# MACHINE LEARNING

## Assignment 1

**Saikrishna Peruka**

700740918

```
#1
import statistics;
ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24]
print(ages);
# sorting the list
ages.sort();
print(ages);
# Min and Max
minimum = min(ages);
maximum = max(ages)
print("Minimum value in list is : " , minimum , " \nMaximum value in list is: " , maximum);
# Adding min and max to list
ages.append(minimum);
ages.append(maximum);
print(ages);
# finding the median age
median = statistics.median(ages)
print("median is ", median);
# average
s = sum(ages);
print("sum of list is : ", s);
l = len(ages);
print("len of list is : ", l);
ave = s/l;
print("Average of list is : ", ave);
#Range
ran = maximum - minimum;
print("range of list is ", ran);
```

✓ 0.1s

```
[19, 22, 19, 24, 20, 25, 26, 24, 25, 24]
[19, 19, 20, 22, 24, 24, 24, 25, 25, 26]
Minimum value in list is :  19
Maximum value in list is:  26
[19, 19, 20, 22, 24, 24, 24, 25, 25, 26, 19, 26]
median is  24.0
sum of list is :  273
len of list is :  12
Average of list is :  22.75
range of list is  7
```

```python
#2

Dog = {}

Dog.update({
    'name' : 'Leo',
    'color' : 'White',
    'breed' : 'Pomeranian',
    'legs' : '4',
    'age' : '2'
    })
#student dictionary
Student = {
    'first_name' : 'Sai_Krishna',
    'last_name' : 'Peruka',
    'gender' : 'Male',
    'age' : '22',
    'marital status' : 'Unmarried',
    'skills' : ["HTML","C","Java"],
    'country' : 'India',
    'city' : 'Warangal',
    'address' : 'Balaji Nagar, Kothawada, Warangal, 506002'
}
#length of student dictionary
print("Length of the Student dictionary is", len(Student))
#Get the value of skills and check the data type
print("skills of student are ", Student['skills'])
print("datatype of skills is", type(Student['skills']))
#modifying
Student['skills'].extend(["Python", "Hybris"])
print("Modified skills in list are", Student['skills'])
#keys as list
print("Keys in the student dictionary are", list(Student.keys()))
print("values in the student dictionary are", list(Student.values()))
```

✓ 0.1s

```
Length of the Student dictionary is 9
skills of student are  ['HTML', 'C', 'Java']
datatype of skills is <class 'list'>
Modified skills in list are ['HTML', 'C', 'Java', 'Python', 'Hybris']
Keys in the student dictionary are ['first_name', 'last_name', 'gender', 'age', 'marital status', 'skills', 'country', 'city', 'address']
values in the student dictionary are ['Sai_Krishna', 'Peruka', 'Male', '22', 'Unmarried', ['HTML', 'C', 'Java', 'Python', 'Hybris'], 'India', 'Warangal', 'Balaji Nagar, Kothawada, Warangal, 506002']
```

```python
#3

# tuple with names of sisters and brothers
sisters = ("Sneha", "Sreeja", "Minnu")
brothers = ("Nithish", "Praveen", "Sai", "Krishna")
print(sisters)
print(brothers)

# joining brothers and sisters

siblings= sisters + brothers
print(siblings)

#count of siblings

siblingscount= len(siblings)
print(siblingscount)

#appending father and mother

father= "Sudhakar"
mother = "Saritha"

family_members = list(siblings)

family_members.append(father)
family_members.append(mother)
print(family_members)
```

✓ 0.2s

```
('Sneha', 'Sreeja', 'Minnu')
('Nithish', 'Praveen', 'Sai', 'Krishna')
('Sneha', 'Sreeja', 'Minnu', 'Nithish', 'Praveen', 'Sai', 'Krishna')
7
['Sneha', 'Sreeja', 'Minnu', 'Nithish', 'Praveen', 'Sai', 'Krishna', 'Sudhakar', 'Saritha']
```

```python
#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
print("The length of set is:", len(it_companies))
#Add 'Twitter' to it_companies
it_companies.add("Twitter")
print(it_companies)
#Insert multiple IT companies at once to the set it_companies
multiple_ITcompanies= ["TCS", "Accenture", "Adobe"]
it_companies.update(multiple_ITcompanies)
print(it_companies)
#Remove one of the companies from the set it_companies
it_companies.remove("TCS")
print(it_companies)
#Difference between remove and discard
#Remove: If item to remove does not exist, remove() will raise an error
#Discard: If item to remove does not exist, discard() will NOT raise an error
#Join A and B
C = A.union(B)
print(C)
#Find A intersection B
D = A.intersection(B)
print(D)
#Is A subset of B
E = A.issubset(B)
print(E)
#Are A and B disjoint sets
F = A.isdisjoint(B)
print(F)
#Join A with B and B with A
G = B.union(A)
print(C,G)
#What is the symmetric difference between A and B
A.symmetric_difference(B)
#Delete the sets completely
del A,B
#print(B)
#Convert the ages to a set and compare the length of the list and the set
S= set(age)
print(len(S)==len(age))
```

