

Machine Learning (Assignment # 1)

Submitted by

Shiva Kandagatla

700745245

Github link : <https://github.com/Shiva-Kandagatla98/ML-Assignment-1.git>

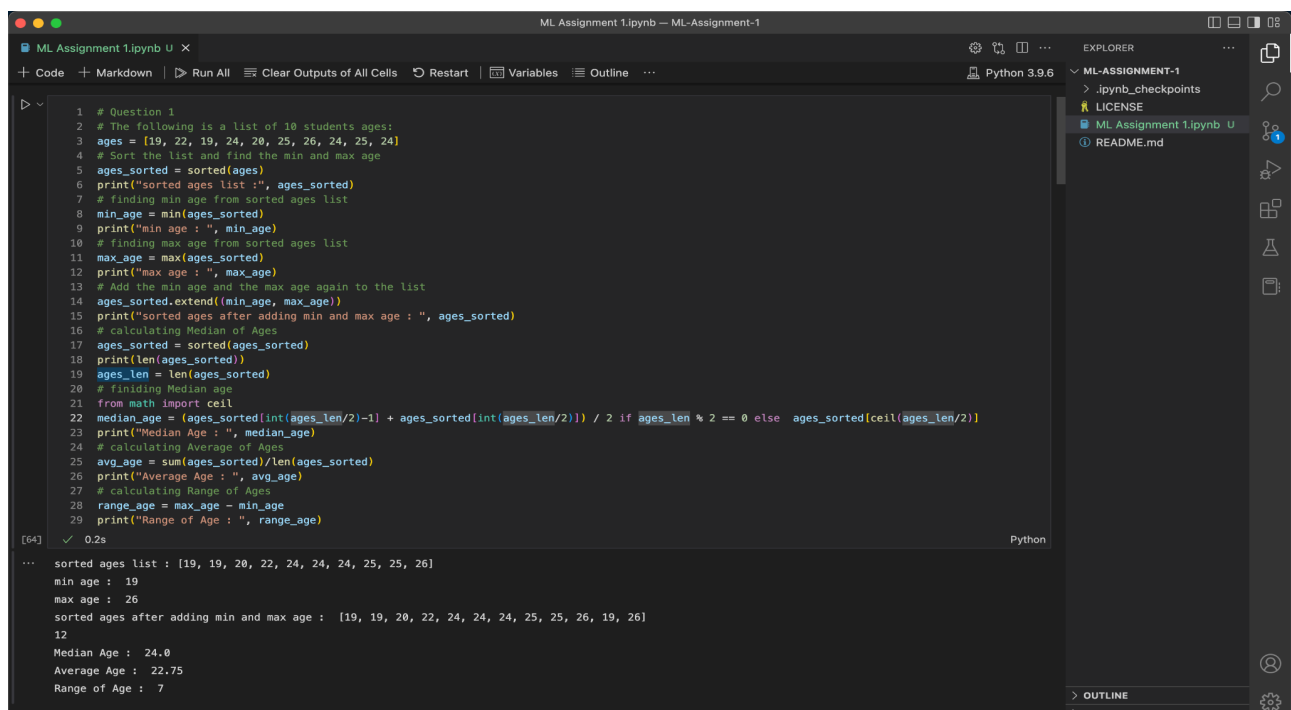
Video link :

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)



