**Machine Learning (Assignment # 1)**

**Solutions Document**

**Question-1:**

#creating a list as ages

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

#1.1 Sort the list and find the min and max age

#sorting the ages

ages.sort()

print (ages)

#max and min of the ages

Minimum = min(ages)

Maximum = max(ages)

print (Minimum)

print (Maximum)

#1.2 Add the min age and the max age again to the list

ages.append(Minimum)

ages.append(Maximum)

print (ages)

#1.3 Find the median age (one middle item or two middle items divided by two)

median = len(ages) // 2

result = (ages[median] + ages[~median]) / 2

print(result)

#1.4 Find the average age (sum of all items divided by their number)

average = sum(ages)/len(ages)

print (average)

#1.5 Find the range of the ages (max minus min)

range = Maximum - Minimum

print (range)

OUTPUT:

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

19

26

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

24.0

22.75

7

**Question-2:**

#2.1 Create an empty dictionary called dog

Dog = {}

#2.2 Add name, color, breed, legs, age to the dog dictionary

Dog = {'Name' : 'sarah','Color' : 'Golden', 'Breed': 'Golden Retriever','Legs' : 4,'Age' : 2}

print (Dog)

#creating empty dictionary as Student

Student= {}

#2.3 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' : 'Sai Sritha', 'Last\_name' : 'Mareedu', 'Gender' : 'Female', 'Age': 21, 'Marital status': 'No', 'Skills': ['Python'], ' Country': 'India', 'City': 'Hyderabad', 'Address': 'Pragathi nagar'}

print (Student)

#2.4 Get the length of the student dictionary

print(len(Student))

#2.5 Get the value of skills and check the data type, it should be a list

value\_skills = Student['Skills']

print(value\_skills, type(value\_skills))

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

Student['Skills'].append('HTML')

print(Student['Skills'])

#2.7 Get the dictionary keys as a list

keys = Student.keys()

print(list(keys))

#2.8 Get the dictionary values as a list

values = Student.values()

print(list(values))

OUTPUT:

{'Name': 'sarah', 'Color': 'Golden', 'Breed': 'Golden Retriever', 'Legs': 4, 'Age': 2}

{'First\_name': 'Sai Sritha', 'Last\_name': 'Mareedu', 'Gender': 'Female', 'Age': 21, 'Marital status': 'No', 'Skills': ['Python'], ' Country': 'India', 'City': 'Hyderabad', 'Address': 'Pragathi nagar'}

9

['Python'] <class 'list'>

['Python', 'HTML']

['First\_name', 'Last\_name', 'Gender', 'Age', 'Marital status', 'Skills', ' Country', 'City', 'Address']

['Sai Sritha', 'Mareedu', 'Female', 21, 'No', ['Python', 'HTML'], 'India', 'Hyderabad', 'Pragathi nagar']

**Question-3:**

#creating an empty tuples as Brothers and Sisters

Brothers = ()

Sisters = ()

#3.1 assigning values to the brothers and sisters tuples

Brothers = ('Srinivas', 'Dharmi', 'Veman', 'Gopi')

print(Brothers)

Sisters = ('Devi', 'Reshmitha', 'Prathyusha', 'Teena')

print(Sisters)

#3.2 joining brothers and sisters tuples as Siblings

Siblings = Brothers+Sisters

print(Siblings)

#3.3 printing NO.Of Siblings

print (len(Siblings))

#adding father and mother names to Siblings tuple

Siblings\_append = Siblings + ('Ram Babu', 'Jyothi')

#assigning Siblings tuple as family\_members

family\_members = Siblings\_append

print (family\_members)

output:

('Srinivas', 'Dharmi', 'Veman', 'Gopi')

('Devi', 'Reshmitha', 'Prathyusha', 'Teena')

('Srinivas', 'Dharmi', 'Veman', 'Gopi', 'Devi', 'Reshmitha', 'Prathyusha', 'Teena')

8

('Srinivas', 'Dharmi', 'Veman', 'Gopi', 'Devi', 'Reshmitha', 'Prathyusha', 'Teena', 'Ram Babu', 'Jyothi')

>

**Question-4:**

it\_companies = {'Facebook', 'Google', 'Microsoft', 'Apple', 'IBM', 'Oracle', 'Amazon'}

#4.1 Find the length of the set it\_companies

print(len(it\_companies))

#4.2 Add 'Twitter' to it\_companies

it\_companies.add('Twitter')

print(it\_companies)

#4.3 Insert multiple IT companies at once to the set it\_companies

it\_companies.update(['TCS', 'Capgemini', 'Maersk', 'Cognizant'])

print(it\_companies)

#4.4 Remove one of the companies from the set it\_companies

it\_companies.remove('Maersk')

print(it\_companies)

#4.5 What is the difference between remove and discard

#it\_companies.remove('Skylink')

#print(it\_companies)

#The above two lines causes error because there was no company named Skylink in the set

#so the remove method will raise an exception instead of printing the original set

it\_companies.discard('Skylink')

print(it\_companies)

#The above two lines print the original set values.

#The difference is discard method prints the original set though if it doesnt have the comapny Skylink in the set but remove method will raise an exceptional error.

