

Machine Learning

Assignment -1

Student: Sowjanya Sunkavalli
ID: 700731896

Question1

* Sort the list and find the min and max age

```
1 ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24] # Given List
2 ages.sort() # sort() is used to sort the given List
3 print('Sorted List is :', ages)
4 mn = min(ages) # min() will return min value from the list
5 print('Minimum age is : ', mn)
6 mx = max(ages) # max() will return max value from the list
7 print('Maximum age is : ', mx)
```

```
Sorted List is : [19, 19, 20, 22, 24, 24, 24, 25, 25, 26]
Minimum age is : 19
Maximum age is : 26
```

- To sort the given list, we use **list.sort()** function.
- To find the max and min of the given list we must use **max(list)** and **min(list)** respectively.

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

```
1 ages.extend([mn,mx]) # extend() is used to add the elements to the list
2 print('Extended list of ages is : ', ages)
```

```
Extended list of ages is : [19, 19, 20, 22, 24, 24, 24, 25, 25, 26, 19, 26]
```

- To add the min and max values again to the existing list we use the **list.extend(values)** function.

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

```
1 ages.sort() # sort the extended list again using sort()
2 if len(ages) % 2 ==1:
3     median = ages[len(ages)//2]
4 else:
5     median =(ages[len(ages)//2-1]+ages[len(ages)//2])/2
6 print('Median age of given list is : ', median)
```

```
Median age of given list is : 24.0
```

- To get the median of the given list, we must get the middle value of the list. But for that we need to check whether list consists of even or odd number of elements.
- If list contains even no. of elements, then we must get median from middle two numbers divided by two or else if the list contains odd number of elements, then the middle element divided by 2 is median.
- Figure out this logic by taking the length of the list from **len(list)** function and checking the logic of divisible by 2 or not.

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

```
1 # Find the average age (sum of all items divided by their number)
2 #avg = mean(ages) # mean() is used to get the average of the given list
3 avg = sum(ages)/len(ages)
4 print('Average age of given list is : ', avg)
```

Average age of given list is : 22.75

- Average is calculated by sum of all elements divided by total number of elements, so here we take sum from **sum(list)** and no. of elements from **len(list)** function and dividing sum with length gives us the average value of the list.

*** Find the range of the ages (max minus min)**

```
1 # Find the range of the ages (max minus min)
2 r = max(ages)-min(ages) # max - min gives the range of ages here
3 print('Range of given age list is : ',r)
```

Range of given age list is : 7

Range of the given list is max value subtracted from min value, so here we can get this value by taking difference between **max(list)** and **min(list)** values.

Question2

*** Create an empty dictionary called dog**

```
1 dog = dict() # dict() is used to create the new dictionary
```

- To create the dictionary, we must use **dict()** keyword and assign it to the variable name.

*** Add name, color, breed, legs, age to the dog dictionary**

```
1 dog = {'name': 'Tommy', 'color': 'brown', 'breed': 'pug', 'legs': 4, 'age': '6 months'} # adding key value pairs
2 dog
```

```
In [9]: {'name': 'Tommy',
        'color': 'brown',
        'breed': 'pug',
        'legs': 4,
        'age': '6 months'}
```

- To add key value pairs in dictionary we must use below format
- **Dict = {'key1': 'value1', 'key2': 'value2' ...}**. In the above code we have added required keys and corresponding values i.e., name, color, breed, legs and age keys and values as shown.

* 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 = dict()
2 student = {'first_name': 'Sowjanya', 'last_name': 'Sunkavalli', 'gender': 'Female', 'age': '26', 'marital status': 'Single'
3 , 'skills': ['Java', 'Python', 'AWS', 'Selenium'], 'country': 'USA', 'city': 'Warrensburg', 'address': 'S Main St, Warrensburg'}
4 student

10]: {'first_name': 'Sowjanya',
      'last_name': 'Sunkavalli',
      'gender': 'Female',
      'age': '26',
      'marital status': 'Single',
      'skills': ['Java', 'Python', 'AWS', 'Selenium'],
      'country': 'USA',
      'city': 'Warrensburg',
      'address': 'S Main St, Warrensburg'}
```

- Here we have created student dictionary and added required key value pairs and added **list of values to the Skill key**.

* Get the length of the student dictionary

```
1 # len(dictionaryname) gives us the total length of the dictionary
2 print('Length of student dictionary is : ', len(student))

Length of student dictionary is : 9
```

- Length of the dictionary can be found by using **len(dictionary)** function.

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

