MACHINE LEARNING #1

Name: Gowtham Paluri

ID: 700741225

Github Link: https://github.com/gowtham8666/Machine-Learning-Assignments.git

Video link: https://drive.google.com/file/d/1VQvfDTiHWgzS6jQVfY_jfJwYJZjTk3Fo/view?usp=sharing

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)

⇨ By using the inbuilt functions from the library we have sorted the list , found the median , average and range of the list.

```
# Q1
import statistics
age_list = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24]
# Sorts age list in ascending order by default
age_list.sort()
print ("Sorted list:", age_list) # Displays sorted values
# Displays min value as we used min() method
print ("Min_age:", min(age_list))
# Displays max value as we used max() methods
print ("Max_age:", max(age_list))
# Adding  min_age and max_age values to the list
age_list.append(min(age_list))
age_list.append(max(age_list))
print ("After adding min_age and max_age values again:",age_list) #Displays the list again with new values
# Median (one middle item or two middle items divided by two, as we imported statistics library it
#calculates easily and provides the output)
median = statistics.median(age_list)
print ("Median:", median)
# Average age
average= sum(age_list)/len(age_list)
print ("Avg = ", average)
# Range
range=max(age_list)-min(age_list)
print ("Range = ", range)
```

```
Sorted list: [19, 19, 20, 22, 24, 24, 24, 25, 25, 26]
Min_age: 19
Max_age: 26
After adding min_age and max_age values again: [19, 19, 20, 22, 24, 24, 24, 25, 25, 26, 19, 26]
Median: 24.0
Avg =  22.75
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

```
# Q2
# Create an empty dictionary called dog
dog = dict()
# Adding data to dog dictionarys
dog['name'] = 'TONY'
dog['color'] = 'WHITE'
dog['breed'] = 'PUG'
dog['legs'] = 4
dog['age'] = 4
print("dog dictionary : ", dog)
# creating student dictionary with data
student = {
    "first_name": "GOWTHAM",
    "last_name": "PALURI",
    "gender": "MALE",
    "age": 23,
    "marital status": "single",
    "skills": ["python", "machine learning"],
    "country": "United States",
    "city": "lees summit",
    "address": "OVERLANDPARK"
}
print("student dictionary : ", student)
# length of the student dictionary
len_student = len(student)
# skills of the student from the dictionary
skills = student['skills']
# type of skills
print("type of skills :",type(skills))
# updating student skills
student['skills'].extend(["Java"])
print("updated student skills : ", student["skills"])
student['skills'].extend(["C#"])
print("updated student skills : ", student["skills"])
# keys of student dictionary
print("keys of student dictionary : ", list(student.keys()))
# values of student dictionary
print("values of student dictionary : ", list(student.values()))
```

```
dog dictionary :  {'name': 'TONY', 'color': 'WHITE', 'breed': 'PUG', 'legs': 4, 'age': 4}
student dictionary :  {'first_name': 'GOWTHAM', 'last_name': 'PALURI', 'gender': 'MALE', 'age': 23, 'marital status': 'sing
le', 'skills': ['python', 'machine learning'], 'country': 'United States', 'city': 'lees summit', 'address': 'OVERLANDPAR
K'}
type of skills : <class 'list'>
updated student skills :  ['python', 'machine learning', 'Java']
updated student skills :  ['python', 'machine learning', 'Java', 'C#']
keys of student dictionary :  ['first_name', 'last_name', 'gender', 'age', 'marital status', 'skills', 'country', 'city',
'address']
values of student dictionary :  ['GOWTHAM', 'PALURI', 'MALE', 23, 'single', ['python', 'machine learning', 'Java', 'C#'],
'United States', 'lees summit', 'OVERLANDPARK']
```

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

```
# Q3
# creating a tuple for brothers
brothers = ("KLAUS","ELIJAH")
print("brothers : ", brothers)
# creating a tuple for sisters
sisters =  ("FREYA","REBECCA")
print("sisters : ", sisters)

# creating a siblings tuple by adding brothers and sisters
siblings = brothers + sisters
print("siblings : ", siblings)

# length of siblings tuple
length_of_siblings = len(siblings)
print("length of siblings tuple", length_of_siblings)

# adding parents to siblings tuple  to create new tuple
family = siblings + ("MICHAEL","ESTHER")
print("family : ", family)
```

```
brothers :  ('KLAUS', 'ELIJAH')
sisters :  ('FREYA', 'REBECCA')
siblings :  ('KLAUS', 'ELIJAH', 'FREYA', 'REBECCA')
length of siblings tuple 4
family :  ('KLAUS', 'ELIJAH', 'FREYA', 'REBECCA', 'MICHAEL', 'ESTHER')
```

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.