#4.6 Join A and B

A = {19, 22, 24, 20, 25, 26}

B = {19, 22, 20, 25, 26, 24, 28, 27}

print(A.union(B))

#4.7 Find A intersection B

print(A.intersection(B))

#4.8 Is A subset of B

print(A.issubset(B)) #It should be true because A is subset of B

#4.9 Are A and B disjoint sets

print(A.isdisjoint(B)) #It should be false because A and B sets have common values so it is joint set

#4.10 Join A with B and B with A

print(A.union(B),B.union(A))

#4.11 What is the symmetric difference between A and B

print(A.symmetric\_difference(B))

#4.12 Delete the sets completely

A.clear()

print(A)

B.clear()

print(B)

#4.13 Convert the ages to a set and compare the length of the list and the set.

age = [22, 19, 24, 25, 26, 24, 25, 24]

age2 = set(age)

print(age2)

if len(age) == len(age2):

print("The length of list and set is equal")

else:

print("The length of list and set is not equal")

OUTPUT :

7

{'Oracle', 'Apple', 'Facebook', 'Microsoft', 'IBM', 'Google', 'Amazon', 'Twitter'}

{'TCS', 'Oracle', 'Capgemini', 'Apple', 'Facebook', 'Microsoft', 'Cognizant', 'IBM', 'Google', 'Amazon', 'Maersk', 'Twitter'}

{'TCS', 'Oracle', 'Capgemini', 'Apple', 'Facebook', 'Microsoft', 'Cognizant', 'IBM', 'Google', 'Amazon', 'Twitter'}

{'TCS', 'Oracle', 'Capgemini', 'Apple', 'Facebook', 'Microsoft', 'Cognizant', 'IBM', 'Google', 'Amazon', 'Twitter'}

{19, 20, 22, 24, 25, 26, 27, 28}

{19, 20, 22, 24, 25, 26}

True

False

{19, 20, 22, 24, 25, 26, 27, 28} {19, 20, 22, 24, 25, 26, 27, 28}

{27, 28}

set()

set()

{19, 22, 24, 25, 26}

The length of list and set is not equal

**Question-5:**

#To obtain pi value importing math package

import math as M

#Assigning radius as r= 30 as per the question

r = 30

#5.1 Calculate the area of a circle and assign the value to a variable name of area\_of\_circle

area\_of\_circle = M.pi\*r\*r

print(area\_of\_circle)

#5.2 Calculate the circumference of a circle and assign the value to a variable name of circum\_of\_circle

circum\_of\_circle = M.pi\*2\*r

print(circum\_of\_circle)

#5.3 Take radius as user input and calculate the area.

print(“Enter the radius: “)

rad = float(input())

area = M.pi\*rad\*rad

print(area)

OUTPUT:

2827.4333882308138

188.49555921538757

Enter the radius:

33

3421.194399759285

>

**Question-6:**

#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.

#Function to return all UniqueWords

def UniqueWords(A):

#It contains all the word count

count = {}

for word in A.split():

count[word] = count.get(word, 0) + 1

#It returns list of UniqueWords

return [word for word in count if count[word] == 1]

#Assigning the given statement to A.

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

print(UniqueWords(A))

OUTPUT :

['am', 'a', 'teacher', 'love', 'to', 'inspire', 'teach', 'people']

>

**Question-7:**

#7 Use a tab escape sequence to get the following lines.

#Name Age Country City

#Asabeneh 250 Finland Helsinki

print("Name\t\tAge\tCountry\tCity")

print("Asabeneh\t250\tFinland\tHelsinki")

OUTPUT:

Name Age Country City

Asabeneh 250 Finland Helsinki

**Question-8:**

#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

print('The area of a circle radius {radius} is {area} meters square.'.format(radius=10, area=3.14\*radius\*\*2))

OUTPUT :

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

**Question-9:**

#9 creating an empty list

lst = []

lst2 = []

#N number of students weights as input

n = int(input("N students weights: "))

#iterating till the range

for i in range(0,n):

weight = int(input())

lst.append(weight) #adding the weights

print(lst)

#converting lbs into kilograms in a seperate list using loop

for j in range(len(lst)):

value = lst[j]\*0.453592

lst2.append(round(value,2))

print(lst2)

**OUTPUT :**

N students weights: 5

34

88

23

94

12

[34, 88, 23, 94, 12]

[15.42, 39.92, 10.43, 42.64, 5.44]

**Question-10:**

Step 1:

Split the given data into train and test sets with 20% of data in the test set.

Make sure that, test set contains data points that represents both the classes uniformly.

Step 2:

Applying KNN algorithm on the training data

Considering K=3

For point at 1, out of three nearest neighbours, 2 are x and one is o, so output=x (incorrect)

For point at 2, out of three nearest neighbours, 2 are x and one is o, so output=x (incorrect)

For point at 3, out of three nearest neighbours, 2 are o and one is x, so output=o (incorrect)

For points at 6, out of three nearest neighbours, 2 are x and one is o, so output=x (correct)

For point at 7, out of three nearest neighbours, 2 are x and one is o, so output=x (incorrect)

For point at 10, out of three nearest neighbours, 1 is x and two are o, so output=o (correct)

For point at 11, out of three nearest neighbours, 2 are o and one is x, so output=o (correct)

Step 3:

Now test the accuracy of the model using test data.

Correct classifications=4

Incorrect classifications=4

Accuracy= 0.5

Sensitivity=2/(2+3)=2/5=0.4

Specificity= 3/3+0=3/3=1