Ages List:", ages)
8 # "min" Gives the minimum number from list
9 min_age = min(ages)
10 # "max" Gives the maximum number from list
11 max_age = max(ages)
12 print("Minimum Age:", min_age)
13 print("Maximum Age:", max_age)
14
15 # Add the min age and the max age again to the list
16 # "append" adds the element to the list
17 ages.append(min_age)
18 ages.append(max_age)
19 print("Updated List of Ages with Min and Max Ages:", ages)
20
```

Sort the ages list again, and calculate the median, average and range of the elements for the ages list

```
21 ages.sort()
22 # Find the median age (one middle item or two middle items divided by two)
23 # Get the length of the list and Divide the length by 2
24 # if remainder is 0, Then get the middle and middle -1 element and divide by 2
25 # if remainder is other than 0, Then get the middle element and divide by 2
26 lst_Length = len(ages)
27 print("Ages List Length:", lst_Length)
28 num = lst_Length // 2
29 if lst_Length % 2 == 0:
30     middlenumber = (ages[num] + ages[num - 1]) / 2
31 else:
32     middlenumber = ages[num]/2
33 print("Median Age Value:", middlenumber)
34
35 # Find the average age (sum of all items divided by their number)
36 # Iterate the element in ages and add it total counter-"sum-num" and then divide by the length of the list
37 sum_num = 0
38 for t in ages:
39     sum_num = sum_num + t
40 avg_ages = sum_num / len(ages)
41 print("Average Age:", avg_ages)
42
43 # Find the range of the ages (max minus min)
44 print("Range of Ages:", max_age - min_age)
```

Question1 Output:

```
C:\Users\Administrator\Documents\GitHub\ML\venv\Scripts\python.exe C:/Users/Administrator/Documents/GitHub/ML/Assignment.py

Question 1
Sorted Ages List: [19, 19, 20, 22, 24, 24, 24, 25, 25, 26]
Minimum Age: 19
Maximum Age: 26
Updated List of Ages with Min and Max Ages: [19, 19, 20, 22, 24, 24, 24, 25, 25, 26, 19, 26]
Ages List Length: 12
Median Age Value: 24.0
Average Age: 22.75
Range of Ages: 7
```

Question2:

Create a empty dictionary of “dog”, and add ‘name’, ‘color’, ‘breed’, ‘legs’, ‘age’

Create a student dictionary with keys first_name, last_name, gender, age, marital status, skills, country, city and address and get the student skills and perform updating of skills and print the keys and values of the student dictionary.

```
File Edit View Navigate Code Refactor Run Tools Git Window Help ML - Assignment.py - Administrator
ML Assignment.py
README.md Assignment.py test.py
C:\
# Question 2
print("\n")
print("Question 2")
# Create an empty dictionary called dog
dog = {}

# Add name, color, breed, legs, age to the dog dictionary
dog['name'] = 'Jimmy'
dog['color'] = 'Black'
dog['breed'] = 'Doberman'
dog['legs'] = 4
dog['age'] = 7
print("Dog Dictionary:", 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': 'Steve', 'last_name': 'Rogers', 'gender': 'Male', 'age': 25, 'marital status': 'Unmarried',
          'skills': ['Python', 'Java'], 'country': 'USA', 'city': 'Newyork', 'address': 'Saint Street'}
print("Student Dictionary:", 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
student_skills = student.get('skills')
print("Existing Student skills:", student_skills)
print("Data type of Student skills:", type(student_skills))
```

```
73
74 # Modify the skills values by adding one or two skills
75 student.update({'skills': ['React', 'JavaScript']})
76 print("Update Student skills:", student.get('skills'))
77
78 # Get the dictionary keys as a list
79 print("Student Dictionary Keys:", student.keys())
80
81 # Get the dictionary values as a list
82 print("Student Dictionary Value:", student.values())
```

Question2 Output:

```

Question 2
Dog Dictionary: {'name': 'Jimmy', 'color': 'Black', 'breed': 'Doberman', 'legs': 4, 'age': 7}
Student Dictionary: {'first_name': 'Steve', 'last_name': 'Rogers', 'gender': 'Male', 'age': 25, 'marital status': 'Unmarried', 'skills': ['Python', 'Java'], 'country': 'USA',
'city': 'Newyork', 'address': 'Saint Street'}
Length of Student Dictionary: 9
Existing Student skills: ['Python', 'Java']
Data type of Student skills: <class 'list'>
Update Student skills: ['React', 'JavaScript']
Student Dictionary Keys: dict_keys(['first_name', 'last_name', 'gender', 'age', 'marital status', 'skills', 'country', 'city', 'address'])
Student Dictionary Value: dict_values(['Steve', 'Rogers', 'Male', 25, 'Unmarried', ['React', 'JavaScript'], 'USA', 'Newyork', 'Saint Street'])

```

Question3:

Created a tuple containing names of your sisters and brothers and add it to sibling's tuple and the add father_name, mother_name to the family tuple