```
# Q4
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]
# length of it_companies set
print("length of it_companies set : ", len(it_companies))
# adding twitter company to it_companies
it_companies.add("Twitter")
# removing company from it_companies
it_companies.discard("Oracle")
print("it companies : ", it_companies)
print(""" remove vs discard :
remove deletes the element from the list if not present it returns Key error,
discard deletes the element from the list otherwise return None""")
# joining A and B sets
print("Join of A and B : ", A.union(B))
# intersection of A and B sets
print("Intersection of A and B : ",A.intersection(B))
# checking if A is subset of B
print("Is A subset of B : ", A.issubset(B))
# check if A is disjoint of B
print("Is A disjoint of B : ", A.isdisjoint(B))
# A union B and B union A
print("A union B : ", A.union(B))
print("B union A : ", B.union(A))
# symmetric difference between two sets
print("set A difference with set B : ",A.difference(B))
# deleting sets A and B
A.clear()
B.clear()
# converting age list to set
set_ages = set(age)
# comparing length of list and length of set
print("Is length of age list same of length of age set : ", len(age) == len(set_ages))
```

```
length of it_companies set :  7
it companies :  {'Twitter', 'Apple', 'Microsoft', 'IBM', 'Facebook', 'Google', 'Amazon'}
 remove vs discard :
remove deletes the element from the list if not present it returns Key error,
discard deletes the element from the list otherwise return 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.

```
# Q5
# The radius of a circle is 30 meters.
# radius r
r = 30
# pi value constant
pi = 3.14
# calculating area of circle
_area_of_circle_ =  pi * r * r
print("Area of circle :", _area_of_circle_)
# calculating cirumference of circle
_circum_of_circle_ = 2 * pi * r
print("Circumference of circle :", _circum_of_circle_)

# input from user
r = float(input("Enter radius of circle : "))
# calculating area of circle from user inputs
area_of_circle = pi * r * r
print("Area of circle : ", area_of_circle)
```

```
Area of circle : 2826.0
Circumference of circle : 188.4
Enter radius of circle : 7
Area of circle :  153.86
```

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.

```
# Q6
sentence = """I am a teacher and I love to inspire and teach people"""
# splitting sentence into words
split_sentence = sentence.split()
print("words : ", split_sentence)
# getting unique words using set
unique_words = set(split_sentence)
print("unique words : ", unique_words)
```

```
words :  ['I', 'am', 'a', 'teacher', 'and', 'I', 'love', 'to', 'inspire', 'and', 'teach', 'people']
unique words :  {'teacher', 'to', 'inspire', 'people', 'a', 'I', 'teach', 'love', 'and', 'am'}
```

Question 7

Use a tab escape sequence to get the following lines.

Name	Age	Country	City
Asabeneh	250	Finland	Helsinki

```
# Q7
# By using a tab escape sequence we get the following lines.
a= "Name\t Age\tCountry\tCity\t\nAsabeneh 250\tFinland\tHelsinki"
print(a)
```

```
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.”

```
# Q8
radius = 10
area = 3.14 * radius ** 2
# print string using format
sent = "The area of a circle with radius {} is {} meters square.".format(radius, area)
print(sent)
```

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

main.py

Run

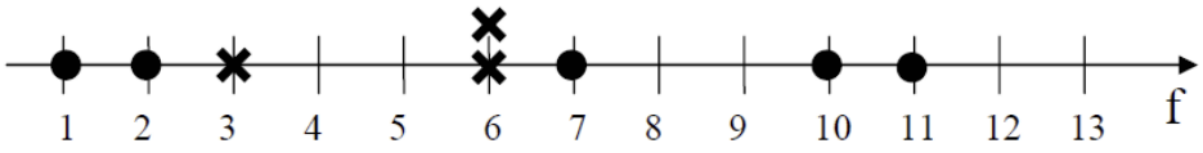
Shell

```
1
2  # input number of students
3  N = int(input("Enter Number of students : "))
4  # lb to kg conversion value
5  lbs_to_kg_convertion_value = 0.4536
6  lbs = []
7  kg = []
8  # input weights of all students
9  for i in range(0,N):
10     lbs.append(int(input("Enter student Weight(lbs) : ")))
11
12 print("weight in lbs : ",lbs)
13
14 # converte weights to kg
15 for weight in lbs:
16     kg.append(round(weight * lbs_to_kg_convertion_value,2))
17
18 print("kg_weights : ", kg)
```

```
Enter Number of students : 2
Enter student Weight(lbs) : 123
Enter student Weight(lbs) : 222
weight in lbs :  [123, 222]
kg_weights :  [55.79, 100.7]
>
```

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 solution, do not write code for it.



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

10).

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

Train set Test set.

1) By using the kNN classifier where k = 3,

$$d = \sqrt{(x - x_1)^2}$$

(x, x_1) $(6,6)$ $(6,3)$ $(6,2)$ $(6,1)$

these points need to be calculated.

i.e; $\Rightarrow d = \sqrt{(6-6)^2} = 0 \Rightarrow (6,6)$

$d = \sqrt{(6-3)^2} = 3 \Rightarrow (6,3)$

$d = \sqrt{(6-2)^2} = 4 \Rightarrow (6,4)$

$d = \sqrt{(6-1)^2} = 5 \Rightarrow (6,5)$

$\Rightarrow (0, 0, 1)$

max = 0 ∴ output is also 0.

for the rest of points is also predicted to be 0.

2). By using confusion matrix.

Accuracy = (TP+TN)/(TN+FP+FN+TP)

Sensitivity = TP/(TP+FN)

specificity = TN/(FP+TN)

	0	1
0	TN = 1	FP = 0
1	FN = 3	TP = 0

$$A = (0+1)/(1+0+3+0)$$

$$= 1/4 = 25\%$$

$$S = 0/0+3 = 0$$

$$S_p = 1/(0+1) = 1$$