

Assignment 1

Name: Rohitha Sai Alla

No: 700734780

Video link: <https://drive.google.com/file/d/1ydzs5YRYD2poR2qDCSg545tFdgQXzO8E/view?usp=sharing>

1.

```
ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24]
#sorting the list
ages.sort()
#finding the min and max age
min = ages[0]
max = ages[-1]
print("Minimum age: ", min, "Maximum age: ", max)

#adding min and max age to list
ages.append(min)
ages.append(max)
print(ages)

#finding the median age
if len(ages) % 2 != 0:
    med = int((len(ages)+1)/2-1)
    median = ages[med]
    print("Median: ", median)
else:
    med1 = int(len(ages)/2 - 1)
    med2 = int(len(ages)/2)
    median = (ages[med1]+ages[med2])/2
    print("Median: ", ages[med1], ages[med2])

#find average age
avg = sum(ages) / len(ages)
print("Average: ", avg)

#finding range of ages
age_range = max - min
print("Range of ages: ", age_range)
```

After sorting add the first and last element to the list and checked if the length of list is even or odd.

If list length is even then I am returning two values else one.

Output:

```
main x
C:\Users\Rohitha\PycharmProjects\ML\venv\Scripts\python.exe C:/Users/Rohitha/PycharmProjects/ML/main.py
Minimum age: 19 Maximum age: 26
[19, 19, 20, 22, 24, 24, 24, 25, 25, 26, 19, 26]
Median: 24 24
Average: 22.75
Range of ages: 7

Process finished with exit code 0
```

2.

Created new dictionary and updated values as asked.

Performed update operations and added skills as list inside dictionary.

Also updated skills with new extra values. Shown the keys and value pairs.

```
dog = {} # creating empty dictionary
dog.update({"name": "Husky", "color": "White", "breed": "pug", "legs": "4",
"age": "3"}) # adding values
student_dict = {
    "first_name": "Rohitha",
    "last_name": "Sai",
    "gender": "Female",
    "age": 24, "maritalStatus": "Unmarried", "skills": ["Python"], "Country":
"India", "City": "Hyderabad",
    "Address": "Overland park"
}
print("Length of student dictionary: ", len(student_dict))
print("Value of skills and datatype: ", student_dict["skills"])

# updating skills in dictionary
student_dict.update({"skills": ["Python", "Numpy"]})
print("Modified skills: ", student_dict)

# Getting dictionary keys as a list
print(list(student_dict.keys()))

# Getting dictionary values as a list
print(list(student_dict.values()))
```

Output:

```

main x
C:\Users\Rohitha\PycharmProjects\ML\venv\Scripts\python.exe C:/Users/Rohitha/PycharmProjects/ML/main.py
Length of dictionary: 9
Value of skills and datatype: ['Python']
{'first_name': 'Rohitha', 'last_name': 'Sai', 'gender': 'Female', 'age': 24, 'maritalStatus': 'Unmarried', 'skills': ['Python', 'Numpy'], 'Country': 'India', 'City': 'Hyderabad',
['first_name', 'last_name', 'gender', 'age', 'maritalStatus', 'skills', 'Country', 'City', 'Address']
['Rohitha', 'Sai', 'Female', 24, 'Unmarried', ['Python', 'Numpy'], 'India', 'Hyderabad', 'Overland park']

```

3. Created two tuples sisters and brothers and merged these two into another tuple named siblings.

Printed the count of the siblings tuple.

Created new tuple named family_members and added father and mother to family_members along with siblings.

```

# Creating tuple with sisters
sisters = ("sister1", "sister2")

# Creating tuple with brothers
brothers = ("brother1", "brother2")

# Merging two tuples
siblings = sisters + brothers

# Counting the siblings
print("Number of siblings: ", len(siblings))

# Adding father and mother
family_members = siblings + ("father", "mother")
print("family members: ", family_members)

```

Output:

```

main x
C:\Users\Rohitha\PycharmProjects\ML\venv\Scripts\python.exe C:/Users/Rohitha/PycharmProjects/ML/main.py
Number of siblings: 4
family members: ('sister1', 'sister2', 'brother1', 'brother2', 'father', 'mother')

Process finished with exit code 0

```

4.

Used len(set_name) to find length of set.

Used .add() to add new element

For inserting multiple companies at once I have stored them into one set and used .update to add them to existing set.

Performed various set operations on it_companies.

