



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University under section 3 of the UGC Act, 1956)

Accredited with "A" Grade by NAAC | Approved by AICTE



RECORD NOTEBOOK

NAME : Mohnish Devaraj

LAB : Machine Learning & Data Analytics Lab

YEAR : 2019 - 23 BRANCH : Computer Science Engineering

ROLL NO : 19S115398 REGISTER NO : 39110636

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" Grade by NAAC | Approved by AICTE

(A Christian Minority Institution)

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai- 600 119.



Name of the Student : MOHNISH DEVARAJ

Register Number : 39110636

Subject code : SCSA2601

Subject Name : MACHINE LEARNING AND DATA ANALYTICS

Certified that the work submitted here in, is the bonafide work of the above mentioned student, carried out in Machine learning & Data Analytics Laboratory from December 2021 to March 2022
(Month) (Month)

Signature of the Lab in Charge

Examiners:

Date: _____

Internal _____

External _____

INDEX

Name of the Laboratory : Machine Learning & Data Analytics Lab

Sub. Code : SCSA 2601

Name of the Staff In-Charge : Dr. Jemisha Mariam and Mrs. Varathi

Sl No.	Date of Experiment	Name of the Experiment	Page No.	Date of Submission	Marks	Staff Signature with Date
1	20-12-2021	Handling Input & Output	1		10	
2	20-12-2021	Looping Constructs	2			
3	20-12-2021	Arrays, sets, lists & Dictionary	3			
4	22-12-2021	Modules and Function	5		10	
5	22-12-2021	File Handling	6			
6	22-12-2021	Exception Handling	7			
7	27-12-2021	List, find second highest value	8		10	
8	29-12-2021	Count specialchar, alpha, digit	9		10	
9	03-01-2022	Wrap string into a paragraph	11		10	
10	05-01-2022	Print string, matrix ^{Numpy} , _{Excel} , ^{modul02}	12		10	
11	10-01-2022	Compute $s = f(x_1) + f(x_2) \dots + f(x_n)$	13		10	
12	15-01-2022	Validate credit system	15		10	

CYCLE - II

1	24-01-2022	Data Preprocessing	17	10	
2	31-01-2022	Manipulate twitter Dataset	19	10	
3	02-02-2022	Implement Regression & CLR Correlation technique	21	10	
4	08-02-2022	Implement Regression & CFA correlation technique	24	10	
5	15-02-2022	Evaluating the results of ML	30	10	
6	22-02-2022	Implement classification Algo	32	10	
7	29-02-2022	K-means Algorithm	35		

Jemisha
22B3P21

INDEX

Name of the Laboratory :

Sub. Code :

Name of the Staff In-Charge :

INDEX

Name of the Laboratory	: Machine Learning & Data Analytics Lab
Sub. Code	: SCSA 2601
Name of the Staff In-Charge	: Dr. Jemisha Mariam and Mrs. Vanathi

Sl No.	Date of Experiment	Name of the Experiment	Page No.	Date of Submission	Marks	Staff Signature with Date
CYCLE - I						
1(a)	20-12-2021	Handling Input & Output	1		10	
1(b)	20-12-2021	Looping Constructs	2			
1(c)	20-12-2021	Arrays, sets, Lists & Dictionary	3			
2(a)	22-12-2021	Modules and Function	5		10	
2(b)	22-12-2021	File Handling	6			
2(c)	22-12-2021	Exception Handling	7			
3	27-12-2021	List, find second highest value	8		10	
4	29-12-2021	Count special char, alpha, digits	9		10	
5	03-01-2022	Wrap string into a paragraph	11		10	
6	05-01-2022	Print string, matrix NxM, del., modulus	12		10	
7	10-01-2022	Compute : $s = f(x_1) + f(x_2) + \dots + f(x_n)$	13		10	
8	12-01-2022	Validate credit system	15		10	
CYCLE - II						
1	24-01-2022	Data Preprocessing	17		10	
2	31-01-2022	Manipulate twitter Dataset	19		10	
3a	02-02-2022	Implement Regression & (LR) Correlation technique	21		10	
3b	08-02-2022	Implement Regression & (CART) Correlation technique	24		10	
4	15-02-2022	Evaluating the results of ML	30		10	
5	22-02-2022	Implement classification Algorithm	32		10	
6	29-02-2022	K-means Algorithm	35			

Jemisha
22/3/22

Jemisha
22/3/22

INDEX

Name of the Laboratory :

Sub. Code

Name of the Staff In-Charge :



#1. a. Handling Input and output

```
print("Handling Input and output")
n=int(input("Enter a number to be multiplied by 5 :"))
print("Your answer for 5 x {0} is {1}".format(n,n*5))
```

Handling Input and output

Enter a number to be multiplied by 5 :6

Your answer for 5 x 6 is 30

Expt. No. 1(a)

Page No. 1

Expt. Name. Handling Input and Output

Date : _____

Aim:

To write a python program Handling Input and Output.