The screenshot shows a Jupyter Notebook window titled "ML Assignment 1.ipynb - ML-Assignment-1". The interface includes a top toolbar with icons for code, markdown, run, clear outputs, restart, variables, and outline. The main area displays a Python script with the following code:

```
1 # Question 1
2 # The following is a list of 10 students ages:
3 ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24]
4 # Sort the list and find the min and max age
5 ages_sorted = sorted(ages)
6 print("sorted ages list :", ages_sorted)
7 # finding min age from sorted ages list
8 min_age = min(ages_sorted)
9 print("min age : ", min_age)
10 # finding max age from sorted ages list
11 max_age = max(ages_sorted)
12 print("max age : ", max_age)
13 # Add the min age and the max age again to the list
14 ages_sorted.extend((min_age, max_age))
15 print("sorted ages after adding min and max age : ", ages_sorted)
16 # calculating Median of Ages
17 ages_sorted = sorted(ages_sorted)
18 print(len(ages_sorted))
19 ages_len = len(ages_sorted)
20 # finding Median age
21 from math import ceil
22 median_age = (ages_sorted[int(ages_len/2)-1] + ages_sorted[int(ages_len/2)]) / 2 if ages_len % 2 == 0 else ages_sorted[ceil(ages_len/2)]
23 print("Median Age : ", median_age)
24 # calculating Average of Ages
25 avg_age = sum(ages_sorted)/len(ages_sorted)
26 print("Average Age : ", avg_age)
27 # calculating Range of Ages
28 range_age = max_age - min_age
29 print("Range of Age : ", range_age)
```

The output of the script is displayed below the code:

```
[64]: ✓ 0.2s
sorted ages list : [19, 19, 20, 22, 24, 24, 24, 25, 25, 26]
min age : 19
max age : 26
sorted ages after adding min and max age : [19, 19, 20, 22, 24, 24, 24, 25, 25, 26, 19, 26]
12
Median Age : 24.0
Average Age : 22.75
Range of Age : 7
```

