

# CS5710- Machine Learning

## Assignment-1

Name – MohamadSuhail Polur

Student ID- 700745813

GitHub Repo Link - <https://github.com/Suhail700745813/Machine-Learning.git>

Video Demo Link - <https://youtu.be/BTSgKZBxung>

I have created the assignment as a jupyter notebook containing both questions and answers.

Question 10 is also there in the notebook along with my scanned mathematical calculation. File is also available separately as “mohamadsuhail.pdf” in the repository.

### 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)

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

Sort the list and find the min and max age

```
[14]: #Sorting the List using sorted function and assigning it to a variable.  
#List can also be sorted using <list>.sort() function and it will sort in place  
sorted_ages_list = sorted(ages)  
ages.sort()  
print("The Sorted List is: ", sorted_ages_list)  
  
#Using min() and max() function, we can get minimum and maximum value of list. Using placeholders, display the output  
min_age, max_age = min(sorted_ages_list), max(sorted_ages_list)  
print("The minimum age is {} and maximum age is {}".format(min_age, max_age))
```

The Sorted List is: [19, 19, 20, 22, 24, 24, 24, 25, 25, 26]

The minimum age is 19 and maximum age is 26

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

```
[15]: #Using the extend function we add the min and max age to the list  
ages.extend([min_age, max_age])  
print(ages)
```

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

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

```
[4]: #Find if the length of list is odd or even  
#If Length is odd, take center term as median; else, find average of middle 2 terms and serve as median  
mid_index = len(ages)//2  
if len(ages)%2==0:  
    #In this case mid_index = 6, total terms will be 12. Therefore ages[5] and ages[6] average will give correct median  
    median = (ages[mid_index] + ages[mid_index - 1]) / 2  
else:  
    median = ages[mid_index]  
  
print("Median is {}".format(median))
```

Median is 24.0

Create a student dictionary and add first\_name, last\_name, gender, age, marital status, skills, country, city and address as keys for the dictionary

```
[1]: student = {'first_name': 'Suhail', 'last_name': 'polur', 'gender': 'Male', 'age': 24, 'marital status': 'Unmarried', 'skills': ['Python', 'Angular', 'Git'], 'country': 'United States', 'city': 'Kansas City', 'address': 'Holmes Road, Kansas City, MO'}
```

Get the length of the student dictionary

```
[10]: #Using len function to get Length of a dictionary
print("Student dictionary length: ", len(student))
```

Student dictionary length: 9

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

```
[11]: #Using type() function to get Data type
print("Data type of skills in student dictionary is {} and values are {}".format(type(student['skills']), student['skills']))
```

Data type of skills in student dictionary is <class 'list'> and values are ['Python', 'Angular', 'Git']

Modify the skills values by adding one or two skills

```
[12]: #We will use extend function of list to add 2 skills to skills list in student dictionary
student['skills'].extend(['HTML','Javascript'])
print("Skills of student are: ", student['skills'])
```

Skills of student are: ['Python', 'Angular', 'Git', 'HTML', 'Javascript']

Get the dictionary keys as a list

```
[13]: #We can use casting and keys() function to directly output a List of keys
print("List of keys of student are: ", list(student.keys()))
```

List of keys of student are: ['first\_name', 'last\_name', 'gender', 'age', 'marital status', 'skills', 'country', 'city', 'address']

Get the dictionary values as a list

```
[14]: #Same as keys, we can use values() function to get values
print("List of values of student are: ", list(student.values()))
```

List of values of student are: ['Sushant', 'Ashish', 'Male', 27, 'Unmarried', ['Python', 'Angular', 'Git', 'HTML', 'Javascript'], 'United States', 'Kansas City', 'Holmes Road, Kansas City, MO']

Find the length of the set it\_companies

```
[10]: it_companies = {'Facebook', 'Google', 'Microsoft', 'Apple', 'Ust', '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("Length of it_companies: ",len(it_companies))
```

Length of it\_companies: 6

Add 'Twitter' to it\_companies

```
[7]: it_companies.add('Twitter')
print(it_companies)
```

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

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

```
[8]: #add() function adds only 1 entry, for multiple, we use update() and pass a List of items to be added
it_companies.update(['Cisco', 'Hexagon', 'Infosys'])
print(it_companies)
```

{'Microsoft', 'Cisco', 'Amazon', 'Apple', 'Google', 'Ust', 'Facebook', 'Oracle', 'Hexagon', 'Twitter', 'Infosys'}

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

```
[11]: #Using discard() function to remove a specific item given as parameter, if the first element is to be removed, then we can use pop()
it_companies.discard('Oracle')
print(it_companies)

['Apple', 'Microsoft', 'Google', 'Amazon', 'Ust', 'Facebook']
```

#### What is the difference between remove and discard

```
[12]: #discard() function performs a check if the intended item to remove exists previously or not. remove() should throw an error
#remove() - removing non-existent element
it_companies.remove('Oracle')

-----
KeyError                                     Traceback (most recent call last)
Cell In[12], line 4
      1 #discard() function performs a check if the intended item to remove exists previously or not. remove() should throw an
error
      2
      3 #remove() - removing non-existent element
----> 4 it_companies.remove('Oracle')

KeyError: 'Oracle'

[24]: # discard() - trying to discard same non-existent element
it_companies.discard('Oracle')
print(it_companies)
```

---

#### Delete the sets completely

```
[31]: #Using clear() function
A.clear()
B.clear()

print(A, B)

set() set()
```

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

```
[32]: #By casting we can convert a list to set
age_set = set(age)
print("List: ", age)
print("Set: ", age_set)

#Set removes duplicate items
print("Length of list is {} and that of set is {}".format(len(age), len(age_set)))

List: [22, 19, 24, 25, 26, 24, 25, 24]
Set: {19, 22, 24, 25, 26}
Length of list is 8 and that of set is 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.

Calculate the area of a circle and assign the value to a variable name of `_area_of_circle_`

