SUMMER 2024 : CS5710
MACHINE LEARNING ICP –1

NAME : NISHANTH KURUGUNTLA
STUDENT ID : 700766566

GITHUB LINK: https://github.com/nishath0708/machine-learning-ass-1

VIDEO LINK :
https://drive.google.com/file/d/1XZcZXTtABJJKMTvO68el3ZeF1FbSHFwA/view?usp=drive_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)

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

# Sort the list and find the min and max age
sorted_ages = sorted(ages)
min_age = sorted_ages[0]
max_age = sorted_ages[-1]

# Add the min age and the max age again to the list
ages.extend([min_age, max_age])

# Find the median age
n = len(ages)
if n % 2 == 0:
    median_age = (sorted_ages[n//2 - 1] + sorted_ages[n//2]) / 2
else:
    median_age = sorted_ages[n//2]

# Find the average age
average_age = sum(ages) / len(ages)

# Find the range of the ages
age_range = max_age - min_age

# Print results
print("Sorted Ages:", sorted_ages)
print("Min Age:", min_age)
print("Max Age:", max_age)
print("Median Age:", median_age)
print("Average Age:", average_age)
print("Age Range:", age_range)
```

```
# Find the range of the ages
age_range = max_age - min_age

# Print results
print("Sorted Ages:", sorted_ages)
print("Min Age:", min_age)
print("Max Age:", max_age)
print("Median Age:", median_age)
print("Average Age:", average_age)
print("Age Range:", age_range)
```

```
Sorted Ages: [19, 19, 20, 22, 24, 24, 24, 25, 25, 26]
Min Age: 19
Max Age: 26
Median Age: 24.0
Average Age: 22.75
Age Range: 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.

```
# Create an empty dictionary called dog
dog = {}

# Add name, color, breed, legs, age to the dog dictionary
dog["name"] = "Fido"
dog["color"] = "Brown"
dog["breed"] = "Labrador"
dog["legs"] = 4
dog["age"] = 3

# Create a student dictionary
student = {
    "first_name": "John",
    "last_name": "Doe",
    "gender": "Male",
    "age": 25,
    "marital_status": "Single",
    "skills": ["Python", "Java", "SQL"],
    "country": "USA",
    "city": "New York",
    "address": "123 Main St"
}

# Get the length of the student dictionary
student_length = len(student)

# Get the value of skills and check the data type
skills_value = student["skills"]
skills_type = type(skills_value)

# Modify the skills values by adding one or two skills
```

```python
# Get the value of skills and check the data type
skills_value = student["skills"]
skills_type = type(skills_value)

# Modify the skills values by adding one or two skills
student["skills"].extend(["JavaScript", "HTML"])

# Get the dictionary keys as a list
student_keys = list(student.keys())

# Get the dictionary values as a list
student_values = list(student.values())

# Print the results
print("Dog Dictionary:", dog)
print("Student Dictionary:", student)
print("Length of Student Dictionary:", student_length)
print("Type of Skills Value:", skills_type)
print("Modified Skills:", student["skills"])
print("Keys of Student Dictionary:", student_keys)
print("Values of Student Dictionary:", student_values)
```

```
Dog Dictionary: {'name': 'Fido', 'color': 'Brown', 'breed': 'Labrador', 'legs': 4, 'age': 3}
Student Dictionary: {'first_name': 'John', 'last_name': 'Doe', 'gender': 'Male', 'age': 25, 'marital_status': 'Single', 'skills': ['Python', 'Java', 'SQL', 'JavaScript', 'HTML'], 'cour
Length of Student Dictionary: 9
Type of Skills Value: <class 'list'>
Modified Skills: ['Python', 'Java', 'SQL', 'JavaScript', 'HTML']
Keys of Student Dictionary: ['first_name', 'last_name', 'gender', 'age', 'marital_status', 'skills', 'country', 'city', 'address']
Values of Student Dictionary: ['John', 'Doe', 'Male', 25, 'Single', ['Python', 'Java', 'SQL', 'JavaScript', 'HTML'], 'USA', 'New York', '123 Main St']
```

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.

```python
# Create a tuple containing names of your sisters and your brothers
sisters = ("Alice", "Emily")
brothers = ("Jack", "Ryan", "Michael")

# Join brothers and sisters tuples and assign it to siblings
siblings = brothers + sisters

# How many siblings do you have?
num_siblings = len(siblings)

# Modify the siblings tuple and add the name of your father and mother and assign it to family_members
father = "John"
mother = "Jane"
family_members = (father, mother) + siblings

# Print the results
print("Siblings:", siblings)
print("Number of Siblings:", num_siblings)
print("Family Members:", family_members)
```

```
Siblings: ('Jack', 'Ryan', 'Michael', 'Alice', 'Emily')
Number of Siblings: 5
Family Members: ('John', 'Jane', 'Jack', 'Ryan', 'Michael', 'Alice', 'Emily')
```

```python
# Given sets and list
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]
```