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

```
ML Assignment 1.ipynb — ML-Assignment-1
ML Assignment 1.ipynb U X
+ Code + Markdown | ▶ Run All | Clear Outputs of All Cells | Restart | Variables | Outline | Python 3.9.6
EXPLORER
ML-ASSIGNMENT-1
> .ipynb_checkpoints
LICENSE
ML Assignment 1.ipynb U
① README.md

1 # Question 2
2 # Create an empty dictionary called dog
3 dog = dict()
4 # Adding data to dog dictionary
5 dog['name'] = 'Jimmy'
6 dog['color'] = 'Black'
7 dog['breed'] = 'German Shepherd'
8 dog['legs'] = 4
9 dog['age'] = 3
10 print("dog dictionary : ", dog)
11 # creating student dictionary with data
12 student = {
13     "first_name": "Shiva",
14     "last_name": "Kandagatla",
15     "gender": "male",
16     "age": 24,
17     "marital status": "single",
18     "skills": ["python", "ML", "web development"],
19     "country": "India",
20     "city": "Hyderabad",
21     "address": "Madhapur, Hyderabad, India"
22 }
23 print("student dictionary : ", student)
24 # length of the student dictionary
25 length_student = len(student)
26 print("length of student dictionary : ", length_student)
27 # skills of the student from the dictionary
28 skills = student['skills']
29 print("skills of student : ", skills)
30 # type of skills
31 print("type of skills : ", type(skills))
32 # updating student skills
33 student['skills'].extend(["Mobile App Development"])
34 print("updated student skills : ", student['skills'])
35 # keys of student dictionary
36 print("keys of student dictionary : ", list(student.keys()))
37 # values of student dictionary
38 print("values of student dictionary : ", list(student.values()))
39

[65] ✓ 0.2s Python
dog dictionary : {'name': 'Jimmy', 'color': 'Black', 'breed': 'German Shepherd', 'legs': 4, 'age': 3}
```

```
ML Assignment 1.ipynb — ML-Assignment-1
ML Assignment 1.ipynb U X
+ Code + Markdown | ▶ Run All | Clear Outputs of All Cells | Restart | Variables | Outline | Python 3.9.6
EXPLORER
ML-ASSIGNMENT-1
> .ipynb_checkpoints
LICENSE
ML Assignment 1.ipynb U
① README.md

12 student = {
13     "first_name": "Shiva",
14     "last_name": "Kandagatla",
15     "gender": "male",
16     "age": 24,
17     "marital status": "single",
18     "skills": ["python", "ML", "web development"],
19     "country": "India",
20     "city": "Hyderabad",
21     "address": "Madhapur, Hyderabad, India"
22 }
23 print("student dictionary : ", student)
24 # length of the student dictionary
25 length_student = len(student)
26 print("length of student dictionary : ", length_student)
27 # skills of the student from the dictionary
28 skills = student['skills']
29 print("skills of student : ", skills)
30 # type of skills
31 print("type of skills : ", type(skills))
32 # updating student skills
33 student['skills'].extend(["Mobile App Development"])
34 print("updated student skills : ", student['skills'])
35 # keys of student dictionary
36 print("keys of student dictionary : ", list(student.keys()))
37 # values of student dictionary
38 print("values of student dictionary : ", list(student.values()))
39

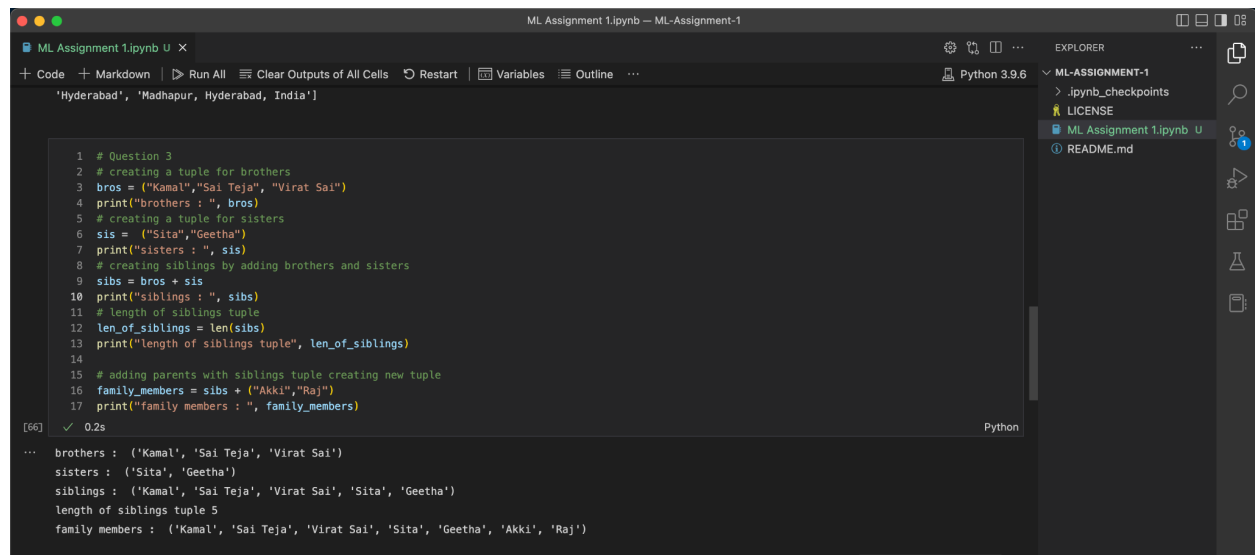
[65] ✓ 0.2s Python

... dog dictionary : {'name': 'Jimmy', 'color': 'Black', 'breed': 'German Shepherd', 'legs': 4, 'age': 3}
student dictionary : {'first_name': 'Shiva', 'last_name': 'Kandagatla', 'gender': 'male', 'age': 24, 'marital status': 'single', 'skills': ['python', 'ML', 'web development'], 'country': 'India', 'city': 'Hyderabad', 'address': 'Madhapur, Hyderabad, India'}
length of student dictionary : 9
skills of student : ['python', 'ML', 'web development']
type of skills : <class 'list'>
updated student skills : ['python', 'ML', 'web development', 'Mobile App Development']
keys of student dictionary : ['first_name', 'last_name', 'gender', 'age', 'marital status', 'skills', 'country', 'city', 'address']
values of student dictionary : ['Shiva', 'Kandagatla', 'male', 24, 'single', ['python', 'ML', 'web development', 'Mobile App Development'], 'India', 'Hyderabad', 'Madhapur, Hyderabad, India']

main* 0 0 0 -- VIM: DISABLED --
<ExtensionDisable> Jupyter Server: Local Ln 22, Col 68 LF Cell 5 of 11 Go Live
```