```
1 #Get the value of skills and check the data type, it should be a list
2 print('Values of Skills key are : ', student['skills'])
3 print('Data Type of Skills values is : ', type(student['skills']).__name__)

Values of Skills key are : ['Java', 'Python', 'AWS', 'Selenium', 'Jenkins']
Data Type of Skills values is : list
```

- To get the values of the corresponding key, **dictionary[key]** is used.
- To get the datatype of the returned values, I have used **type(dict[key]).__name__** function.

* Modify the skills values by adding one or two skills

```
1 student['skills'].append('Jenkins')
2 student

12]: {'first_name': 'Sowjanya',
      'last_name': 'Sunkavalli',
      'gender': 'Female',
      'age': '26',
      'marital status': 'Single',
      'skills': ['Java', 'Python', 'AWS', 'Selenium', 'Jenkins'],
      'country': 'USA',
      'city': 'Warrensburg',
      'address': 'S Main St, Warrensburg'}
```

- To modify the skills value, i.e., add new value here I have used **dictionary[key]. append (new values)** function.

* Get the dictionary keys as a list

```
1 #Get the dictionary keys as a list
2 student.keys()

4]: dict_keys(['first_name', 'last_name', 'gender', 'age', 'marital status', 'skills', 'country', 'city', 'address'])
```

- To get all the keys as a set I have **dictionary.keys()** function.

* Get the dictionary values as a list

```
1 # Get the dictionary values as a list
2 student.values()

52]: dict_values(['Sowjanya', 'Sunkavalli', 'Female', '26', 'Single', ['Java', 'Python', 'AWS', 'Selenium', 'Test', 'Jenkins'],
'USA', 'Warrensburg', 'S Main St, Warrensburg'])
```

To get all the keys as a set I have **dictionary.keys()** function.

Question3

* Create a tuple containing names of your sisters and your brothers (imaginary siblings are fine)

```
1 # Create a tuple containing names of your sisters and your brothers (imaginary siblings are fine)
2 brothers = ('Mahi','Jagu','Karthi','Abhi') # brothers tuple
3 print('Brothers tuple is : ', brothers)

Brothers tuple is :  ('Mahi', 'Jagu', 'Karthi', 'Abhi')

1 sisters = ('Kaveri','Bharu','Lakshmi','Raji') # sisters tuple
2 print('Sisters tuple is : ', sisters)

Sisters tuple is :  ('Kaveri', 'Bharu', 'Lakshmi', 'Raji')
```

- Here, I have created Brothers and sisters tuples.
- Tuples are created as **tuple = (tuple values)**

* Join brothers and sisters tuples and assign it to siblings

```
1 # Join brothers and sisters tuples and assign it to siblings
2 siblings = brothers + sisters
3 print('Siblings tuple is : ', siblings)

Siblings tuple is :  ('Mahi', 'Jagu', 'Karthi', 'Abhi', 'Kaveri', 'Bharu', 'Lakshmi', 'Raji')
```

- Here I have joined two tuples and assigned the result tuple to Siblings variable.
- Tuples can be added by using **result tuple = tuple1+tuple2**

* How many siblings do you have?

```
1 # How many siblings do you have?
2 print('Length of siblings tuple is : ',len(siblings))

Length of siblings tuple is :  8
```

- Here to find the total number of siblings, we need to find out total length of the tuple.
- Length of the tuple is found by using **len(tuple)**

*** Modify the siblings tuple and add the name of your father and mother and assign it to family_members**

```
1 # Modify the siblings tuple and add the name of your father and mother and assign it to family_members
2 family_members = siblings + ('Lakshmi','Anjaneyelu')
3 print('Family members tuple is : ',family_members )
```

Family members tuple is : ('Mahi', 'Jagu', 'Karthi', 'Abhi', 'Kaveri', 'Bharu', 'Lakshmi', 'Raji', 'Lakhmi', 'Anjaneyelu')

Here I have modified the tuple values by adding the father and mother values to the existing tuple and assigning that new tuple to the variable name Family Members.

To modify / add the new tuple members here I have used the + operator and added another tuple to the existing tuple.

New tuple = existing tuple + (tuple of values)

Question4

*** Find the length of the set it_companies**

```
1 it_companies = {'Facebook', 'Google', 'Microsoft', 'Apple', 'IBM', 'Oracle', 'Amazon'}
2 A = {19, 22, 24, 20, 25, 26}
3 B = {19, 22, 20, 25, 26, 24, 28, 27}
4 age = [22, 19, 24, 25, 26, 24, 25, 24]
```