To delete sets completely I have used A.clear()

```

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 set it_companies
print("Length of it_companies: ", len(it_companies))

# Adding twitter to it_companies
it_companies.add("Twitter")
print("After adding twitter to it_companies: ", it_companies)

# Inserting multiple it_companies at once
new_companies = {"Deloitte", "TCS"}
it_companies.update(new_companies)
print("Updated companies: ", it_companies)

# To remove a company
it_companies.remove("Deloitte")
print("it_companies after removing one company: ", it_companies);

# to implement discard
it_companies.discard("Deloitte")
print(
    "Discard operation removes element from set and does not raise an error
    if item is not present in the set, where remove deleted element from set,
    raises an error if the element is not present in the set.")

# Join A and B
print("Join A and B: ", A.union(B))

# Intersection A and B
C = A.intersection(B)
print("A intersection B: ", C)

# To check if A is subset of B
print("If A is subset of B: ", A.issubset(B))

# A and B are disjoint sets
print("If A and B are disjoint sets: ", A.isdisjoint(B))

# Join A with B and B with A
print("Join A with B", A.union(B))
print("Join B with A", B.union(A))

# Symmetric difference between A and B
print("Symmetric difference between A and B: ", A.symmetric_difference(B))

# Delete the sets completely
A.clear()
B.clear()
C.clear()

# Convert age to set
age_set = set(age)

```

```
print("Length of age set is: ", len(age_set), "and length of age list is: ", len(age))
```

Discard operation removes the element from set and remove is also used for the same. But discard does not raise an error if element is not present in the set. Where remove raises an error if element is not present in set.

Output:

```
C:\Users\Rohitha\PycharmProjects\ML\venv\Scripts\python.exe C:/Users/Rohitha/PycharmProjects/ML/main.py
Length of it_companies: 7
After adding twitter to it_companies: {'Facebook', 'Twitter', 'Microsoft', 'Oracle', 'IBM', 'Google', 'Apple', 'Amazon'}
Updated companies: {'Twitter', 'IBM', 'Facebook', 'TCS', 'Microsoft', 'Oracle', 'Google', 'Apple', 'Amazon', 'Deloitte'}
it_companies after removing one company: {'Twitter', 'IBM', 'Facebook', 'TCS', 'Microsoft', 'Oracle', 'Google', 'Apple', 'Amazon'}
Discard operation removes element from set and does not raise an error if item is not present in the set, where remove deleted element from set, raises an error if the element is
Join A and B: {19, 20, 22, 24, 25, 26, 27, 28}
A intersection B: {19, 20, 22, 24, 25, 26}
If A is subset of B: True
If A and B are 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}
Symmetric difference between A and B: {27, 28}
Length of age set is: 5 and length of age list is: 8

Process finished with exit code 0
```

Length of age set is 5 and length of age list is 8

5.

```
import math

# Value given in question
radius = 30
# Calculate area of circle and assign value to _area_of_circle_
_area_of_circle_ = math.pi * radius * radius
print("Area of circle: ", _area_of_circle_)
# Calculate circumference of circle and assign value to _circum_of_circle_
_circum_of_circle_ = 2 * math.pi * radius
print("Circumference of circle: ", _circum_of_circle_)

# Take radius as per user input and calculate area
radius = int(input("Enter radius: "))
print("Area of circle: ", math.pi * radius * radius)
```

Calculated area and circumference with given radius and computed with user input.

Output:

```
C:\Users\Rohitha\PycharmProjects\ML\venv\Scripts\python.exe C:/Users/Rohitha/PycharmProjects/ML/main.py
Area of circle: 2827.4333882308138
Circumference of circle: 188.49555921538757
Enter radius: 10
Area of circle: 314.1592653589793

Process finished with exit code 0
```

6. I have converted the given string to a set inorder to get unique words

```
x = "I am a teacher and I love to inspire and teach people"
# find unique words in the sentence
words = x.split(" ")
unique_words = set(words)
print(unique_words)

# To find unique words
print("Number of unique words: ", len(unique_words))
```

Output:

```
main x
C:\Users\Rohitha\PycharmProjects\ML\venv\Scripts\python.exe C:/Users/Rohitha/PycharmProjects/ML/main.py
{'I', 'and', 'am', 'love', 'inspire', 'teacher', 'a', 'to', 'people', 'teach'}
Number of unique words: 10

Process finished with exit code 0
```

7. Used \t – tab escape to give spaces in the words.

```
#Using tab escape sequence
print("Name\tAge\tCountry\tCity")
print("Asabeneh\t250\tFinland\tHelsinki")
```

Output:

```
C:\Users\Rohitha\PycharmProjects\ML\venv\Scripts\python.exe C:/Users/Rohitha/PycharmProjects/ML/main.py
Name    Age Country City
Asabeneh    250 Finland Helsinki

Process finished with exit code 0
```

8.Used \n to navigate to next line.

Used string formatting to get the float values in desired format.