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



```
1 # Question 3
2 # creating a tuple for brothers
3 bros = ("Kamal","Sai Teja", "Virat Sai")
4 print("brothers : ", bros)
5 # creating a tuple for sisters
6 sis = ("Sita","Geetha")
7 print("sisters : ", sis)
8 # creating siblings by adding brothers and sisters
9 sibs = bros + sis
10 print("siblings : ", sibs)
11 # length of siblings tuple
12 len_of_siblings = len(sibs)
13 print("length of siblings tuple", len_of_siblings)
14
15 # adding parents with siblings tuple creating new tuple
16 family_members = sibs + ("Akki","Raj")
17 print("family members : ", family_members)
```

[66] ✓ 0.2s Python

```
... brothers : ('Kamal', 'Sai Teja', 'Virat Sai')
sisters : ('Sita', 'Geetha')
siblings : ('Kamal', 'Sai Teja', 'Virat Sai', 'Sita', 'Geetha')
length of siblings tuple 5
family members : ('Kamal', 'Sai Teja', 'Virat Sai', 'Sita', 'Geetha', 'Akki', 'Raj')
```

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.

The screenshot shows a Jupyter Notebook titled "ML Assignment 1.ipynb" with the following code and output:

```

1 # Question 4
2 it_companies = {'Facebook', 'Google', 'Microsoft', 'Apple', 'IBM', 'Oracle', 'Amazon'}
3 A = {19, 22, 24, 28, 25, 26}
4 B = {19, 22, 20, 25, 26, 24, 28, 27}
5 age = [22, 19, 24, 25, 26, 24, 25, 24]
6
7 # length of it_companies set
8 print("length of it_companies set : ", len(it_companies))
9 # adding company to it_company
10 it_companies.add("Twitter")
11 # removing company from it_company
12 it_companies.discard("Oracle")
13 print("it companies : ", it_companies)
14 print(""" remove vs discard :
15 Remove method deletes the element from the list if not present it throws Key error where as,
16 Discard method deleted the element from the list otherwise it returns None""")
17 # joining A and B sets
18 print("Join of A and B : ", A.union(B))
19 # intersection of A and B sets
20 print("Intersection of A and B : ", A.intersection(B))
21 # checking if A is subset of B
22 print("Is A subset of B : ", A.issubset(B))
23 # check if A is disjoint of B
24 print("Is A disjoint of B : ", A.isdisjoint(B))
25 # A union B and B union A
26 print("A union B : ", A.union(B))
27 print("B union A : ", B.union(A))
28 # symmetric difference between two sets
29 print("set A difference with set B : ", A.difference(B))
30 # deleting sets A and B
31 A.clear()
32 B.clear()
33 # converting age list to set
34 set_age = set(age)
35 # comparing length of list and length of set
36 print("Is length of age list same of length of age set : ", len(age) == len(set_age))

```

The output of the code is as follows:

```

length of it_companies set : 7
it companies : {'Amazon', 'Google', 'Microsoft', 'Twitter', 'IBM', 'Facebook', 'Apple'}

```

The notebook interface includes a file explorer on the right showing "ML-ASSIGNMENT-1", ".ipynb_checkpoints", "LICENSE", and "ML Assignment 1.ipynb". The status bar at the bottom indicates "Jupyter Server: Local" and "Cell 5 of 11".

```
14 print(""" remove vs discard :
15 Remove method deletes the element from the list if not present it throws Key error where as,
16 Discard method deleted the element from the list otherwise it returns None""")
17 # joining A and B sets
18 print("Join of A and B : ", A.union(B))
19 # intersection of A and B sets
20 print("Intersection of A and B : ",A.intersection(B))
21 # checking if A is subset of B
22 print("Is A subset of B : ", A.issubset(B))
23 # check if A is disjoint of B
24 print("Is A disjoint of B : ", A.isdisjoint(B))
25 # A union B and B union A
26 print("A union B : ", A.union(B))
27 print("B union A : ", B.union(A))
28 # symmetric difference between two sets
29 print("set A difference with set B : ",A.difference(B))
30 # deleting sets A and B
31 A.clear()
32 B.clear()
33 # converting age list to set
34 set_age = set(age)
35 # comparing length of list and length of set
36 print("Is length of age list same of length of age set : ", len(age) == len(set_age))
```