```

File Edit View Navigate Code Refactor Run Tools Git Window Help ML - Assignment.py - Administrator
ML Assignment.py
README.md Assignment.py Test.py
Q- 0 results
83
84 # Question 3
85 print("\n")
86 print("Question 3")
87
88 # Create a tuple containing names of your sisters and your brothers
89 brother = ('Steve', 'Jim')
90 sisters = ('Rachel', 'Jessy')
91
92 # Join brothers and sisters tuples and assign it to siblings
93 siblings = brother + sisters
94 print("The total siblings data is:", siblings)
95
96 # How many siblings do you have?
97 print("Total No of siblings:", len(siblings))
98
99 # Modify the siblings tuple and add the name of your father and mother and assign it to family_members
100 father_name = ('John',)
101 mother_name = ('Jenny',)
102 family_members = siblings + father_name + mother_name
103 print("Family Members Info:", family_members)

```

Question3 Output:

```

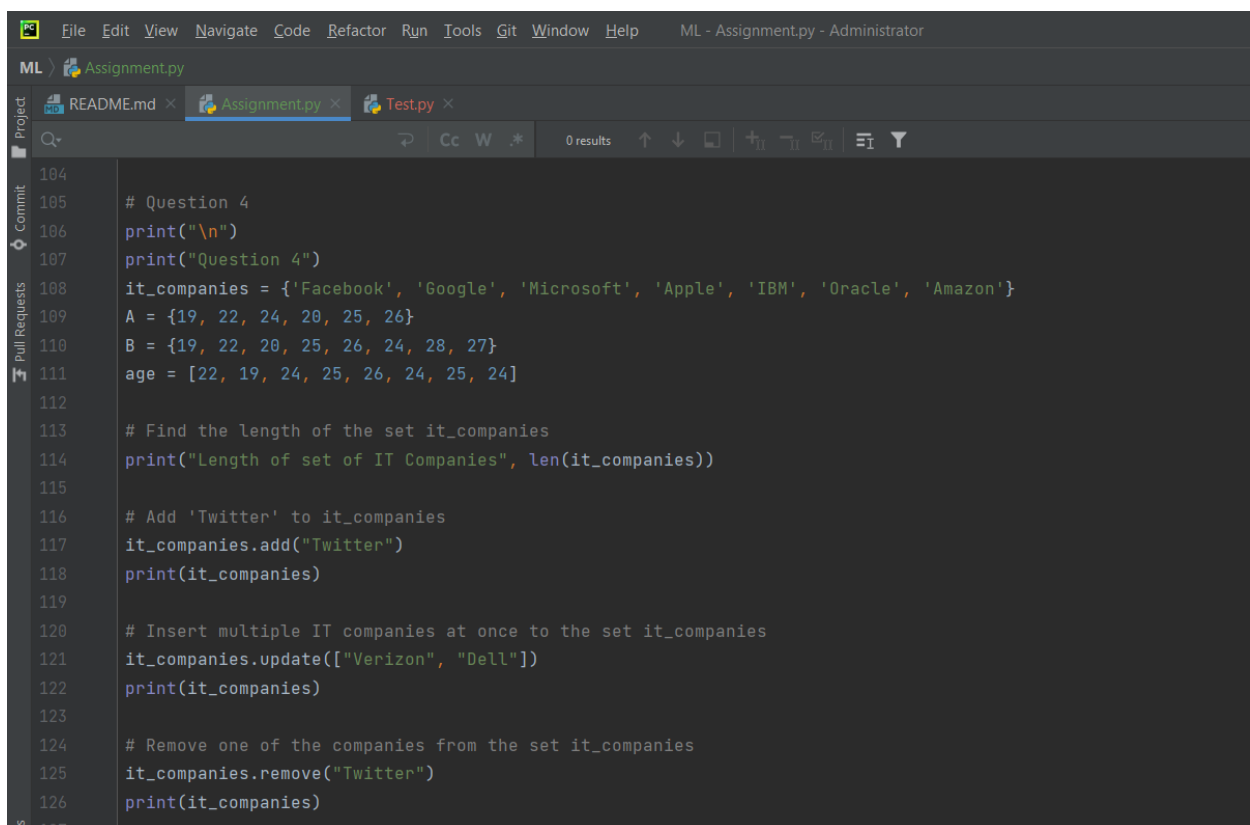
Question 3
The total siblings data is: ('Steve', 'Jim', 'Rachel', 'Jessy')
Total No of siblings: 4
Family Members Info: ('Steve', 'Jim', 'Rachel', 'Jessy', 'John', 'Jenny')

```

Question4:

Four set of Information given. And perform add, multiple insert, remove, operations on “it_companies”

On Set A and Set B, Join, Intersection, and check operations of disjoint, subset in A & B

A screenshot of a code editor window titled "ML - Assignment.py - Administrator". The editor shows a Python script with line numbers 104 to 127. The code defines a set of IT companies, two sets A and B, and a list 'age'. It then performs several operations on the 'it_companies' set: printing its length, adding 'Twitter', updating with 'Verizon' and 'Dell', and removing 'Twitter'.