Algorithm:

step1: start

step2: get the input from the user

step3: print the output by multiplying the input value by 5

step4: stop

Program:

```
print("Handling Input and Output")
n = int(input("Enter a number to be multiplied by 5: "))
print("Your answer for 5x{} is {}".format(n, n*5))
```

Result:

The above program is executed successfully and the output is verified.

✓
Jyoti
22/3/22

#b. Looping Construts|

```
print("Looping constructs")
n=int(input("Enter a Number to get their Multiplication tables :"))
for i in range(1,11):
    print("{0} x {1} = {2}".format(n,i,n*i))
```

Looping constructs

Enter a Number to get their Multiplication tables :6

6 x 1 = 6

6 x 2 = 12

6 x 3 = 18

6 x 4 = 24

6 x 5 = 30

6 x 6 = 36

6 x 7 = 42

6 x 8 = 48

6 x 9 = 54

6 x 10 = 60

Expt. No. 1 (b)

Page No. 2

Expt. Name. Looping Constructs

Date : _____

Aim:

To write a python program using Looping constructs

Algorithm:

step 1: start

step 2: Get the input from the user

step 3: Using for loop given range between 1 and 11

step 3.1: print the multiplication of input value and number

step 4: stop

Program:

```
print("Looping Constructs")
n=int(input("Enter a number to get their Multiplication tables:"))
for i in range(1,11):
    print("{} x {} = {}".format(n, i, n*i))
```

Result:

The above program is executed successfully and the output is verified.

Ques 22/3/22

```
#c.Arrays, Lists, Sets and Dictionaries
import array as arr
print("Arrays, Lists, Sets and Dictionaries\n")
print("Arrays")
a = arr.array('i', [1, 2, 3])
for i in range (0, 3):
    print (a[i], end = " ")
print()
print("Appending Lists\n")
l1 = ["a", "b" , "c"]
l2 = [1, 2, 3]
for x in l2:
    l1.append(x)
print("appended list : ", end=" ")
print(l1)
print("\nAppending Sets\n")
s1 = {"a", "b" , "c"}
s2 = {1, 2, 3}
set3 = s1.union(s2)
print("appended Set : ", end=" ")
print(set3)
car = {'brand': 'Ford', 'model': 'Mustang', 'year': 1964}
print("\nDetails of car before updating : \n{0}".format(car))
car["color"]="white"
print("\nDetails of a car after updating : \n{0}".format(car))
```

Arrays, Lists, Sets and Dictionaries

Arrays

1 2 3

Appending Lists

appended list : ['a', 'b', 'c', 1, 2, 3]

Appending Sets

appended Set : {1, 2, 3, 'c', 'a', 'b'}

Details of car before updating :

{'brand': 'Ford', 'model': 'Mustang', 'year': 1964}

Details of a car after updating :

{'brand': 'Ford', 'model': 'Mustang', 'year': 1964, 'color': 'white'}

Expt. No. 1 (c)

Page No. 3

Expt. Name. Arrays, List, set and Dictionaries

Date:

Aim:

To write a python program using Arrays, lists, set and Dictionary.

Algorithm:

step 1: start

step 2: import array library

step 3: print the array given in the code

step 4: initialize two lists

step 5: Append the two lists and print it

step 6: Initialize two sets

step 7: Append the two sets and print it

step 8: Initialize a dictionary

step 9: update the dictionary and print it

step 10: stop

Program:

```
import array as ar
print("Arrays, Lists, sets and Dictionary")
print("Arrays")
a=ar.array('i',[1,2,3])
for i in range(0,3):
    print(a[i],end=" ")
print()
print("Appending Lists\n")
L1=[“a”, “b”, “c”]
L2=[1, 2, 3]
for x in L2:
    L1.append(x)
```

Expt. No. _____

Page No. 4

Expt. Name. _____

Date : _____

print("Append lists : ", end="")

print(L1)

print("In Appending sets. \n")