```
1 it_companies = {'Facebook', 'Google', 'Microsoft', 'Apple', 'IBM', 'Oracle', 'Amazon'}
2 it_companies
```

47]: {'Amazon', 'Apple', 'Facebook', 'Google', 'IBM', 'Microsoft', 'Oracle'}

```
1 # Find the Length of the set it_companies
2 print('Length of set it_companies is : ',len(it_companies))
```

Length of set it_companies is : 7

- To find the length of the set, I have used len(set) function.

*** Add 'Twitter' to it_companies**

```
1 # Add 'Twitter' to it_companies
2 it_companies.add('Twitter')
3 print('New Set after adding Twitter is : ',it_companies )
```

New Set after adding Twitter is : {'Google', 'Amazon', 'Apple', 'IBM', 'Microsoft', 'Twitter', 'Facebook', 'Oracle'}

- To add new value to the existing set I have used **set.add('new value')** function.
- Here I have added Twitter to the existing set.

*** Insert multiple IT companies at once to the set it_companies**

```
1 # Insert multiple IT companies at once to the set it_companies
2 new_companies = ['Tesla','Intel','VMWare']
3 it_companies.update(new_companies)
4 print('New Set after adding : ',it_companies )
```

New Set after adding : {'IBM', 'Apple', 'Tesla', 'Google', 'Oracle', 'VMWare', 'Amazon', 'Intel', 'Facebook', 'Microsoft', 'Twitter'}

- Here I have used **update()** function to insert the new IT companies to the existing set.
- Firstly, I have taken new set with new values and passed that new set in the update function to get the companies inserted to the existing list.

* Remove one of the companies from the set it_companies

```
1 it_companies.remove('Facebook')
2 it_companies
```

```
i1]: {'Amazon',
      'Apple',
      'Google',
      'IBM',
      'Intel',
      'Microsoft',
      'Oracle',
      'Tesla',
      'Twitter',
      'VMware'}
```

- Here I have used the **remove(value)** function to remove the specific value from the set.
- Here I have removed the value Facebook from the Set.

* What is the difference between remove and discard

```
1 ##The discard() method and remove() method removes the specified item from the set.
2 ## This discard() method is different from the remove() method, because the remove() method
3 ## will raise an error if the specified item does not exist in set, and the discard() method will not raise
4 ## an error if the element is not present in set.
```

- **discard()** and **remove()** both are used to remove the specific item from the set, but the only difference is that **discard()** won't throw any error if the specific item is not present in the set where as **remove()** throws an error if the specified item is not there in the given set.

* Join A and B

```
1 print('A union B is : ', A.union(B))
```

A union B is : {19, 20, 22, 24, 25, 26, 27, 28}

* Find A intersection B

```
1 print('A intersection B is :',A.intersection(B))
```

A intersection B is : {19, 20, 22, 24, 25, 26}

* Is A subset of B

```
1 # issibset() is used to identify if two sets one set is subset of other set
2 print('Is A subset of B :',A.issubset(B))
```

Is A subset of B : True

* Are A and B disjoint sets

```
1 # isdisjoint() is used to know if two sets are disjoint or not - it returns boolean value
2 print('Is A disjoint of B :',A.isdisjoint(B))
```

Is A disjoint of B : False

- Here I have used **set1.union(set2)**, **set1.intersection(set2)**, **set1.issubset(set2)**, **set1.isdisjoint(set2)** to get the union, intersection, issubset, isdisjoint functions for the given sets to get union, intersection values and also to verify whether both sets have any relationship based on subset and disjoint sets.
- Union and Intersection gives the output set of union values and intersection values of Set A and B, whereas **issubset** and **isdisjoint** functions gives the corresponding **Boolean** values.

* Join A with B and B with A

```
: In 1: print('Joining A with B :', A.union(B))
Joining A with B : {19, 20, 22, 24, 25, 26, 27, 28}

: In 1: print('Joining B with A :', B.union(A))
Joining B with A : {19, 20, 22, 24, 25, 26, 27, 28}
```

* What is the symmetric difference between A and B

```
: In 1: print(' Symmetric difference between A and B is :',A.symmetric_difference(B))
Symmetric difference between A and B is : {27, 28}
```

* Delete the sets completely

```
: In 1: del(A) # del(set) - permanently delete the set
2: del(B)
```

- To join two sets, I have used union function as **set1.union(set2)**
- To get the symmetric difference of two sets I have used **set1.symmetricdifference(set2)**
- To delete the two sets permanently I have used **del(set)** function, then this set will be deleted permanently.

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

