

Machine Learning (Assignment #1)

Question 1

```
# Question 1
ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24]

# Sort the list. Find the min and max age
ages.sort()
min_age = min(ages)
max_age = max(ages)

print(ages, "\n", f"The min age is {min_age} and the max age is {max_age}\n") # format string

# Add the min age and max age again to the list
ages.insert(0, min_age)
ages.insert(-1, max_age)
print(ages)

# Find the median age
total = len(ages) # total number of items in ages list

if total % 2 == 0: # If number of items is even, select the two items in the middle and divide by 2
    median = (ages[(total//2) - 1] + ages[total//2]) // 2
    print(f"\nThe median is {median}")
else:
    median = (ages[total//2]) # If total number of items is odd, select the item in the middle of the list
    print(f"\nThe median is {median}")

# Find the average age
average = sum(ages) / total
print("The average age is", average)

# Find the range of the ages(max - min)
age_range = max_age - min_age
print("The age range is %d." % age_range)
```

[40] Python

```
... [19, 19, 20, 22, 24, 24, 24, 25, 25, 26]
    The min age is 19 and the max age is 26

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

The median is 24
The average age is 22.75
The age range is 7.
```

Question 2

```
# Question 2
# Create an empty dictionary
dog = dict()

# Add name, color, breed, legs, age to the dog dictionary
dog = {'name': 'Cookie', 'color': 'light brown', 'breed': 'pomeranian', 'legs': 4, 'age': '2 years old'}
print(dog.values())

# 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': 'Harry', 'last_name': 'Potter', 'gender': 'Male', 'age': 43, 'marital status': 'married',
           'skills': ['Magical Resistance', 'Parsel Tongue', 'Legilimency'],
           'address': '12 Grimmauld Place', 'city': 'London', 'country': 'England'}

# Get the length of the student dictionary, the value of skills and its data type
print("\nThe length of the student dictionary is", len(student),
      "\nSkills include: ", student.get('skills'),
      "\nThe data type of the values of skills is", type(student.get('skills'))))

# Modify the skills values by adding one or two skills
student['skills'] = student['skills'] + ['Occlumency', 'Transfiguration']

# To delete one element from the skills list
# student['skills'].remove('Parsel Tongue')

# Get the dictionary keys and values as lists
print("\nSkills: ", student.get('skills'), "\nKeys as list:", student.keys(),
      "\nValues as list:", student.values())
```

[52] Python

```
... dict_values(['Cookie', 'light brown', 'pomeranian', 4, '2 years old'])

The length of the student dictionary is 9
Skills include:  ['Magical Resistance', 'Parsel Tongue', 'Legilimency']
The data type of the values of skills is <class 'list'>

Skills:  ['Magical Resistance', 'Parsel Tongue', 'Legilimency', 'Occlumency', 'Transfiguration']
Keys as list: dict_keys(['first_name', 'last_name', 'gender', 'age', 'marital status', 'skills', 'address', 'city', 'country'])
Values as list: dict_values(['Harry', 'Potter', 'Male', 43, 'married', ['Magical Resistance', 'Parsel Tongue', 'Legilimency', 'Occlumency', 'Transfiguration'], '12 Grimmauld Place', 'London', 'England'])
```

Question 3

```
# Question 3
# Create a tuple containing names of your sisters and brothers
sisters = ('Claire', 'Hailey', 'Alex')
brothers = ('Phil', 'Luke', 'Mitchell', 'Cameron')

# Join brothers and sisters tuples and assign it to siblings
siblings = sisters + brothers
print("How many siblings due you have?", len(siblings))

# Modify the siblings tuple and add the name of your father and mother and assign it to family_members
family_members = siblings + ('Jay', 'Gloria')
print(family_members)
```

[53] Python

```
... How many siblings due you have? 7
('Claire', 'Hailey', 'Alex', 'Phil', 'Luke', 'Mitchell', 'Cameron', 'Jay', 'Gloria')
```

Question 4

```

# 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]

print("The length of the set it_companies is: ", len(it_companies))

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

# Insert multiple IT companies at once to the set it_companies
it_companies.update(['HP', 'Accenture', 'Infosys']) # can also use union method: it_companies | {'HP', 'Accenture', 'Infosys'}
print(it_companies)

# Remove one of the companies from the set it_companies
it_companies.remove('Facebook')
print("\nAfter removing one company:", it_companies, ":", len(it_companies))

# What is the difference between remove and discard
'''
The difference is in their behavior when called to remove an element that
doesn't exist in the set.
set.remove will throw a key error exception and
set.discard does not
'''

# The element exist in the set
it_companies.discard('HP')
it_companies.remove('IBM')
print("\nBehavior of remove and discard if element exists in set:", it_companies, ":", len(it_companies))

# The element do not exist in the set
it_companies.discard('Facebook')
#it_companies.remove('Facebook')

print("\nJoin A and B:", A.union(B),
      "\nA Intersection B:", A.intersection(B),
      "\nIs A subset of B?", A.issubset(B),
      "\nAre A and B disjoint sets?", A.isdisjoint(B),
      "\nJoin A with B:", A.union(B), "\nJoin B with A:", B.union(A),
      "\nWhat is the symmetric difference between A and B?", A ^ B
)

# Convert the ages to a set and compare the length of the list and the set
age_set = set(age)
print("The length of the list is equal to that of the set.", len(age) == len(age_set))

del(A, B) # Delete the sets completely
#print(B) # This throw an error because the set does not exist anymore
print(A)