S1 = {"a", "b", "c"}

S2 = {1, 2, 3}

SCT3 = S1. Union(S2)

print("Append set : " end="")

print(SCT3)

CAR = {'brand': 'Ford', 'model': 'Mustang', 'year': 1964}

print("In Details of car before updating: In %s".format(CAR))

CAR['Color'] = "white"

print("In Details of a car after updating: In %s".format(CAR))

Result:

The above program is executed successfully and the output is verified

(JEL/22/3/22)

#2. a.Modules and Functions|

```
def add(x,y):  
    print("Sum of {0} + {1} = {2} ".format(x,y,x+y))  
def sub(x,y):  
    print("Difference of {0} - {1} = {2} ".format(x,y,x-y))  
def mul(x,y):  
    print("Product of {0} x {1} = {2} ".format(x,y,x*y))  
def div(x,y):  
    print("Quotient of {0} / {1} = {2} ".format(x,y,x/y))  
add(2,3)  
sub(5,3)  
mul(10,5)  
div(10,2)
```

Sum of 2 + 3 = 5

Difference of 5 - 3 = 2

Product of 10 x 5 = 50

Quotient of 10 / 2 = 5.0

Expt. No. 2(a)

Page No. 5

Expt. Name. Modules and Function

Date: _____

Aim:

To write a python program using Modules and Function

Algorithm:

step 1: start

step 2: Create a function for sum, difference, multiplication and Division

step 3: Perform the respective operations in their function

step 4: Call the function with the parameters

step 5: stop

Program:

```
def add(x, y):  
    print("sum of {} + {} = {}".format(x, y, x+y))  
def sub(x, y):  
    print("Difference of {} - {} = {}".format(x, y, x-y))  
def mul(x, y):  
    print("Product of {} * {} = {}".format(x, y, x*y))  
def div(x, y):  
    print("Quotient of {} / {} = {}".format(x, y, x/y))
```

add(2, 3)

sub(5, 3)

mul(10, 5)

div(10, 2)

Result:
2 2 3 5 2

The above program is executed successfully and the output is verified.



#b. File Handling

```
fo = open("data.txt", "wb")
print ("File Name: ", fo.name)
print ("Mode of Opening: ", fo.mode)
print ("Is Closed: ", fo.closed)
```

File Name: data.txt

Mode of Opening: wb

Is Closed: False

Expt. No. 2(b)

Page No. 6

Expt. Name. File Handling

Date : _____

Aim:

To write a python program using File Handling

Algorithm:

step 1: start

step 2: take the file as input

step 3: print the file name, mode of opening

step 4: stop

Program:

```
fo = open ("data.txt", "wb")
print ("Mode of opening: ", fo.name)
print ("File name: ", fo.name)
print ("Is closed: ", fo.closed)
```

Result:

The above program is executed successfully and the output is verified.

Q3/22/22



```
#c.Exception Handling|
```

```
a = [1, 2, 3]
```

```
try:
```

```
    print ("Second element = %d" %(a[1]))
```

```
    # Throws error since there are only 3 elements in array
```

```
    print ("Fourth element = %d" %(a[3]))
```

```
except:
```

```
    print ("An error occurred")
```

```
Second element = 2
```

```
An error occurred
```

Expt. No. 2(c)

Page No. 7

Expt. Name. Exception Handling

Date : _____

Aim:

To write a python program using Exception Handling

Algorithm:

step 1: start

step 2: initialize a list

step 3: Using try print the elements

step 4: Using except print the error occurred

step 5: Stop

Program:

a = [1, 2, 3]

try:

 print("Second element = ", a[1])

 print("Fourth element = ", a[3])

except:

 print("An error occurred")

Result:

The above program is executed successfully and the output is verified.

Ques 21/3 for



#3. write python code to , find the second highest value from the given input list |

```
arr=[6, 5, 2, 1, 6, 4]
arr.sort()
print("second highest element in list is :",arr[-2])
```

Second highest element in list is : 6

Expt. No. 3

Page No. 8

Expt. Name. Given list, find second highest value

Date: _____

Aim:

To write a python program to find the second highest value in a given list

Algorithm:

step 1: start

step 2: initialize the array

step 3: sort array using sort method

step 4: print the second last element in the list

step 5: Stop

Program:

```
arr = [6, 5, 2, 1, 6, 4]
```

```
arr.sort()
```

```
print("second highest element in list is:", arr[-2])
```

Result:

The above program is executed successfully and the output is verified.