```
In 1: print('Converting Ages to a Set :', set(age))
Converting Ages to a Set : {19, 22, 24, 25, 26}
```

```
In 1: print('Length of age list :', len(age))
Length of age list : 8
```

```
In 1: print('Lenght of Age Set :', len(set(age)))
Lenght of Age Set : 5
```

```
In 1: # len(age) > len(set(age))
```

To convert ages list into set, I have used **set(list)** function.

To get the length of the age set I have used **len(age)** function.

Here, when we compare length of ages list and ages set, **length of list > length of set.**

Question5

* Calculate the area of a circle and assign the value to a variable name of *area_of_circle*

```
1 # The radius of a circle is 30 meters.
2 # Calculate the area of a circle and assign the value to a variable name of _area_of_circle_
3 r = 30
4 area = 3.14 * r * r
5 _area_of_the_circle_ = area # Import Math Library and get the pi value
6 print (" The area of the given circle in meters is: ", _area_of_the_circle_)
```

The area of the given circle in meters is: 2826.0

* Calculate the circumference of a circle and assign the value to a variable name of *circum_of_circle*

```
1 # Calculate the circumference of a circle and assign the value to a variable name of _circum_of_circle_
2 circumference = 2*3.14*r
3 _circum_of_circle_ = circumference
4 print (" The Circumference of the given circle in meters is: ", _circum_of_circle_)
```

The Circumference of the given circle in meters is: 188.4

* Take radius as user input and calculate the area.

```
1 # Take radius as user input and calculate the area.
2 r = input("Enter radius of circle in Meters:")
3 area = (3.14) * float(r) * float(r)
4 print ("The area of the given circle in meters is: ",area)
```

Enter radius of circle in Meters:30

The area of the given circle in meters is: 2826.0

Here, I have calculated area from given radius as $\text{area} = 3.14 * r * r$ and assigned that value to the `_area_of_the_circle_`

Similarly, I have calculated circumference from given radius as $\text{circumference} = 2 * 3.14 * r$ and assigned that value to `_circum_of_circle_`

For the radius input, I have used `input()` function to get the radius input from user and calculated the area from the formula $\text{area} = 3.14 * r * r$

Question6

* How many unique words have been used in the sentence? Use the split methods and set to get the unique words.

```
] : 1 x = 'I am a teacher and I love to inspire and teach people'
    2 x
```

t[40]: 'I am a teacher and I love to inspire and teach people'

```
] : 1 words = x.split()
    2 words
```

```
t[41]: ['I',
        'am',
        'a',
        'teacher',
        'and',
        'I',
        'love',
        'to',
        'inspire',
        'and',
        'teach',
        'people']
```

```
] : 1 # Find the unique words --- this wrong -- should try properly with set
    2 unique_words = set(words)
    3 unique_words
```

t[42]: {'I', 'a', 'am', 'and', 'inspire', 'love', 'people', 'teach', 'teacher', 'to'}

```
] : 1 len(unique_words)
```

t[43]: 10

To get the unique words, first split the string into individual words it will give a list. I have used **split ()** function for the same.

Then we need to convert that list into set to have only unique words. I have used **set(list)** for this.

Then finally, we need to take length of the set to the no. of unique words in the given string. Used **len(set)** for getting length.

Question7

* Use a tab escape sequence to get the following lines.

```
1 # Name Age Country City
2 # Asabeneh 250 Finland Helsinki
```

```
1 #\t is used for tab and \n is used for next line
2 print('Name\tAge\tCountry\tCity\nAsabeneh\t250\tFinland\tHelsinki')
```

Name	Age	Country	City
Asabeneh	250	Finland	Helsinki

I have used **\t** and **\n** escape characters to get the tab space and next line for printing the required output.

Question8

* Use the string formatting method to display the following:

```
1 radius = 10
2 area = 3.14 * radius ** 2
3 text = "The area of a circle with radius is {} and area is {} meters square.".format(radius,area)
4 text
```

```
16]: 'The area of a circle with radius is 10 and area is 314.0 meters square.'
```

- To pass the strings as format values, I have used **.format(string)** function here, to pass area and radius.
- The **format()** method formats the specified value(s) and insert them inside the string's placeholder.

Question9

* 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)

```
1 N = int(input('Enter No. of students : '))
2 weights_kgs = []
3 weights_lbs = []
4
5 for i in range(N):
6     weights_lbs.append(int(input('Enter student{} weight in lbs: '.format(i+1))))
7     print('Weights in Lbs is :', weights_lbs)
8
9     for i in weights_lbs:
10        x = i*0.453592 #1 Lb = 0.453592 Kg
11        weights_kgs.append(float("%.3f" %x))
12    print('Weights in Kgs is :', weights_kgs)