[67] ✓ 0.2s Python

... length of it_companies set : 7
it companies : {'Amazon', 'Google', 'Microsoft', 'Twitter', 'IBM', 'Facebook', 'Apple'}
remove vs discard :
Remove method deletes the element from the list if not present it throws Key error where as,
Discard method deleted the element from the list otherwise it returns None
Join of A and B : {19, 20, 22, 24, 25, 26, 27, 28}
Intersection of A and B : {19, 20, 22, 24, 25, 26}
Is A subset of B : True
Is A disjoint of B : False
A union B : {19, 20, 22, 24, 25, 26, 27, 28}
B union A : {19, 20, 22, 24, 25, 26, 27, 28}
set A difference with set B : set()
Is length of age list same of length of age set : False

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.

```
1 # Question 5
2 # The radius of a circle is 30 meters.
3 r = 30
4 pi = 3.14 # Pi value
5 # calculating area of circle using Pi r square formula
6 _area_of_circle_ = pi * r * r
7 print("Area of given circle :", _area_of_circle_)
8 # calculating circumference of circle
9 _circum_of_circle_ = 2 * pi * r
10 print("Circumference of circle :", _circum_of_circle_)
11 # user input
12 r = float(input("Enter radius : "))
13 # calculating area of circle from user inputs
14 print("You entered radius : " + str(r))
15 area_of_circle = pi * r * r
16 print("Area of your circle is : ", area_of_circle)
```

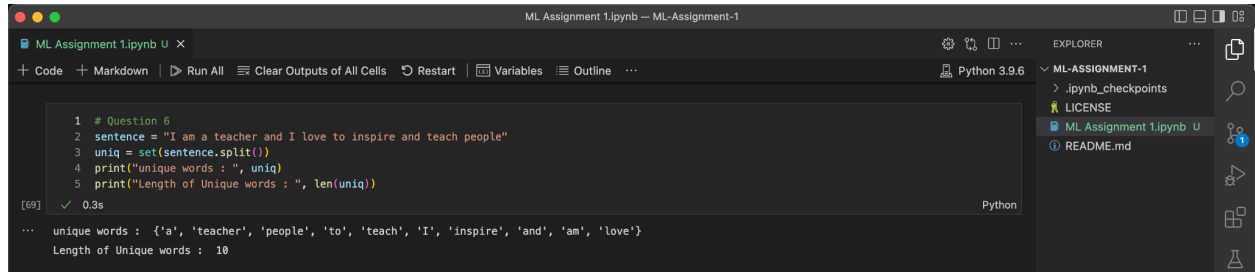
[68] ✓ 6.8s Python

... Area of given circle : 2826.0
Circumference of circle : 188.4
You entered radius : 63.0
Area of your circle is : 12462.660000000002

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 code cell contains the following Python code:

```
1 # Question 6
2 sentence = "I am a teacher and I love to inspire and teach people"
3 uniq = set(sentence.split())
4 print("unique words : ", uniq)
5 print("Length of Unique words : ", len(uniq))
```