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.

```python
# Given sets and list
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
length_it_companies = len(it_companies)

# Add 'Twitter' to it_companies
it_companies.add('Twitter')

# Insert multiple IT companies at once to the set it_companies
it_companies.update(['LinkedIn', 'Netflix'])

# Remove one of the companies from the set it_companies
it_companies.remove('Netflix')

# What is the difference between remove and discard
# - remove(): Raises KeyError if element is not present.
# - discard(): Does not raise any error if element is not present.
# Example:
# it_companies.remove('Netflix') # Raises KeyError
# it_companies.discard('Netflix') # No error raised

# Join A and B
joined_set = A.union(B)

# Find A intersection B
intersection_set = A.intersection(B)
```

```python
# Is A subset of B
is_A_subset_of_B = A.issubset(B)

# Are A and B disjoint sets
are_disjoint = A.isdisjoint(B)

# Join A with B and B with A
joined_AB = A.union(B)
joined_BA = B.union(A)

# What is the symmetric difference between A and B
symmetric_difference = A.symmetric_difference(B)

# Delete the sets completely
del it_companies
del A
del B

# Convert the ages to a set and compare the length of the list and the set
ages_set = set(age)
length_age_list = len(age)
length_age_set = len(ages_set)

# Print the results
print("Length of it_companies:", length_it_companies)
#print("it_companies after adding 'Twitter' and multiple companies:", it_companies)
print("Joined set of A and B:", joined_set)
print("Intersection of A and B:", intersection_set)
print("Is A subset of B:", is_A_subset_of_B)
print("Are A and B disjoint sets:", are_disjoint)
print("Joined A with B:", joined_AB)
```

```
# Convert the ages to a set and compare the length of the list and the set
ages_set = set(age)
length_age_list = len(age)
length_age_set = len(ages_set)

# Print the results
print("Length of it_companies:", length_it_companies)
#print("it_companies after adding 'Twitter' and multiple companies:", it_companies)
print("Joined set of A and B:", joined_set)
print("Intersection of A and B:", intersection_set)
print("Is A subset of B:", is_A_subset_of_B)
print("Are A and B disjoint sets:", are_disjoint)
print("Joined A with B:", joined_AB)
print("Joined B with A:", joined_BA)
print("Symmetric difference between A and B:", symmetric_difference)
print("Length of age list:", length_age_list)
print("Length of age set:", length_age_set)
```

```
Length of it_companies: 7
Joined set 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
Are A and B disjoint sets: False
Joined A with B: {19, 20, 22, 24, 25, 26, 27, 28}
Joined B with A: {19, 20, 22, 24, 25, 26, 27, 28}
Symmetric difference between A and B: {27, 28}
Length of age list: 8
Length of age set: 5
```

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.

```
import math

# Given radius
radius = 30

# Calculate the area of a circle
area_of_circle = math.pi * radius ** 2

# Calculate the circumference of a circle
circum_of_circle = 2 * math.pi * radius

# Print the results
print("Area of the circle:", area_of_circle)
print("Circumference of the circle:", circum_of_circle)

# Take radius as user input and calculate the area
user_radius = float(input("Enter the radius of the circle: "))
user_area_of_circle = math.pi * user_radius ** 2
print("Area of the circle with radius", user_radius, ":", user_area_of_circle)
```

```
Area of the circle: 2827.4333882308138
Circumference of the circle: 188.49555921538757
Enter the radius of the circle: 1
Area of the circle with radius 1.0 : 3.141592653589793
```

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.

```
sentence = "I am a teacher and I love to inspire and teach people"

# Split the sentence into words
words = sentence.split()

# Convert the list of words into a set to get unique words
unique_words = set(words)

# Find the number of unique words
num_unique_words = len(unique_words)

# Print the result
print("Number of unique words:", num_unique_words)
```

```
Number of unique words: 10
```

Question 7: Use a tab escape sequence to get the following lines. Name Age Country City Asabeneh 250 Finland Helsinki.

```
print("Name\tAge\tCountry\tCity")
print("charitha\t21\tFinland\tHelsinki")
```

```
Name    Age     Country City
charitha        21      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.”

```
radius = 10
area = 3.14 * radius ** 2

result = f"The area of a circle with radius {radius} is {area} meters square."
print(result)

# Or if you want to display the area as an integer:
result_int = f"The area of a circle with radius {radius} is {int(area)} meters square."
print(result_int)
```

```
The area of a circle with radius 10 is 314.0 meters square.
The area of a circle with radius 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].

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

# Initialize an empty list to store the weights in pounds
weights_lbs = []

# Read the weights of N students into the list
for i in range(N):
    weight_lbs = float(input(f"Enter the weight of student {i+1} in pounds: "))
    weights_lbs.append(weight_lbs)

# Convert the weights from pounds to kilograms and store them in a separate list
weights_kg = []
for weight_lbs in weights_lbs:
    weight_kg = weight_lbs * 0.453592  # 1 pound is approximately 0.453592 kilograms
    weights_kg.append(round(weight_kg, 2))  # Round to two decimal places

# Print the converted weights in kilograms
print("Weights in kilograms:", weights_kg)
```

```
Enter the number of students: 5
Enter the weight of student 1 in pounds: 9
Enter the weight of student 2 in pounds: 9
Enter the weight of student 3 in pounds: 9
Enter the weight of student 4 in pounds: 9
Enter the weight of student 5 in pounds: 9
Weights in kilograms: [4.08, 4.08, 4.08, 4.08, 4.08]
```