To not get decimal places for radius I have used %s. (to get same output as shown in question)

```
#print using string formatting
radius = 10
area = 3.14 * (radius ** 2)
print("radius = %1.0f\narea = 3.14 * radius ** 2\nThe area of a circle with\nradius %s is %1.0f meters square."%(radius, radius, area))
```

Output:

```
C:\Users\Rohitha\PycharmProjects\ML\venv\Scripts\python.exe C:/Users/Rohitha/PycharmProjects/ML/main.py
radius = 10
area = 3.14 * radius ** 2
The area of a circle with radius 10 is 314 meters square.

Process finished with exit code 0
```

9.

Read N value from user and used a for loop to take those values from user.

Stored all these values in a list.

Later divided each list item with 2.2046 to convert from lbs to kgs.

```
# to read n number of list elements as input
L1 = []
N = int(input("Enter number of elements : "))

# to take values till the number of elements
for i in range(0, N):
    lbs = int(input())

    L1.append(lbs)

# converting from lbs to kgs
print(L1)
list_kg = [item / 2.2046 for item in L1]
print(list_kg)
```

Output:

```
C:\Users\Rohitha\PycharmProjects\ML\venv\Scripts\python.exe C:/Users/Rohitha/PycharmProjects/ML/main.py
Enter number of elements : 3
23
45
34
[23, 45, 34]
[10.432731561280958, 20.411866098158395, 15.422298829719676]

Process finished with exit code 0
```

10.

Initially took all points marked as values to x-axis and all points as points into dataframe called data.

Divided the data to test and train.

Used sklearn module to compute knn value for 3.

Computed predict value, confusion matrix.

From confusion matrix, calculated false positive, true negative, true positive, false negative.

Using those values sensitivity, specificity are derived.

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn import metrics

# giving dataframe the data
data = {'x-axis': ['1', '2', '3', '6', '6', '7', '10', '11'], 'points': ['o',
'o', 'x', 'x', 'x', 'o', 'o', 'o']}
df = pd.DataFrame(data)
print(df)

# arranging data in plottable form
x = df.loc[:, ["x-axis"]]
dots = df.loc[:, ["points"]]
x_train, x_test, dot_train, dot_test = train_test_split(x, dots,
random_state=0, train_size=0.5) # divided to test and train

# importing sklearn for knn
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors=3) # taking knn as 3
knn.fit(x_train, dot_train) # fit with train data
predictoutput = knn.predict(x_test) # predict with test data
print("Predicted output for test samples: ", predictoutput)

# to calculate accuracy
acc_knn = round(knn.score(x_test, dot_test) * 100, 2)
print('KNN accuracy is:', acc_knn)

# confusion matrix
confusion_matrix = metrics.confusion_matrix(x_test, dot_test)
print("Confusion matrix: ", confusion_matrix)

confusion_matrix = np.matrix(confusion_matrix)
FP = confusion_matrix.sum(axis=0) - np.diag(confusion_matrix)
FN = confusion_matrix.sum(axis=1) - np.diag(confusion_matrix)
TP = np.diag(confusion_matrix)
TN = confusion_matrix.sum() - (FP + FN + TP)

# sensitivity values
sensitivity_value = TP/(TP+FN)
print("Sensitivity value: ", sensitivity_value)

# specificity value
specificity = TN/(TN+FP)
print("Specificity value: ", specificity)
```


Output:

```
C:\Users\Rohitha\PycharmProjects\ML\venv\Scripts\python.exe C:/Users/Rohitha/PycharmProjects/ML/main.py
x-axis points
0      1      o
1      2      o
2      3      x
3      6      x
4      6      x
5      7      o
6     10      o
7     11      o
```

```
Predicted output for test samples: ['x' 'x' 'x' 'x']
```

```
KNN accuracy is: 25.0
```

```
Confusion matrix: [[0 0 0 0 1 0]
```

```
[0 0 0 0 1 0]
```

```
[0 0 0 0 1 0]
```

```
[0 0 0 0 0 1]
```

```
[0 0 0 0 0 0]
```

```
[0 0 0 0 0 0]]
```

```
Sensitivity value: [[ 0.  0.  0.  0.  0.  0.]
```

```
[ 0.  0.  0.  0.  0.  0.]
```

```
[ 0.  0.  0.  0.  0.  0.]
```

```
[ 0.  0.  0.  0.  0.  0.]
```

```
[nan nan nan nan nan nan]
```

```
[nan nan nan nan nan nan]]
```

```
Specificity value: [[1.      1.      1.      1.      0.      0.66666667]
```

```
[1.      1.      1.      1.      0.      0.66666667]
```

```
[1.      1.      1.      1.      0.      0.66666667]
```

```
[1.      1.      1.      1.      0.25     0.75     ]
```

```
[1.      1.      1.      1.      0.25     0.75     ]]
```