

Machine Learning Assignment 1

Rishma Reddy Nalla
700752

Video Link: <https://youtu.be/MDU27G32sxxk>

Github Link: <https://github.com/RishmaReddy-Nalla/CS-5710/tree/main/Assignment1>

Question 1:

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

For this array find

Sort the array and finding max and min values

Adding min and max values to sorted array

Finding median age

Finding average age

Finding range of values

Code:

```
import math
# given list of 10 students age
ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24]

# 1) sorting list values and finding min, max values of ages
## step 1: Sorting list values
sorted_ages = sorted(ages)
print("Sorted List values: ", sorted_ages)

## step 2: Finding min and max values
min_age = min(sorted_ages)
max_age = max(sorted_ages)
print(f"Maximum Age is {min_age} and Maximum age is {max_age}" )

## Adding min and max values to list again
sorted_ages.extend([min_age, max_age])
print("list after adding min and max values: ", sorted_ages)

## Finding median age of list
sorted_ages = sorted(sorted_ages)
n = math.floor(len(sorted_ages))
median = (sorted_ages[int(n/2)] + sorted_ages[int(n/2)+1])/2
print(median)

## Finding Average age
avg_age = sum(sorted_ages)/len(sorted_ages)
print(avg_age)

## finding Range of list
range_ages = max(sorted_ages) - min(sorted_ages)
print(range_ages)
```

Result:

```
Sorted List values: [19, 19, 20, 22, 24, 24, 24, 25, 25, 26]
Maximum Age is 19 and Maximum age is 26
list after adding min and max values: [19, 19, 20, 22, 24, 24, 24, 25, 25, 26, 19, 26]
24.0
22.75
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

Code:

```
## creating an empty dictionary named dog
dog = {}
dog.update({'Name': 'Scoopy', 'Color': 'Brown', 'Breed': 'German Shepard', 'Legs': 4, 'Age': 3})
student = {'first_name': 'Jhon', 'last_name': 'Pal', 'gender': 'Male', 'age': 22, 'marital status': 'Single', 'skills': ['python', 'ruby', 'node.js'], 'country': 'USA', 'city': 'St.petersburg', 'address': '4532 west, 124th st, kp town, 65342'}

## printing length of student dictionary
print("Length of student dictionary:", len(student))

## listing values of skills and checking data type
print(student['skills'], "Data type: ", type(student['skills']))

## modifying skills in student dictionary
student['skills'].extend(['java', 'html'])
print(student)

## Printing dictionary keys as list\
print(list(student.keys()))

## Printing student dictionary values as list
print(list(student.values()))
```

Result:

```
Length of student dictionary: 9
['python', 'ruby', 'node.js'] Data type: <class 'list'>
{'first_name': 'Jhon', 'last_name': 'Pal', 'gender': 'Male', 'age': 22, 'marital status': 'Single', 'skills': ['python', 'ruby', 'node.js', 'java', 'html'], 'country': 'USA', 'city': 'St.petersburg', 'address': '4532 west, 124th st, kp town, 65342'}
['Jhon', 'Pal', 'Male', 22, 'Single', ['python', 'ruby', 'node.js', 'java', 'html'], 'USA', 'St.petersburg', '4532 west, 124th st, kp town, 65342']
```

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

Code:

```
## create brothers and sisters tuple
brothers = ('Jhon', 'Jimmy', 'Michel')
sisters = ('Lara', 'Roosi')
print("Brothers: ",brothers,"Sisters: ", sisters)

## adding brothers and sisters into sublings
siblings = brothers + sisters
print("Concatnation of brothers and sisters: ",siblings)

# length of siblings
print("No of Siblings :", len(siblings))

# adding father mother and siblings to family tuple
family = siblings + ('Mike', 'Joo')
print("Family tuple: ",family)
```

Result:

```
✓ Brothers: ('Jhon', 'Jimmy', 'Michel') Sisters: ('Lara', 'Roosi')
Concatnation of brothers and sisters: ('Jhon', 'Jimmy', 'Michel', 'Lara', 'Roosi')
No of Siblings : 5
Family tuple: ('Jhon', 'Jimmy', 'Michel', 'Lara', 'Roosi', 'Mike', 'Joo')
```

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.

Code:

```

# Creating the dictionary It_companies
it_companies = {'Facebook', 'Google', 'Microsoft', 'Apple', 'IBM', 'Oracle', 'Amazon'}

#creating A&B sets
A = {19, 22, 24, 20, 25, 26}
B = {19, 22, 20, 25, 26, 24, 28, 27}