Enter No. of students : 4
Enter student1 weight in lbs: 2
Enter student2 weight in lbs: 12
Enter student3 weight in lbs: 21
Enter student4 weight in lbs: 5
Weights in Lbs is : [2, 12, 21, 5]
Weights in Kgs is : [0.907, 5.443, 9.525, 2.268]
```

Here I have used **input()** method to take the user input for no.of students and after that declared new sets for both weights in kgs and weights in lbs.

Then declared for loop for range of N no.of students and used **append()** function to add the user input into a list of lbs.

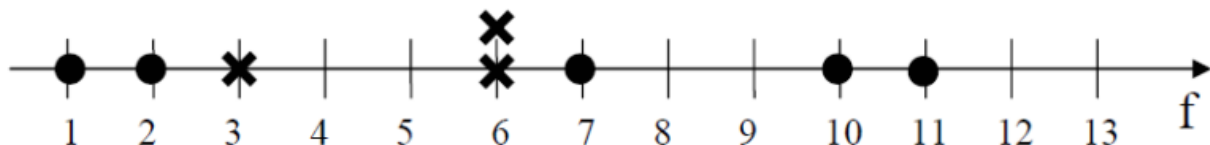
Here I have used format method to specific the student ID from the user input.

Finally, created another for **loop** and pass our weights_lbs list into this function to **iterate** through all the elements and passing the lbs to kg logic for each element and appending that element into the new weights_kgs list.

For this I have used **.append()** function and used %3f to get the decimal value till 3 digits.

Question10

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.



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.

First, convert the given dataset into a table as shown below and divide the first set into Training Set and second set into Testing Set.

1.Ans

From given set, consider dot as O and Crossmark as X

Feature Label

1	O	}	Set1 – Training Set
2	O		
3	X		
6	X		
6	X	}	Set2 – Testing Set
7	O		
10	O		
11	O		

And then find out the distance from each testing data point to the Training Data Point as shown below.

From the below tabular form we need to consider the nearest 3 training points based on the KNN classification on K=3

Distance between Data Points

	1	2	3	6
6	$6-1=5$	$6-2=4$	$6-3=3$	$6-6=0$
7	$7-1=6$	$7-2=5$	$7-3=4$	$7-6=1$
10	$10-1=9$	$10-2=8$	$10-3=7$	$10-6=4$
11	$11-1=10$	$11-2=9$	$11-3=8$	$11-6=5$

So, based on the $K = 3$ consideration, nearest data training points for 6, 7, 10 and 11 are 6, 3 and 2.

So, 2 represents O and 6,3 represents X, means majority is X. So, the predicted values for the Testing Data points will be as shown below

Testing Data	Class Type	Predicted Class Type
6	X	X
7	O	X
10	O	X
11	O	X

2.Ans

We need to compute confusion matrix for this and calculate accuracy, sensitivity, and specificity values.

Confusion Matrix:

TN	FP
FN	TP

Here, TN – True Negative, FP – False Positive, FN – False Negative, TP – True Positive.

Consider Class X as Positive and O as Negative.

Then based on the Prediction Values we can construct the Confusion Matrix as shown below for our Testing Data Points

Testing Data	Class Type	Predicted Class Type	TP/FP/TF/FN
6	X	X	TP
7	O	X	FP
10	O	X	FP
11	O	X	FP

So, from the constructed table, we got the following values with respective confusion matrix parameters,

TP- 1, FP-3, TN – 0, FN – 0

Then, confusion matrix will be as follows,

TN - 0	FP - 3
FN - 0	TP - 1

We know that,

Accuracy

$$= (TP+TN)/(P+N)$$

Sensitivity or true positive rate (TPR)

$$= TP/(TP+FN) = TP/P$$

Specificity or TNR

$$= TN/(FP+TN) = TN/N$$

Then the final calculated Values will be as follows,

$$P = TP+FN; N = FP+TN$$

$$\text{Accuracy} = (1+0) / (1+3) = 1/4 = 0.25$$

$$\text{Sensitivity} = (1) / (1+0) = 1$$

$$\text{Specificity} = (0) / (3+0) = 0$$

Video Link:

<https://youtu.be/DzAY9JdHdkM>

Git Hub Link:

<https://github.com/sunkavallisowjanya/Machine-Learning>