```

[2] 0.1s Python

```

... The length of the set it_companies is: 7
{'Oracle', 'Google', 'Twitter', 'Amazon', 'Microsoft', 'IBM', 'Facebook', 'Apple'}
{'Oracle', 'Google', 'Amazon', 'Facebook', 'HP', 'Apple', 'Twitter', 'Microsoft', 'IBM', 'InfoSys', 'Accenture'}

After removing one company: {'Oracle', 'Google', 'Amazon', 'HP', 'Apple', 'Twitter', 'Microsoft', 'IBM', 'InfoSys', 'Accenture'} : 10

Behavior of remove and discard if element exists in set: {'Oracle', 'Google', 'Amazon', 'Apple', 'Twitter', 'Microsoft', 'InfoSys', 'Accenture'} : 8

Join A and B: {19, 20, 22, 24, 25, 26, 27, 28}
A Intersection B: {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}
What is the symmetric difference between A and B? {27, 28}
The length of the list is equal to that of the set. False

</>
-----
NameError                                Traceback (most recent call last)
c:\Users\Caroline\Documents\Spring 2023\CS5710 _ Machine Learning\Homework\Assignment_1.ipynb Cell 4 in <cell line: 51>()
    49 del(A, B) # Delete the sets completely
    50 #print(B) # This throw an error because the set does not exist anymore
--> 51 print(A)

NameError: name 'A' is not defined

```

Question 5

```

# Question 5
# The radius of a circle is 30 meters

PI = 3.14159
radius = 30

# Calculate the area of a circle
_area_of_circle_ = PI * radius ** 2
print("The area of a circle with radius %d meters is %.2f meters squared." %(radius, _area_of_circle_))

# Calculate the circumference of a circle
_circum_of_circle = 2 * PI * radius
print("The perimeter of a circle with radius %d meters is %.2f meters squared." %(radius, _circum_of_circle))

# Take the radius as user input and calculate the area

new_radius = eval(input("Enter the radius of a circle: "))
new_area_of_circle = PI * new_radius ** 2
print("The area of the user's circle with radius %.2f meters is %.2f meters squared." %(new_radius, new_area_of_circle))

```

[54] Python

```

... The area of a circle with radius 30 meters is 2827.43 meters squared.
The perimeter of a circle with radius 30 meters is 188.50 meters squared.
The area of the user's circle with radius 100.00 meters is 31415.90 meters squared.

```

Question 6

```
# Question 6
# How many unique words have been used in the sentence. Use split() and set
str1 = "I am a teacher and I love to inspire and teach people"
str1_list = str1.split()
print(str1_list, "\nNumber of words in the sentence:", len(str1_list), "\n") # Using print statement for debugging
str1_set = set(str1_list)
print(str1_set, "\nNumber of unique words in the sentence:", len(str1_set))
```

[95] Python

```
... ['I', 'am', 'a', 'teacher', 'and', 'I', 'love', 'to', 'inspire', 'and', 'teach', 'people']
Number of words in the sentence: 12

{'people', 'I', 'teach', 'inspire', 'to', 'teacher', 'a', 'love', 'and', 'am'}
Number of unique words in the sentence: 10
```

Question 7

```
# Question 7
# Use a tab escape sequence to get the following lines.
print("Name Age Country City : Name\tAge\tCountry\tCity")
print("Asabeneh 250 Finland Helsinki : Asabeneh\t250\tFinland \tHelsinki")
```

[56] Python

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

Question 8

```
# Question 8
# Use the string formatting method to display data
radius = 10
area = 3.14 * radius ** 2
print("The area of a circle with radius %d is %d meters square." % (radius, area))
```

[57] Python

```
... The area of a circle with radius 10 is 314 meters square.
```

Question 9

```
# Question 9
# Write a program that reads weights of N students into a list and convert these weights to Kgs
weights_in_lbs = [] # Empty list to store user input

lbs = eval(input("Enter weights in pounds:"))
while lbs != 0: # Value used to end the loop
    weights_in_lbs.append(lbs)
    lbs = eval(input("Enter weights in pounds:"))

print("L1: ", weights_in_lbs)

weights_in_kgs = [] # Empty list to store converted weights
for lb in weights_in_lbs:
    weights_in_kgs.append(round((lb / 2.2046), 2)) # Conversion 1kg = 2.20462262lb

print("Output: ", weights_in_kgs)
```

[58] Python

```
... L1: [150, 155, 145, 148]
Output: [68.04, 70.31, 65.77, 67.13]
```

Question 10

Machine Learning — Assignment 1

Qn 10

Dataset: 2 classes $\Rightarrow \bullet \rightarrow \text{Dog}$
 $\times \rightarrow \text{Cat}$

KNN classifier, $K=3$

(1)



Actual data	
Input	Labels
1	Dog
2	Dog
3	Cat
6	Cat
6	Cat
7	Dog
10	Dog
11	Dog

\Rightarrow Divide data equally in two parts.
 training data = [1, 3, 6, 10] *Randomly selected.*
 testing data = [2, 6, 7, 11]

\Rightarrow Find distance between test data and training data points
 $K=3$

	training data				Predicted output Majority
	Dog 1	Cat 3	Cat 6	Dog 10	
2	1	1	4	8	Cat
6	5	3	0	4	Cat
7	6	4	1	3	Cat
11	10	8	5	1	Cat

Predicted output

② Compute the confusion matrix.

TN (0)	FP (3)
FN (0)	TP (1)

Calculate:

$$i) \text{ Accuracy} = \frac{(TP+TN)}{P+N} = \frac{1+0}{4} = 25\%$$

$$ii) \text{ Sensitivity} = \frac{TP}{(TP+FN)} = \frac{TP}{P} = \frac{1}{1+0} = 1$$

$$iii) \text{ Specificity} = \frac{TN}{FP+TN} = \frac{0}{3} = 0$$