✓ 0.1s

```
The length of set is: 7
{'Google', 'Twitter', 'IBM', 'Microsoft', 'Oracle', 'Amazon', 'Apple', 'Facebook'}
{'Oracle', 'Apple', 'Facebook', 'Google', 'Accenture', 'TCS', 'Adobe', 'Microsoft', 'Amazon', 'Twitter', 'IBM'}
{'Oracle', 'Apple', 'Facebook', 'Google', 'Accenture', 'Adobe', 'Microsoft', 'Amazon', 'Twitter', 'IBM'}
{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}
False
```

```python
#5

radius = 30
pi=3.14

#area of a circle

area_of_circle = pi*radius*radius
print("area of the circle is", area_of_circle)

#circumference of a circle

circum_of_cirlce = 2*pi*radius
print("circumference of the circle is", circum_of_cirlce)

#calculating area with radius as input


radius = float(input ("Enter the radius of the circle : "))
area= pi*radius*radius
print ("The area of the circle is", area)

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
area of the circle is 2826.0
circumference of the circle is 188.4
Enter the radius of the circle : 30
The area of the circle is 2826.0
PS C:\Users\saikr>
```

```python
#6

#printing unique words

str =  "I am a teacher and I love to inspire and teach people"
s=set(str.split(" "))
print(s)
```

✓ 0.1s

```
{'and', 'to', 'love', 'teacher', 'a', 'people', 'am', 'inspire', 'I', 'teach'}
```

```python
#7

#printing the sequence

print("Name\t\tAge\t\tCountry\t\tCity\nAsabeneh\t250\t\tFinland\t\tHelsinki")
```

✓ 0.1s

```
Name            Age             Country         City
Asabeneh        250             Finland         Helsinki
```

```python
#8

radius = 10
area = 3.14 * radius ** 2

print("Area of a circle with radius is %d meters square" %area)
```

✓ 0.7s

```
Area of a circle with the given radius is 314 meters square
```

```python
#9

import math
Students = int(input("Enter the count of students:"))
weights=[]
weights1=[]
for i in range(Students):
    weights.append(int(input()))
for j in weights:
    a=(math.floor((j/2.2046) * 100 ) )/ 100;
    weights1.append(a)
print(weights1)

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
  File "c:\Users\saikr\OneDrive\Desktop\ML Ass\Question9.py", line 4, in <modu
    Students = int(input("Enter the count of students:"))
ValueError: invalid literal for int() with base 10: '& C:/Users/saikr/AppData/
xe "c:/Users/saikr/OneDrive/Desktop/ML Ass/Question9.py"'
PS C:\Users\saikr> & C:/Users/saikr/AppData/Local/Microsoft/WindowsApps/Python
p/ML Ass/Question9.py"
Enter the count of students:4
150
155
145
148
[68.03, 70.3, 65.77, 67.13]
PS C:\Users\saikr>
```

10)

| A = datapoints | B = K |
|---|---|
| 1 | 0 |
| 2 | 0 |
| 3 | 1 |
| 6 | 1 |
| 6 | 1 |
| 7 | 0 |
| 10 | 0 |
| 11 | 0 |

Dividing data equally into training and test data

A_train = [1, 6, 10, 11]

B_train = [0, 1, 0, 0]

A_test = [2, 3, 6, 7]

B_test = [0, 1, 1, 0]

By using KNN K = 3

The output that we are predecting for A_test data

$$ed = \sqrt{(x-x_1)^2+(y-y_1)^2+(z-z_1)^2}$$ [∵ ed = echidian distance]

The A_test [0] = 2 $\Rightarrow \sqrt{(2-1)^2} = 1$

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

$\rightarrow \sqrt{(2-10)^2} = 8$

$\rightarrow \sqrt{(2-11)^2} = 9$

B_pred [0] = 0

A_test[1] = 3 $= \sqrt{(3-1)^2} = 2$

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

$= \sqrt{(3-10)^2} = 7$

$= \sqrt{(3-11)^2} = 8$

B_pred[1] = 0

A_test[2] = 6

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

$= \sqrt{(6-6)^2} = 0$

$= \sqrt{(6-10)} = 4$

$= \sqrt{(6-11)^2} = 5$

A_test[3] = 7 B_pred[2] = 0

$= \sqrt{(7-1)^2} = 6$

$= \sqrt{(7-6)^2} = 1$

$= \sqrt{(7-10)^2} = 3$

$= \sqrt{(7-11)^2} = 4$

B_pred[3] = 0

B_pred = [0, 0, 0, 0]

B_test = [0, 1, 1, 0]

true positives = 0/4 = 0

false positives = 0/4 = 0

true negatives = 2/4 = 0.5

false negatives = 2/4 = 0.5

$$\text{Accuracy} = \left(\frac{\text{true positives} + \text{true negative}}{\text{total data points}}\right) = \frac{0+0.5}{4}$$

Accuracy = 12.5

$$\text{Sensitivity} = \frac{\text{true positives}}{(\text{true positives} + \text{false negative})} = \frac{0}{0+0.5}$$

Sensitivity = 0 = 0%

$$\text{Specificity} = \frac{\text{true negative}}{(\text{false positive} + \text{true negative})} \quad \frac{0.5}{0+0.5}$$

Specificity = 0 = 0.1