✓
Jen 22/3/22

#4.Count the special characters, alphabets, digits, lowercase and uppercase characters.|

```
def count_chars(str):
    upper,lower,special_char,digit=0,0,0,0
    for i in range (len(str)):
        if str[i]>='A' and str[i]<='Z':
            upper=upper+1
        elif str[i]>='a' and str[i]<='z':
            lower=lower+1
        elif str[i]>='0' and str[i]<='9':
            digit=digit+1
        elif str[i]!=' ':
            special_char=special_char+1
    print("Upper Case chars : ",upper)
    print("Lower Case chars : ",lower)
    print("Special chars : ",special_char)
    print("Digits : ",digit)
    print("Alphabets : ",upper+lower)
str=input()
count_chars(str)
```

Sathyabama 2019 @

Upper Case chars : 1

Lower Case chars : 9

Special chars : 1

Digits : 4

Alphabets : 10

Expt. Name. Count the special character, alphabet, digits, lowercase character Date: _____

Aim:

To write a python program on counting the special character, alphabets, digits, lowercase and uppercase character

Algorithm:

Step 1: Start

step 2: Get the string input from the user

step 3: Initialize upper, lower, special-char and digit to zero

step 4: for loop for the length of the string

step 4.1: check if the char is between A and z, then
increment upper to one else go to step 4.2step 4.2: check if the char is between a and z, then increment
lower to one else go to step 4.3step 4.3: check if the char is between 0 to 9, then increment
digit to one else increment special-char to 1

step 5: print upper, lower, special-char, digit, Alphabet.

step 6: Stop

Program:

```
def count_chars(str):
    upper, lower, special_char, digit = 0, 0, 0, 0
    for i in range(len(str)):
        if str[i] >= 'A' and str[i] <= 'Z':
            upper = upper + 1
        elif str[i] >= 'a' and str[i] <= 'z':
            lower = lower + 1
        elif str[i] >= '0' and str[i] <= '9':
            digit = digit + 1
        else:
            special_char = special_char + 1
    print("Upper case letters: ", upper)
    print("Lower case letters: ", lower)
    print("Special characters: ", special_char)
    print("Digits: ", digit)
```

Expt. No. _____

Page No. 10

Expt. Name. _____ Date : _____

if str[i] != " ":

 special_char = special_char + 1

```
print ("Upper case chars : ", upper)
print ("Lower case chars : ", lower)
print ("special chars : ", special_char)
print ("Digits ", digit)
print ("Alphabets : ", upper + lower)
```

```
str = input()
count_chars(str)
```

Result:

The above program is executed successfully and the output is verified

Ques ✓ 22/3/22

#5. For given Input String (s) and Width (w). Wrap the string into a paragraph of width w.

```
import textwrap  
s=input("Input a string: ")  
w = int(input("Input the width of the paragraph: ").strip())  
print("Result: ")  
print(textwrap.fill(s,w))
```

Input a string: sathyabama

Input the width of the paragraph: 3

Result:

sat

hya

bam

a

Expt. No. 5

Page No. 11

Expt. Name. Wrap the string into a paragraph of width Date :

Aim:

To write a python program for given input string (s) and width(w). Wrap the string into a paragraph of width w.

Algorithm:

step 1: start

step 2: import textwrap library

step 3: Get the string input from the user

step 4: Get the width input from the user

step 5: print the output using .fill function

step 6: stop

Program:

```
import textwrap
s = input('Input a string :')
w = int(input('Input the width of the paragraph :')) strip()
print ('Result :')
print (textwrap.fill(s,w))
```

Result:

The above program is executed successfully and the output is verified

22/3/22

Q #6 Print of the String "Welcome". Matrix size must be N X M. (N is an odd natural number, and #M is 3 times N.).

```
a=input('Enter the string:')
n, m = map(int,input("value of n m:").split())
pattern = [('|'+'*'(2*i + 1)).center(m, '-') for i in range(n//2)]
print('\n'.join(pattern + [a.center(m, '-')] + pattern[::-1]))
```

Enter the string:Welcome

value of n m:10 20

```
-----|,-----  
----|,.,|,.,|,----  
--|,.,|,.,|,.,|,---  
|,.,|,.,|,.,|,.,|,  
|,.,|,.,|,.,|,.,|,.  
----Welcome----  
|,.,|,.,|,.,|,.,|,.,|,  
|,.,|,.,|,.,|,.,|,  
--|,.,|,.,|,.,|,.,---  
----|,.,|,.,|,----  
-----|,-----
```

Expt. No. 6

Page No. 12

Expt. Name. Print a string, matrix size $N \times M$, should use 1, and Date:

Aim:

To write a python program to print "Welcome", Matrix size $\overset{\text{should}}{N \times M}$, where N is odd number and M is 3 times of N. The design should have "Welcome" in the center. The design should only use 1, . and - characters.

