Assignment 1

Video Link:

https://drive.google.com/file/d/1-QywnJS0XTr9auxDLOUGttuKyuBGNsqL/view?usp=share_link

Github Link:

https://github.com/sxn01020/ML_Assignment_1.git

Question1:

README.md | Question1.py | Question2.py | Question3.py | Question4.py | Question5.py | Question6.py | Question7.py | Question8.py | Question9.py

```python
import math #Importing math package

#list of 10 student's ages
ages= [19, 22, 19, 24, 20, 25, 26, 24, 25, 24]

#sorting the list of ages
ages.sort()
print("Sorted ages list is: ",ages)

#Finding the minimum age from the list of ages
min_age=min(ages)
print("Minimum age from the sorted list is: ",min_age)

#Finding the maximum age from the list of ages
max_age=max(ages)
print("Maximum age from the sorted list is: ",max_age)

#Adding minimum and maximum ages again to the list of ages
ages.append(min_age)
ages.append(max_age)
print("Student's ages list after adding minimum and maximum ages back to the list: ",ages)

#Finding the median age from the list
length=len(ages)
#Finding the median age from the list when the length of the list is odd
if length%2!=0:
    median=length//2  #length of the list
    median=math.ceil(median)  #calculating the index of the median
    print("Median age from the list of ages is: ",ages[median])  #printing the median age from the list when the length of the list is odd

# Finding the median age from the list when the length of the list is even
else:
    median=length//2  #length of the list
    print("Median age from the list of ages is: ",(ages[median-1]+ages[median])/2)  #printing the median age from the list when the length of the list is eve

#Finding the average age from the list of ages
avg=sum(ages)/len(ages)
print("Average age from the list og ages is: ",avg)

#Finding the range of the ages
range=max_age-min_age
print("Range of the ages is: ",range)
```

Output:

Run: Question1

```
C:\Users\nalab\anaconda3\python.exe "C:\Users\nalab\Documents\Important Docs\UCM\Spring 2023\Github Desktop\ML_Assignment_1\Question1.py"
Sorted ages list is:  [19, 19, 20, 22, 24, 24, 24, 25, 25, 26]
Minimum age from the sorted list is:  19
Maximum age from the sorted list is:  26
Student's ages list after adding minimum and maximum ages back to the list:  [19, 19, 20, 22, 24, 24, 24, 25, 25, 26, 19, 26]
Median age from the list of ages is:  24.0
Average age from the list og ages is:  22.75
Range of the ages is:  7

Process finished with exit code 0
```

Explanation:

1. Given a list called ages, I sorted the list.
2. Found the maximum and minimum age from the age list.
3. Added min and max age to the age list again.
4. Found median from the list of ages based on the length of the list. If it is odd median will be the center value from the list and if it is even then the median will be the average of center two values.

5. Found the average age from the list.
6. Found the range of ages from the age list using the maximum and minimum age.

Question2:

README.md | Question1.py | Question2.py | Question3.py | Question4.py | Question5.py | Question6.py | Question7.py | Question

```python
#Creating an empty dictionary called dog
dog={}

#Adding data to the dog dictionary
dog["name"]="Flash"
dog["color"]="brown"
dog["breed"]="beagle"
dog["legs"]=4
dog["age"]=2
print("Dog dictionary after adding keys and values is: ",dog)

#Creating a student dictionary and adding the data
student={}
student["first_name"]="Sriram"
student["last_name"]="Nalabolu"
student["geneder"]="male"
student["age"]=25
student["marital status"]="single"
student["skills"]=["python","big data","devops","cloud"]
student["country"]="United States of America"
student["city"]="Kansas"
student["address"]="6600 W 140th Street"
print("Student dictionary after adding keys and values is: ",student)

#Printing the length of student dictionary
length=len(student)
print("Length of student dictionary is: ",length)

#Printing the value of skills and checking the data type of skills
print("Value of skills is: ",student["skills"])
print("Data Type of skills is: ",type(student["skills"]))

#Modifying the skills values by adding two skills
student["skills"].append("java")
student["skills"].append("ansible")
print("Skills values after adding two skills is: ",student["skills"])

#Printing the Dog dictionary keys and values as a list
print("Dog keys as a list is: ",dog.keys())
print("Dog values as a list is: ",dog.values())

#Printing the Student dictionary keys and values as a list
print("Student keys as a list is: ",student.keys())
print("Student values as a list is: ",student.values())
```

Output:

Question2

```
C:\Users\nalab\anaconda3\python.exe "C:\Users\nalab\Documents\Important Docs\UCM\Spring 2023\Github Desktop\ML_Assignment_1\Question2.py"
Dog dictionary after adding keys and values is:  {'name': 'Flash', 'color': 'brown', 'breed': 'beagle', 'legs': 4, 'age': 2}
Student dictionary after adding keys and values is:  {'first_name': 'Sriram', 'last_name': 'Nalabolu', 'geneder': 'male', 'age': 25, 'marital status': 'single', 'skills': ['pytho
Length of student dictionary is:  9
Value of skills is:  ['python', 'big data', 'devops', 'cloud']
Data Type of skills is:  <class 'list'>
Skills values after adding two skills is:  ['python', 'big data', 'devops', 'cloud', 'java', 'ansible']
Dog keys as a list is:  dict_keys(['name', 'color', 'breed', 'legs', 'age'])
Dog values as a list is:  dict_values(['Flash', 'brown', 'beagle', 4, 2])
Student keys as a list is:  dict_keys(['first_name', 'last_name', 'geneder', 'age', 'marital status', 'skills', 'country', 'city', 'address'])
Student values as a list is:  dict_values(['Sriram', 'Nalabolu', 'male', 25, 'single', ['python', 'big data', 'devops', 'cloud', 'java', 'ansible'], 'United States of America', '

Process finished with exit code 0
```

Explanation:

1. Created an empty dictionary called dog.
2. Added values to it like name, color, breed, legs, age.
3. Created another dictionary named Student and added first_name, last_name, gender, age, marital status, skills, country, city and address to it.
4. Found length of the student dictionary.
5. Using Student[skills] found the skills of the student.
6. Modified the skills values by adding java and ansible to it.
7. Printed the list of keys and values separately for both the dictionaries.

Question3:

README.md | Question1.py | Question2.py | Question3.py | Question4.py | Question5.py | Question6.py | Question7.py | Question

```python
#Creating tuples containing names of sisters and brothers
brothers=("lakshman","bharath")
sisters=("sravani","harivina")
print("My Brothers are: ",brothers)
print("My Sisters are: ",sisters)

#Joining brothers and sisters tuples into siblings tuple
siblings=brothers+sisters
print("My Siblings are: ",siblings)

#Calculating the total number of siblings
print("Total number of siblings I have are: ",len(siblings))

#Adding mother, father and sublings to family_members tuple
family_members=("Ravi","Gayathri")+siblings
print("My family members are: ",family_members)
```

Output:

Run: Question3

```
C:\Users\nalab\anaconda3\python.exe "C:\Users\nalab\Documents\Important Docs\UCM\Spring 2023\Github Desktop\ML_Assignment_1\Question3.py"
My Brothers are:  ('lakshman', 'bharath')
My Sisters are:  ('sravani', 'harivina')
My Siblings are:  ('lakshman', 'bharath', 'sravani', 'harivina')
Total number of siblings I have are:  4
My family members are:  ('Ravi', 'Gayathri', 'lakshman', 'bharath', 'sravani', 'harivina')

Process finished with exit code 0
```

Explanation:

1. Created tuples with names of sisters and brothers.
2. Joined brothers and sisters tuples to the new tuple called siblings.
3. By using the length function found the number of siblings.
4. Created a family_members tuple and added siblings tuple and added parents names to it.

Question4:

README.md | Question1.py | Question2.py | Question3.py | Question4.py | Question5.py | Question6.py | Question7.py | Questio

```python
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]

#Finding the length of it_companies
print("Length of it_companies set is: ",len(it_companies))

#Adding Twitter to it_companies
it_companies.add("Twitter")
print("it_companies set after adding twitter to it is: ",it_companies)

#Inserting multiple it companies to the set it_companies
it_companies.update({"Tesla","Cap Gemini","Tech Mahindra","Accenture"})
print("it_companies set after adding multiple companies to it is: ",it_companies)

#Removing one of the companies from the set it_companies
it_companies.remove("Cap Gemini")
print("it_companies set after removing Cap Gemini from the set is: ",it_companies)

#Removing one of the companies from the set it_companies
it_companies.discard("Cap Gemini")
#Discard will not throw an error even if the element is not present in the set whereas Remove will throw an error.

#Joining A and B sets
print("Set after joining A and B sets is: ",A.union(B))

#Finding A intersection B
print("Set after intersection of A and B sets is: ",A.intersection(B))

#Finding if A is a subset of B
print("A is subset of B: ",A.issubset(B))

#Finding if A and B are disjoint sets
print("A is disjoint of B: ",A.isdisjoint(B))

#Joining A with B and B with A
print("set after joining A with B: ",A.union(B))
print("set after joining B with A: ",B.union(A))

#Finding the symmetric difference between A and B
print("Symmetric difference between A and B is: ",A.symmetric_difference(B))

#Deleting the sets completely
print(A.clear())
print(B.clear())
print(it_companies.clear())

#Converting the age list to set and comparing the length of age set and age list
print("Age set after converting from List to set is: ",set(age))
print("Length of age set: ",len(set(age)))
print("Length of age list: ",len(age))
```