#created list age
age = [22, 19, 24, 25, 26, 24, 25, 24]

# printing length of it_companies dictionary
print("Length of it_companies: ", len(it_companies))

# adding Twitter to it_companies dictionary
it_companies.add('Twitter')
print("Added Twitter company to it_companies: ", it_companies)

# adding multiple companies to it_companies dictionary
it_companies.update({'Capital One', 'X', 'TCS'})
print("Added multiple IT companies: ", it_companies)

# removing Capital one from it_companies dict
it_companies.remove('Capital One')
print(it_companies)

# Joining two sets A & B
A.union(B)
print("Added A & B sets: ", A)

# Intersection b/w A and B
intersectionAB = A.intersection(B)
print("A intersection B: ", intersectionAB)

# finding subset and disjoint set
print(A.issubset(B))
print(A.isdisjoint(B))

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

B.union(A)
print("Joining B with A: ", B)

# finding symmetric difference
A.symmetric_difference(B)
print(A)

# deleting set A and set B
del(A) # this will delete set A completely
del(B) # this will delete set B completely

# converting age list to set, then compared list and set of age
age_set = set(age)
print("Type of Age_set: ", type(age_set))

print(len(age) == len(age_set))

```

Result:

```
Length of it_companies: 7
Added Twitter company to it_companies: {'Google', 'Twitter', 'Facebook', 'Oracle', 'IBM', 'Amazon', 'Microsoft', 'Apple'}
Added multiple IT companies: {'Oracle', 'Amazon', 'TCS', 'Google', 'Twitter', 'X', 'Facebook', 'Capital One', 'IBM', 'Microsoft', 'Apple'}
{'Oracle', 'Amazon', 'TCS', 'Google', 'Twitter', 'X', 'Facebook', 'IBM', 'Microsoft', 'Apple'}
Added A & B sets: {19, 20, 22, 24, 25, 26}
A intersection B: {19, 20, 22, 24, 25, 26}
True
False
Joining set A with B: {19, 20, 22, 24, 25, 26}
Joining B with A: {19, 20, 22, 24, 25, 26, 27, 28}
{19, 20, 22, 24, 25, 26}
Type of Age_set: <class '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.

Code & Results:

```
[9] import math
    # calculate area of circle with static radius
    Radius = 30
    _area_of_circle_ = math.pi * (Radius**2)
    print(_area_of_circle_)

2827.4333882308138
```

```
[10] # calculating circumference of circle
    _circum_of_circle_ = 2* math.pi * Radius
    print(_circum_of_circle_)

188.49555921538757
```

```
[12] # calculating area of circle with user defined radius
    user_radius = int(input("Enter Radius for circle: "))
    area_of_circle = math.pi * (user_radius**2)
    print(area_of_circle)

Enter Radius for circle: 15
706.8583470577034
```

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.

Code & Result:

```
[13] # finding count of unique words in given string
      string = "I am a teacher and I love to inspire and teach people"
      set_words = set(string.split(" "))
      print("No of unique words are: ", len(set_words))
```

```
➦ No of unique words are: 10
```

Question 7:

Use a tab escape sequence to get the following lines.

Name	Age	Country	City
Asabeneh	250	Finland	Helsinki

Code & Results:

```
[14] # using \t (tab spacing) and \n (new line) to print tab seperated values
      print("Name\tAge\tCountry\tCity\nAsabeneh\t250\tFinland\tHelsinki")
```

```
➦ 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.”

Code & Result:

```
[15] # printing text using string format method
      radius = 10
      area = 3.14 * radius**2
      print("The area of circle with {} is {} meters square".format(radius, int(area)))
```

```
➦ The area of circle with 10 is 314 meters square
```

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]

Code & Result:

```
# converting weight in lbs to kgs by taking user input for number students and their respective weights in lbs
n = int(input("Enter no of students: "))
lbs_weights = []
count = 0
while count < n:
    lb_w = int(input(f"Enter student {count+1} weight: "))
    lbs_weights.append(lb_w)
    count += 1

print("Weights in Lbs: ", lbs_weights)

kgs_weights = []
for i in lbs_weights:
    kg_w = i * 0.453592
    kgs_weights.append(round(kg_w, 2))
print("Weights in Kgs: ", kgs_weights)
```

Enter no of students: 4
Enter student 1 weight: 150
Enter student 2 weight: 155
Enter student 3 weight: 145
Enter student 4 weight: 148
Weights in Lbs: [150, 155, 145, 148]
Weights in Kgs: [68.04, 70.31, 65.77, 67.13]