```
104
105 # Question 4
106 print("\n")
107 print("Question 4")
108 it_companies = {'Facebook', 'Google', 'Microsoft', 'Apple', 'IBM', 'Oracle', 'Amazon'}
109 A = {19, 22, 24, 20, 25, 26}
110 B = {19, 22, 20, 25, 26, 24, 28, 27}
111 age = [22, 19, 24, 25, 26, 24, 25, 24]
112
113 # Find the length of the set it_companies
114 print("Length of set of IT Companies", len(it_companies))
115
116 # Add 'Twitter' to it_companies
117 it_companies.add("Twitter")
118 print(it_companies)
119
120 # Insert multiple IT companies at once to the set it_companies
121 it_companies.update(["Verizon", "Dell"])
122 print(it_companies)
123
124 # Remove one of the companies from the set it_companies
125 it_companies.remove("Twitter")
126 print(it_companies)
127
```

```
127
128 # What is the difference between remove and discard
129
130 # Both the Remove and discard methods are used to remove the specified item from the set
131 # by passing as argument to respective methods. The difference between those two are
132 # "Remove" method raises a KeyError exception
133 # "Discard" method does not raise an exception
134
135 # Join A and B
136 print("Union of A & B sets:", A | B)
137
138 # Find A intersection B
139 print("Intersection of A & B sets:", A & B)
140
141 # Is A subset of B
142 print("Is A subset of B:", A <= B)
143
144 # Are A and B disjoint sets
145 print("Are A and B disjoint sets:", A.isdisjoint(B))
146
```

```
File Edit View Navigate Code Refactor Run Tools Git Window Help ML - Assignment.py - Administrator
ML > Assignment.py
Project README.md x Assignment.py x Test.py x
Q. 0 results
146
147 # Join A with B and B with A
148 print("Join A with B:", A.union(B))
149 print("Join B with A:", B.union(A))
150
151 # What is the symmetric difference between A and B
152 print("symmetric difference:", A.symmetric_difference(B))
153
154 # Delete the sets completely
155 print("After Clearing set A, Elements in A:", A.clear())
156 print("After Clearing set B, Elements in B:", B.clear())
157
158 # Convert the ages to a set and compare the length of the list and the set.
159 age_set = {x for x in age}
160 print("Convert the ages to a set:", age_set)
161 print("Length of Ages List", len(age))
162 print("Length of Ages Set", len(age_set))
```

Question4 Output:

```
Question 4
Length of set of IT Companies 7
{'Microsoft', 'Twitter', 'Facebook', 'Oracle', 'Apple', 'IBM', 'Amazon', 'Google'}
{'Microsoft', 'Twitter', 'Facebook', 'IBM', 'Oracle', 'Dell', 'Apple', 'Verizon', 'Amazon', 'Google'}
{'Microsoft', 'Facebook', 'IBM', 'Oracle', 'Dell', 'Apple', 'Verizon', 'Amazon', 'Google'}
Union of A & B sets: {19, 20, 22, 24, 25, 26, 27, 28}
Intersection of A & B sets: {19, 20, 22, 24, 25, 26}
Is A subset of B: True
Are A and B disjoint sets: False
Join A with B: {19, 20, 22, 24, 25, 26, 27, 28}
Join B with A: {19, 20, 22, 24, 25, 26, 27, 28}
symmetric difference: {27, 28}
After Clearing set A, Elements in A: None
After Clearing set B, Elements in B: None
Convert the ages to a set: {19, 22, 24, 25, 26}
Length of Ages List 8
Length of Ages Set 5
```

Question5:

Calculate the area and circumference of circle using the defined radius and also radius from the user input.