Output:

Run: Question4

```
it_companies set after adding twitter to it is:  {'Apple', 'Oracle', 'Facebook', 'IBM', 'Amazon', 'Google', 'Microsoft', 'Twitter'}
it_companies set after adding multiple companies to it is:  {'Accenture', 'Oracle', 'Amazon', 'Cap Gemini', 'Twitter', 'Apple', 'Facebook', 'Tesla', 'Tech Mahindra', 'IBM', 'Goog
it_companies set after removing Cap Gemini from the set is:  {'Accenture', 'Oracle', 'Amazon', 'Twitter', 'Apple', 'Facebook', 'Tesla', 'Tech Mahindra', 'IBM', 'Google', 'Microso
Set after joining A and B sets is:  {19, 20, 22, 24, 25, 26, 27, 28}
Set after intersection of A and B sets is:  {19, 20, 22, 24, 25, 26}
A is subset of B:  True
A is disjoint of B:  False
set after joining A with B:  {19, 20, 22, 24, 25, 26, 27, 28}
set after joining B with A:  {19, 20, 22, 24, 25, 26, 27, 28}
Symmetric difference between A and B is:  {27, 28}
None
None
None
Age set after converting from List to set is:  {19, 22, 24, 25, 26}
Length of age set:  5
Length of age list:  8

Process finished with exit code 0
```

Explanation:

1. Added twitter company to the list of given it_companies.
2. Added multiple companies to the list of it_companies.
3. Removed a company called Cap Gemini from the list of it_companies.
4. Explained the difference between remove and discard as Discard will not throw an error even if the element is not present in the set whereas Remove will throw an error.
5. Joined the given sets A and B using Union.
6. Found the intersection of the sets A and B using the intersection method.
7. Found if A is subset of B.
8. Found if A is disjoint of B.
9. Found the symmetric difference between the sets A and B.
10. Deleted all the sets using clear.
11. Converted the age list to set and compared the lengths of age set and age list.

Question5:

README.md | Question1.py | Question2.py | Question3.py | Question4.py | Question5.py | Question6.py | Question7.py | Question

```python
import math #importing math package

radius=30 #initalizing radius as 30

#Calculating the area of circle and storing it in _area_of_circle variable
_area_of_circle=int((math.pi)*(radius**2))
print("Area of circle is: ",_area_of_circle)

#Calculating the perimeter of circle and storing it in _circum_of_circle
_circum_of_circle=int(2*math.pi*radius)
print("Circumference of circle is: ",_circum_of_circle)

#Taking the user input for radius and storing it in radius variable
radius=int(input("Enter the radius of circle: "))

#Calculating the area using the user's input radius and storing it in _area_of_circle variable
_area_of_circle=int((math.pi)*(radius**2))
print("Area of the circle using user's input radius is: ",_area_of_circle)
```

Output:

Run: Question5

```
C:\Users\nalab\anaconda3\python.exe "C:\Users\nalab\Documents\Important Docs\UCM\Spring 2023\Github Desktop\ML_Assignment_1\Question5.py"
Area of circle is:  2827
Circumference of circle is:  188
Enter the radius of circle: 20
Area of the circle using user's input radius is:  1256

Process finished with exit code 0
```

Explanation:

1. Calculated the area of circle with radius 30 and was assigned it to the variable named _area_of_circle_ .
2. Calculated the circumference of the circle and assigned it the variable named _circum_of_circle_ .
3. Again taken radius as input from user and found area of the circle using that radius.

Question6:

README.md | Question1.py | Question2.py | Question3.py | Question4.py | Question5.py | Question6.py | Question7.py | Questio

```
1  sentence="I am a teacher and I love to inspire and teach people" #input sentence
2
3  #Calculating the unique words from the sentence
4  words=sentence.split() #splitting the sentence into a list of words
5  print("List of words after splitting the sentence is: ",words)
6  print("Number of unique words from the given sentence is: ",len(set(words)))
```

Output:

Run: Question6

```
C:\Users\nalab\anaconda3\python.exe "C:\Users\nalab\Documents\Important Docs\UCM\Spring 2023\Github Desktop\ML_Assignment_1\Question6.py"
List of words after splitting the sentence is:  ['I', 'am', 'a', 'teacher', 'and', 'I', 'love', 'to', 'inspire', 'and', 'teach', 'people']
Number of unique words from the given sentence is:  10

Process finished with exit code 0
```

Explanation:

1. Splitted the sentence into a list of words.
2. Converted the list to set and found the number of unique words from the sentence by applying length method to this list of words.