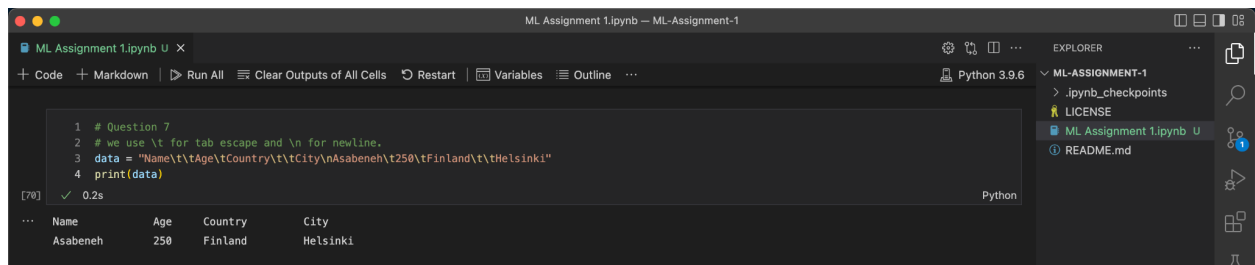
The output of the code is displayed below the cell:

```
unique words : {'a', 'teacher', 'people', 'to', 'teach', 'I', 'inspire', 'and', 'am', 'love'}
Length of Unique words : 10
```

Question 7

Use a tab escape sequence to get the following lines.

Name	Age	Country	City
Asabeneh	250	Finland	Helsinki



A screenshot of a Jupyter Notebook interface. The code cell contains the following Python code:

```
1 # Question 7
2 # we use \t for tab escape and \n for newline.
3 data = "Name\tAge\tCountry\tCity\nAsabeneh\t250\tFinland\tHelsinki"
4 print(data)
```

The output of the code is displayed below the cell, showing the data with tab and newline escape sequences:

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

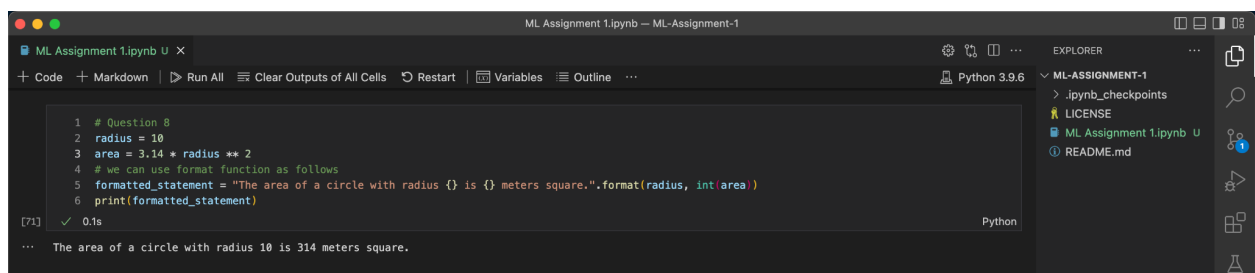
Question 8

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.”

A screenshot of a Jupyter Notebook interface. The top bar shows 'ML Assignment 1.ipynb — ML-Assignment-1'. The left sidebar has 'Code' selected. The main area shows a code cell with the following Python code:

```
1 # Question 8
2 radius = 10
3 area = 3.14 * radius ** 2
4 # we can use format function as follows
5 formatted_statement = "The area of a circle with radius {} is {} meters square.".format(radius, int(area))
6 print(formatted_statement)
```