```
File Edit View Navigate Code Refactor Run Tools Git Window Help ML - Assignment.py - Administrator
ML Assignment.py
README.md Assignment.py Test.py
0 results
164 # Question 5
165 print("\n")
166 print("Question 5")
167
168 # Calculate the area of a circle and assign the value to a variable name of _area_of_circle_
169 radius_of_circle = 30
170 PI = 3.142
171 _area_of_circle_ = PI * (radius_of_circle * radius_of_circle)
172 print("Area of Circle", _area_of_circle_)
173
174 # Calculate the circumference of a circle and assign the value to a variable name of _circum_of_circle_
175 _circum_of_circle_ = 2 * PI * radius_of_circle
176 print("Circumference of Circle", _circum_of_circle_)
177
178 # Take radius as user input and calculate the area
179 radius_input = float(input(' Please Enter the radius of a circle: '))
180 _input_area_of_circle_ = PI * (radius_input * radius_input)
181 print("Area of Circle with user Input radius:", _input_area_of_circle_)
182
```

Question5 Output:

Question 5

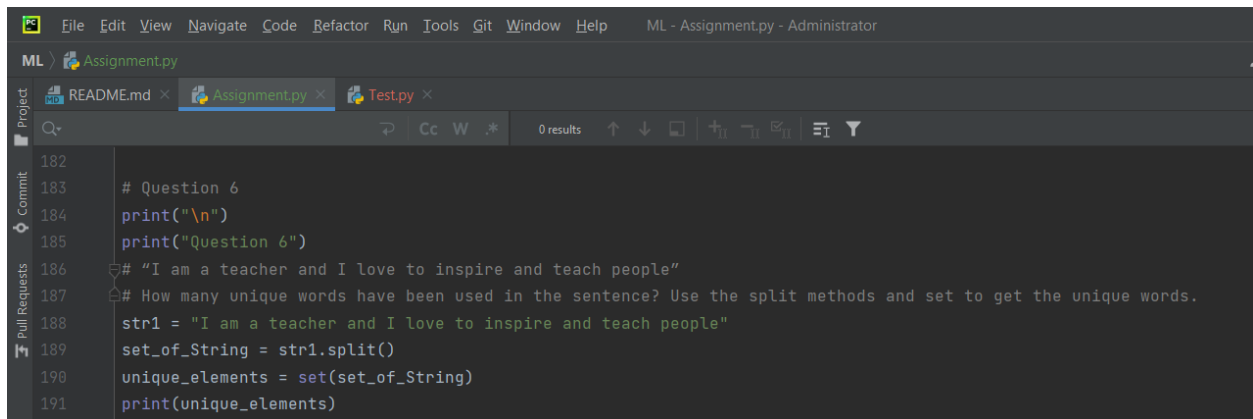
Area of Circle 2827.7999999999997

Circumference of Circle 188.51999999999998

Please Enter the radius of a circle: 10

Area of Circle with user Input radius: 314.2

Question6:



The screenshot shows an IDE window titled 'ML - Assignment.py - Administrator'. The editor has three tabs: 'README.md', 'Assignment.py', and 'Test.py'. The 'Assignment.py' tab is active, showing the following code:

```
182
183 # Question 6
184 print("\n")
185 print("Question 6")
186 # "I am a teacher and I love to inspire and teach people"
187 # How many unique words have been used in the sentence? Use the split methods and set to get the unique words.
188 str1 = "I am a teacher and I love to inspire and teach people"
189 set_of_String = str1.split()
190 unique_elements = set(set_of_String)
191 print(unique_elements)
```

Question6 Output:

Question 6

{'am', 'and', 'people', 'a', 'teach', 'I', 'inspire', 'love', 'teacher', 'to'}

Question7:

For the sentence given, print the sentence with tab escape sequence

```
File Edit View Navigate Code Refactor Run Tools Git Window Help ML - Assignment.py - Administrator
ML > Assignment.py
Project README.md x Assignment.py x Test.py x
0 results
193 # Question 7
194 print("\n")
195 print("Question 7")
196 # Use a tab escape sequence to get the following lines.
197 # Name Age Country City
198 # Asabeneh 250 Finland Helsinki
199 print("Name \tAge \tCountry \tCity")
200 print("Asabeneh \t250 \tFinland \tHelsinki")
201
```

Question7 Output:

```
Question 7
Name      Age      Country    City
Asabeneh  250      Finland   Helsinki
```

Question8:

Print the area of the circle in the specific format along with the radius

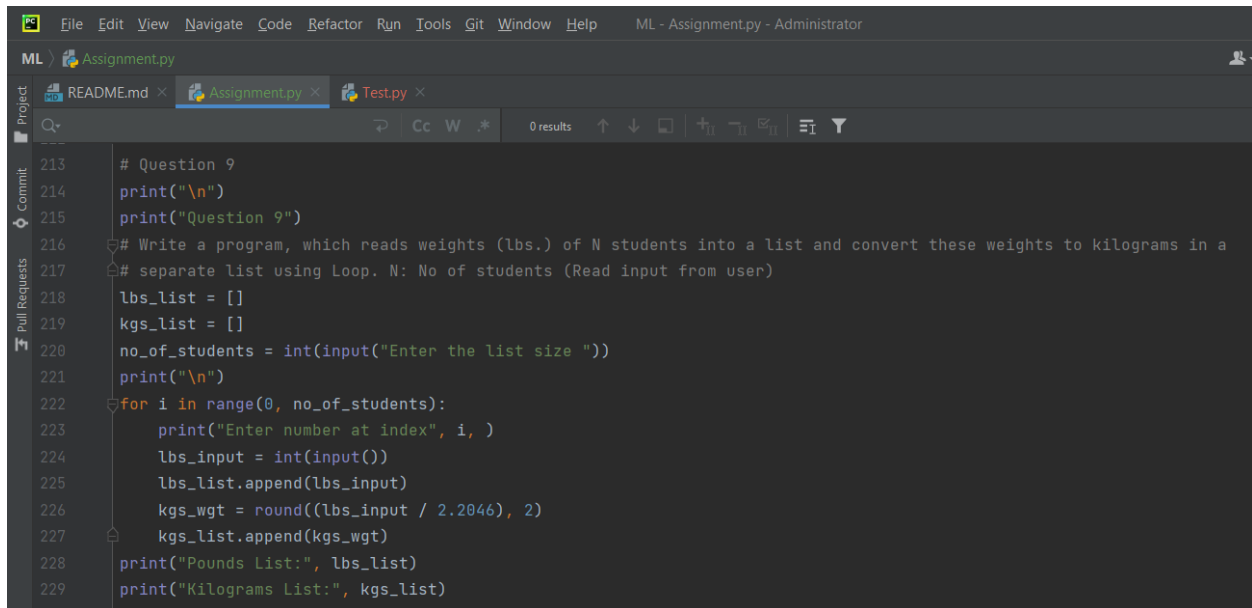
```
File Edit View Navigate Code Refactor Run Tools Git Window Help ML - Assignment.py - Administrator
ML > Assignment.py
Project README.md x Assignment.py x Test.py x
0 results
202 # Question 8
203 print("\n")
204 print("Question 8")
205 # Use the string formatting method to display the following:
206 # radius = 10
207 # area = 3.14 * radius ** 2
208 # "The area of a circle with radius 10 is 314 meters square."
209 radius = 15
210 area = 3.14 * radius ** 2
211 print("The area of a circle with radius %d is %d meters square." % (radius, area))
```

Question8 Output:

```
Question 8
The area of a circle with radius 15 is 706 meters square.
```

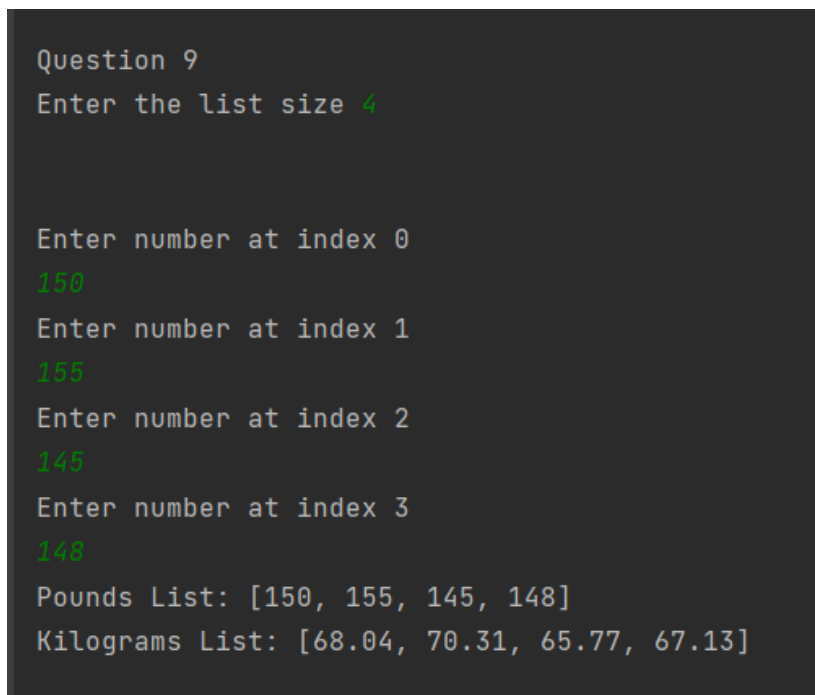
Question9:

Convert the list of “lbs” from the user input and then converting to “kilograms” by iterating the each element of lbs list and dividing by 2.2046 and printing the converted weight of kgs.



```
File Edit View Navigate Code Refactor Run Tools Git Window Help ML - Assignment.py - Administrator
ML Assignment.py
README.md Assignment.py Test.py
0 results
213 # Question 9
214 print("\n")
215 print("Question 9")
216 # Write a program, which reads weights (lbs.) of N students into a list and convert these weights to kilograms in a
217 # separate list using Loop. N: No of students (Read input from user)
218 lbs_list = []
219 kgs_list = []
220 no_of_students = int(input("Enter the list size "))
221 print("\n")
222 for i in range(0, no_of_students):
223     print("Enter number at index", i, )
224     lbs_input = int(input())
225     lbs_list.append(lbs_input)
226     kgs_wgt = round((lbs_input / 2.2046), 2)
227     kgs_list.append(kgs_wgt)
228 print("Pounds List:", lbs_list)
229 print("Kilograms List:", kgs_list)
```

Question9 Output:



```
Question 9
Enter the list size 4

Enter number at index 0
150
Enter number at index 1
155
Enter number at index 2
145
Enter number at index 3
148
Pounds List: [150, 155, 145, 148]
Kilograms List: [68.04, 70.31, 65.77, 67.13]
```

Question10:

Question 10:-

The data is divided equally to two sets. They are

Training Part:- 1, 2, 3, 10

Testing Part:- 6, 6, 7, 11

Here $k=3$

Euclidean distance, $d = \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2}$

\Rightarrow distance from 6 to 1, 2, 3, 10

$$d_1 = \sqrt{(6-1)^2} = 5$$

$$d_2 = \sqrt{(6-2)^2} = 4$$

$$d_3 = \sqrt{(6-3)^2} = 3$$

$$d_4 = \sqrt{(6-10)^2} = 4$$

There are maximum no of dots (.)'s, 6 is changed from X to .

⇒ distance from 7 to 1, 2, 3, 10

$$d_1 = \sqrt{(7-1)^2} = 6$$

$$d_2 = \sqrt{(7-2)^2} = 5$$

$$d_3 = \sqrt{(7-3)^2} = 4$$

$$d_4 = \sqrt{(7-10)^2} = 3$$

There are maximum no. of dots (.)'s, 10 is remains same as (.)

⇒ Distance from 11 to 1, 2, 3, 10

$$d_1 = \sqrt{(11-1)^2} = 10$$

$$d_2 = \sqrt{(11-2)^2} = 9$$

$$d_3 = \sqrt{(11-3)^2} = 8$$

$$d_4 = \sqrt{(11-10)^2} = 1$$

There are maximum no. of dots (.)'s, so 11 remains same as (.)

2) Confusion matrix :-

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

$$\text{Accuracy} :- \frac{TP + TN}{P + N} = \frac{2 + 0}{2 + 2} = \frac{2}{4} \\ = 0.5$$

$$\text{Sensitivity} :- \frac{TP}{TP + FN} = \frac{2}{2 + 0} = 1$$

$$\text{Specificity} :- \frac{TN}{FP + TN} = \frac{0}{2 + 0} = 0$$

Related Links:

Source Code:

<https://github.com/VijayTarakaRamarao/ML/blob/main/Assignment.py>

Video Recording:

https://github.com/VijayTarakaRamarao/ML/blob/main/ML_Assignment_Recording.mp4