```
[33]: #Assigning radius and value of pi, area= pi*r*r
#Assuming pi=3.14

pi = 3.14
radius = 30
_area_of_circle_ = pi*(radius**2)
print("Area of circle: {} m*m".format(_area_of_circle_))

#We can get more accurate value if we use Math module
import math
_area_of_circle_ = math.pi*(radius**2)
print("Area of circle(Math module): {} m*m".format(_area_of_circle_))

Area of circle: 2826.0 m*m
Area of circle(Math module): 2827.4333882308138 m*m
```

## 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."
```

```
[38]: #As in questions above, using same {} and format() function, we can represent multiple values
radius = 10
area = int(3.14*(radius**2))

print("The area of a circle with radius {} is {} meters square.".format(radius, area))

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]
```

```
[39]: number_of_students = int(input("Enter number of Students: "))
input_weights = []
output_weights = []
```

**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]
```

```
[39]: number_of_students = int(input("Enter number of Students: "))  
input_weights = []  
output_weights = []  
for i in range(number_of_students):  
    weight_pounds = float(input("Weight in pounds: "))  
    input_weights.append(weight_pounds)  
    #1lb = 0.45359237kg  
    weight_kg = round(weight_pounds * 0.45359237, 2)  
    output_weights.append(weight_kg)  
  
print("L1: ", input_weights)  
print("Output: ", output_weights)
```

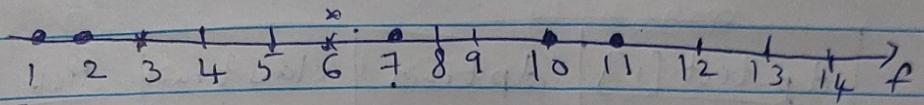
Enter number of Students: 5  
Weight in pounds: 150  
Weight in pounds: 155  
Weight in pounds: 145  
Weight in pounds: 148  
Weight in pounds: 200  
L1: [150.0, 155.0, 145.0, 148.0, 200.0]  
Output: [68.04, 70.31, 65.77, 67.13, 90.72]

## Question 10

CS 5710 - Assignment 1 (2002)

Polar Mohamad Suhail

Q. No. 10 :-



Assume class A = [1, 2, 7, 10, 11]

class B = [3, 6, 5]

splitting data points to equal sets

dp-train = [1, 2, 3, 6]

class-train = [A, A, B, B]

dp-test = [6, 7, 10, 11]

class-test = [B, A, A, A]

KNN where k=3      Euc. Dist =  $\sqrt{(x-x_i)^2}$

as there is only 1 feature

dp-test [0] = 6 => d-test [0] / dp-train dist class-train

$$\text{eg. } \sqrt{(6-1)^2}$$

$$\approx \sqrt{25} = 5$$

dp-train	dist	class-train
1	5	A
2	4	(A) $\rightarrow k=3$
3	3	(B) $\rightarrow k=2$
6	0	(B) $\rightarrow k=1$

Majority  $\Rightarrow B$  \* Prediction = B

dp-test [1] = 7

dp-test [1]	dp-train	class-train	dist	Majority=B pred=B
7	1	A	6	
7	2	A	5 = 7   k=3	
7	3	B	4 => k=2	
7	6	B	1 = 7   k=1	

dp-test [2] = 10

dp-test [2]	dp-train	class-train	dist	Majority=B pred=B
10	1	A	9	
10	2	(A)	8 => k=3	
10	3	(B)	7 => k=2	
10	6	(B)	4 => k=1	

dp-test [3] = 11

dp-test [3]	dp-train	class-train	dist	Majority=B pred=B
11	1	A	10 =	
11	2	(A)	9 => k=3	
11	3	(B)	8 => k=2	
11	6	(B)	5 => k=1	

$$dp\_test = [6, 7, 10, 11]$$

$$class\_test = [B, A, A, A] \Rightarrow \text{Actual classification}$$

$$pred\_test = [B, B, B, B] \Rightarrow \text{predicted output}$$

Part 2

We should Assume A = Positive

B = Negative

confusion Matrix

		Actual	
		positive	Negative
Pred	P	TP	FP
	N	FN	TN

In this case =)

	P	N	
P	0	3	
N	0	1	

$\Rightarrow$  confusion Matrix

$$\text{Now, Accuracy} = \frac{TN + TP}{TN + TP + FN + FP} = \frac{1}{11} = 0.25$$

$$\boxed{\text{Accuracy} = 25\%}$$

$$dp\_test = [6, 7, 10, 11]$$

$$class\_test = [B, A, A, A] \Rightarrow \text{Actual classification}$$

$$pred\_test = [B, B, B, B] \Rightarrow \text{predicted output}$$

Part 2

We should Assume A = Positive

B = Negative

confusion Matrix

		Actual	
		positive	Negative
Pred	P	TP	FP
	N	FN	TN

In this case =)

	P	N
P	0	3
N	0	1

$\Rightarrow$  confusion matrix

$$\text{Now, Accuracy} = \frac{TN + TP}{TN + TP + FN + FP} = \frac{1}{11} = 0.25$$

$$\boxed{\text{Accuracy} = 25\%}$$

Sensitivity =  $\frac{TP}{TP + FN}$  =  $\frac{0}{0} \Rightarrow$  Worst case  
Sensitivity is represented by 0.

[Sensitivity = 0%]

Specificity =  $\frac{TN}{TN + FP} = \frac{1}{4} = 0.25$ .

[Specificity = 25%]