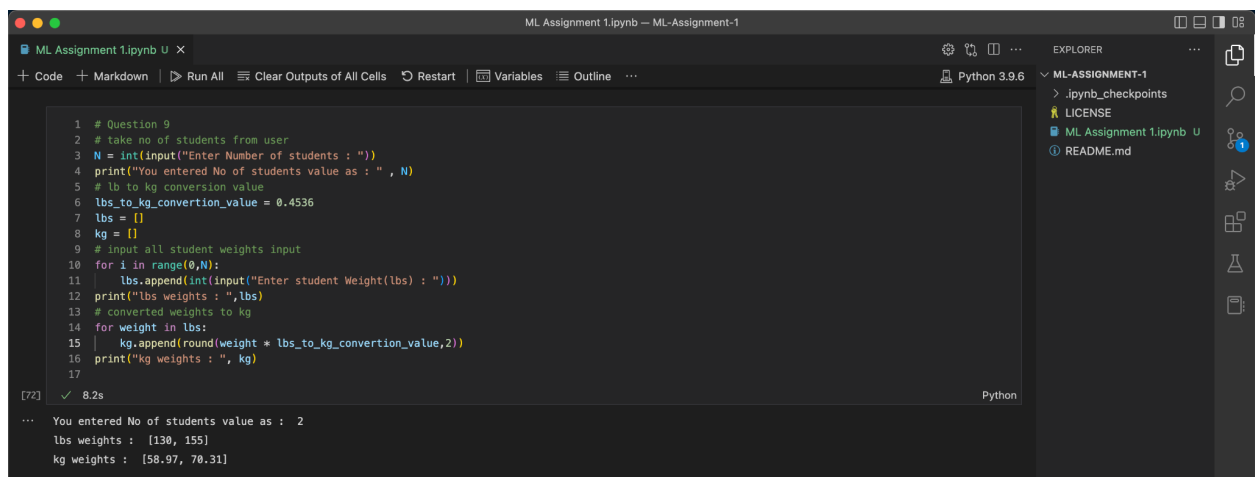
The output of the cell is: 'The area of a circle with radius 10 is 314 meters square.' The right sidebar shows the 'EXPLORER' view with files: 'ML-ASSIGNMENT-1', '.ipynb_checkpoints', 'LICENSE', 'ML Assignment 1.ipynb', and 'README.md'.

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]

A screenshot of a Jupyter Notebook interface. The top bar shows 'ML Assignment 1.ipynb — ML-Assignment-1'. The left sidebar has 'Code' selected. The main area shows a code cell with the following Python code:

```
1 # Question 9
2 # take no of students from user
3 N = int(input("Enter Number of students : "))
4 print("You entered No of students value as : ", N)
5 # lb to kg conversion value
6 lbs_to_kg_conversion_value = 0.4536
7 lbs = []
8 kg = []
9 # input all student weights input
10 for i in range(0,N):
11     lbs.append(int(input("Enter student Weight(lbs) : ")))
12     print("lbs weights : ",lbs)
13 # converted weights to kg
14 for weight in lbs:
15     kg.append(round(weight * lbs_to_kg_conversion_value,2))
16     print("kg weights : ", kg)
17
```

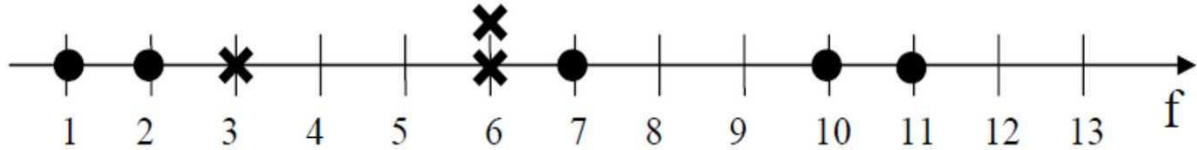
The output of the cell is:

```
You entered No of students value as : 2
lbs weights : [130, 155]
kg weights : [58.97, 70.31]
```

The right sidebar shows the 'EXPLORER' view with files: 'ML-ASSIGNMENT-1', '.ipynb_checkpoints', 'LICENSE', 'ML Assignment 1.ipynb', and 'README.md'.

Question 10

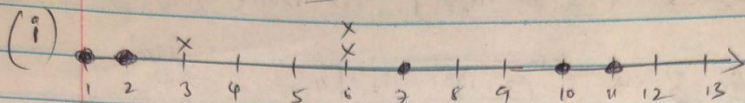
The diagram below shows a dataset with 2 classes and 8 data points, each with only one feature value, labeled f . Note that there are two data points with the same feature value of 6. These are shown as two x 's one above the other. Provide stepwise mathematical solutions, do not write code for it.