Question7:

README.md | Question1.py | Question2.py | Question3.py | Question4.py | Question5.py | Question6.py | Question7.py | Questic

```
1  #Using the tab space to print the strings
2  print("Name\t\tAge\tCountry\tCity")
3  print("Asabeneh\t250\tFinland\tHelsinki")
```

Output:

Run: Question7

```
C:\Users\nalab\anaconda3\python.exe "C:\Users\nalab\Documents\Important Docs\UCM\Spring 2023\Github Desktop\ML_Assignment_1\Question7.py"
Name        Age Country City
Asabeneh    250 Finland Helsinki

Process finished with exit code 0
```

Explanation:

1. Printing the strings using the tab space escape character as shown in above output.

Question8:

README.md | Question1.py | Question2.py | Question3.py | Question4.py | Question5.py | Question6.py | Question7.py | Questic

```
1  #Initializing the radius of the circle
2  radius=10
3
4  #Calculating the area of the circle
5  area=3.14*radius**2
6
7  #Using the String formatting method to display the area and radius of the circle
8  print("The area of a circle with radius {} is {} meters square.".format(radius,int(area)))
```

Output:

```
Run:  Question8
C:\Users\nalab\anaconda3\python.exe "C:\Users\nalab\Documents\Important Docs\UCM\Spring 2023\Github Desktop\ML_Assignment_1\Question8.py"
The area of a circle with radius 10 is 314 meters square.

Process finished with exit code 0
```

Explanation:

1. Initializing the radius of circle and calculating the area of circle using that radius.
2. Printing the radius and area of circle using the string formatting method.

Question9:

README.md | Question1.py | Question2.py | Question3.py | Question4.py | Question5.py | Question6.py | Question7.py | Questio

```python
weights=[] #creating weights list
kilograms=[] #creating kilograms list

#Taking the number of students from user
N=int(input("Enter the number of students: "))

#Taking the weights of these students from user in lb
for i in range(N):
    weights.append(int(input("Enter the weight of student{}: ".format(i+1))))

print("Weight of students in lb is: ",weights) #printing the list of weights in lb

#Calculating kilograms using weights in lb and appending them to a list
for i in weights:
    kilograms.append(round(float(i/2.2046),2))

#printing the list of weights in kilograms
print("List of weights of students in kilograms is: ",kilograms)
```

Output:

```
Run:  Question9
C:\Users\nalab\anaconda3\python.exe "C:\Users\nalab\Documents\Important Docs\UCM\Spring 2023\Github Desktop\ML_Assignment_1\Question9.py"
Enter the number of students: 4
Enter the weight of student1: 150
Enter the weight of student2: 155
Enter the weight of student3: 145
Enter the weight of student4: 148
Weight of students in lb is:  [150, 155, 145, 148]
List of weights of students in kilograms is:  [68.04, 70.31, 65.77, 67.13]

Process finished with exit code 0
```

Explanation:

1. Taken the input as number of students from the user.
2. Also taken Weights as input in lb from the user and calculated in terms of kgs.

Question 10

→ From the given Question, there are two labels, ● & X. Considering or classes ● as 1 and X as 0. So the feature values and labels, or classes are as below.

f	class
1	1
2	1
3	0
6	0
6	0
7	1
10	1
11	1

(1)

Since we need to divide the data equally into two parts. Splitting the data into train & test data.

Train data → {1, 2, 3, 6}

class → {1, 1, 0, 0}

Test data → {6, 7, 10, 11}

class → {0, 1, 1, 1}

Confusion matrix for Test data will be as

Actual ↓ \ Predicted →	0	1
0	TN	FP
1	FN	TP

Using Euclidean distance, find the distance between the Test sample 6 and Train samples 1, 2, 3, 6, we get

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

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

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

$d_4 = \sqrt{(6-6)^2} = 0$

So ~~Train~~ Using 3-NN classifier, train samples nearer to the Test sample 6 are 6, 3, 2 & their labels/classes are 0, 0, 1. Taking the maximum out of them, the predicted output is 0. So ~~it is~~ Actual is 0 & Predicted is 0. So it is TN

Illy for Test samples 7, 10 & 11 we get

for 7 → 3-NN are 0, 0, 1 So FN Since actual is 1 & Predicted is 0

10 → 3-NN are 0, 0, 1 So FN Since actual is 1 & Predicted is 0

11 → 3-NN are 0, 0, 1 So FN Since actual is 1 & Predicted is 0

(2) So, Confusion matrix is

	0	1
0	1	0
1	3	0

⇒ TN = 1
FP = 0
FN = 3
TP = 0

So, Accuracy $= \frac{TP+TN}{P+N} = \frac{TP+TN}{TN+FP+FN+TP} = \frac{1}{4} = 0.25$

Sensitivity $= \frac{TP}{TP+FN} = \frac{TP}{P} = \frac{0}{0+3} = 0$

Specificity $= \frac{TN}{FP+TN} = \frac{TN}{N} = \frac{1}{0+1} = 1$