1. Divide this data equally into two parts. Use the first part as training and the 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

Machine Learning Assignment #1

Question 10



The given data can be represented by using tabular format as:

data points	1	2	3	6	6	7	10	11
y	0	0	1	1	1	0	0	0

Splitting into two Equal halves

$$x_{\text{train}} = [1, 2, 6, 7]$$

$$y_{\text{train}} = [0, 0, 1, 0]$$

$$x_{\text{test}} = [3, 6, 10, 11]$$

$$y_{\text{test}} = [1, 1, 0, 0]$$

If using N value as 3, we can predict the output as with using the Euclidean distance formula.

$$\text{Euclidean dist} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (\text{for 2-D space})$$

We use the above distance formula for N value 3 i.e., 3 nearest (closest) points.

$$x_{\text{test}} = [3, 6, 10, 11] ; x_{\text{train}} = [1, 2, 6, 7]$$

$$\begin{aligned} \text{for } 3 &: \rightarrow \left. \begin{aligned} \sqrt{(3-1)^2} &= \sqrt{4} = 2 \rightarrow 0 \\ \sqrt{(3-2)^2} &= \sqrt{1} = 1 \rightarrow 0 \\ \sqrt{(3-6)^2} &= \sqrt{9} = 3 \rightarrow 1 \\ \sqrt{(3-7)^2} &= \sqrt{16} = 4 \rightarrow 0 \end{aligned} \right\} \begin{array}{l} \text{These are} \\ \text{the 3 closest} \end{array} \end{aligned}$$

$$\therefore y_{\text{pred}}[0] = 0$$

$$\begin{aligned} \text{for } 6 &: \rightarrow \left. \begin{aligned} \sqrt{(6-1)^2} &= \sqrt{25} = 5 \rightarrow 0 \\ \sqrt{(6-2)^2} &= \sqrt{16} = 4 \rightarrow 0 \\ \sqrt{(6-6)^2} &= \sqrt{0} = 0 \rightarrow 1 \\ \sqrt{(6-7)^2} &= \sqrt{1} = 1 \rightarrow 0 \end{aligned} \right\} \begin{array}{l} \text{These 3} \\ \text{are closest} \end{array} \end{aligned}$$

$$\therefore y_{\text{pred}}[1] = 0$$

$$\begin{aligned} \text{for } 10 &: \rightarrow \left. \begin{aligned} \sqrt{(10-1)^2} &= \sqrt{81} = 9 \rightarrow 0 \\ \sqrt{(10-2)^2} &= \sqrt{64} = 8 \rightarrow 0 \\ \sqrt{(10-6)^2} &= \sqrt{16} = 4 \rightarrow 1 \\ \sqrt{(10-7)^2} &= \sqrt{9} = 3 \rightarrow 0 \end{aligned} \right\} \begin{array}{l} \text{These 3} \\ \text{are closest} \end{array} \end{aligned}$$

$$\therefore y_{\text{pred}}[2] = 0$$

for 11: $\sqrt{(11-1)^2} = \sqrt{100} = 10 \xrightarrow{y_{train}} 0$

$\sqrt{(11-2)^2} = \sqrt{81} = 9 \rightarrow 0$

$\sqrt{(11-6)^2} = \sqrt{25} = 5 \rightarrow 1$

$\sqrt{(11-7)^2} = \sqrt{16} = 4 \rightarrow 0$

} These 3 are closest

$y_{pred}[3] = 0$

$y_{pred} = [0, 0, 0, 0]$

$y_{test} = [1, 1, 0, 0]$

(ii) Confusion matrix for the above data is

$\eta = 1$

	Predicted NO	Predicted Yes	
Actual NO	TN 2	FP 0	2
Actual Yes	FN 2	TP 0	2
	4	0	\downarrow $\Rightarrow N=4$

Accuracy = Accuracy means how often the classifier is correct.

$$\bullet \frac{TP + TN}{\text{total}} = \frac{0 + 2}{4} = 0.5 = (50\%)$$

Sensitivity = when it's actually Yes, how often it is predicting Yes.

$$\bullet \frac{TP}{TP + FN} = \frac{0}{0 + 2} = 0 \quad (0\%)$$

Specificity = when it is actually No, how often it is predicting No.

$$\bullet \frac{TN}{TN + FP} = \frac{2}{2 + 0} = 1 \quad (100\%)$$