Algorithm:

step 1: start

step 2: Get the string input from the user

step 3: Get the value of n and m from the user

step 4: calculate the pattern

step 5: print the pattern

step 6: stop

Program:

```
a = int(input("Enter the string: "))
n, m = map(int, input("Value of nm: ").split())
pattern = [('1.' * (2*i+1)).center(m, '-') for i in range(n//2)]
print('\n'.join(pattern + [a.center(m, '-')] + pattern[::-1]))
```

Result:

The above program is executed successfully and the output is verified.

QJL 22/3/22

```
[ ] #7.Consider a function f(X) = X3. Input is 'N' list. Each list contains 'M' elements. From the list, End  
#the maximum element. Compute  
def f(x):  
    return x**3  
N=int(input("Enter N:"))  
M=int(input("Enter M:"))  
l=[]  
mx=[]  
for i in range(N):  
    for j in range(M):  
        l.append(int(input("enter elements:")))  
mx.append(max(l))  
l=[]  
Z=int(input("Enter Z:"))  
s=0  
for i in mx:  
    s+=f(i)  
print(s%Z)
```

Enter N:2

Enter M:4

enter elements:7

enter elements:2

enter elements:8

enter elements:2

enter elements:9

enter elements:1

enter elements:9

enter elements:3

Enter Z:8

1

Expt. No. 7

Page No. 13

Expt. Name. Compute: $s = (f(x_1) + f(x_2) + \dots + f(x_n)) \text{ Modulo } 2$ Date: _____

Aim:

To write a python program by considering a function $f(x) = x^3$.
Input is 'N' list. Each list contains 'M' elements. From the list,
find the maximum element. Compute: $s = (f(x_1) + f(x_2) + \dots + f(x_n)) \text{ Modulo } 2$

Algorithm:

step 1: start

step 2: Get the input value of m & n from the user

step 3: initialize a empty list l and mx

step 4: Get the input & append it in mx

step 5: Using double for loop for N and M

step 6: Initialize a empty list l = []

step 7: Get the input value of z from the user

step 8: Initialize s is equal to zero

step 9: for loop in mx

step 8.1: increment s by increasing the power of 3

step 10: print s/z

step 10: stop

Program:

```
def f(x):
```

```
    return x**3
```

```
N = int(input("Enter N:"))
```

```
M = int(input("Enter M:"))
```

```
l = []
```

```
mx = []
```

```
for i in range(N):
```

```
    for j in range(M):
```

Expt. No. _____

Page No. 14

Expt. Name. _____

Date : _____

l.append (int(input("Enter elements: ")))

mxe.append (max(l))

l = []

z = int(input("Enter z"))

s = 0

for i in mx :

3

st = f(i)

print(s/2)

Result:

The above program is executed successfully and the output is verified.

(Ans) 22322

Aim:

To write a python program by validating the credit numbers based on the following condition: Begin with 4, 5 or 6, contain exactly 16 digits, contains only number (0 to 9). For every 4 digits a hyphen (-) may be included (not mandatory). No other special character permitted. Must not have 4 or more consecutive same digits.

Algorithm:

step 1: start

step 2: import the library re and itertools

step 3: initialize the text

step 4: print the length of the text

step 5: initialize the d to k, sum (i for i in g) for k, g in itertools.
groupby(text)

step 6: check the condition, if yes print "passed" else "False"

step 7: stop

Program:

```
import re
import itertools
text = "5133-3387-8912-3456"
print(len(text))
d = [(k, sum(i for i in g)) for k, g in itertools.groupby(text)]
if re.search(r'^\d{4}-\d{4}-\d{4}-\d{4}$', text) and len(text) == 16 and research
    (r'\d{1}\d', text) and all([v <= 3 for k, v in d]) and bool(re.search
        (r'\d{4}', text)) is False and bool(re.search(r'\d{4}-\d{4}', text)) is
        False or (bool(re.search(r'-', text))) is True and len(text) == 19):
    print("it passed")
```

Expt. No. _____

Expt. Name. _____

Page No. 16

Date : _____

else:

print("False")

Result:

The above program is executed successfully and the output is verified

✓ Yes 22/3/22

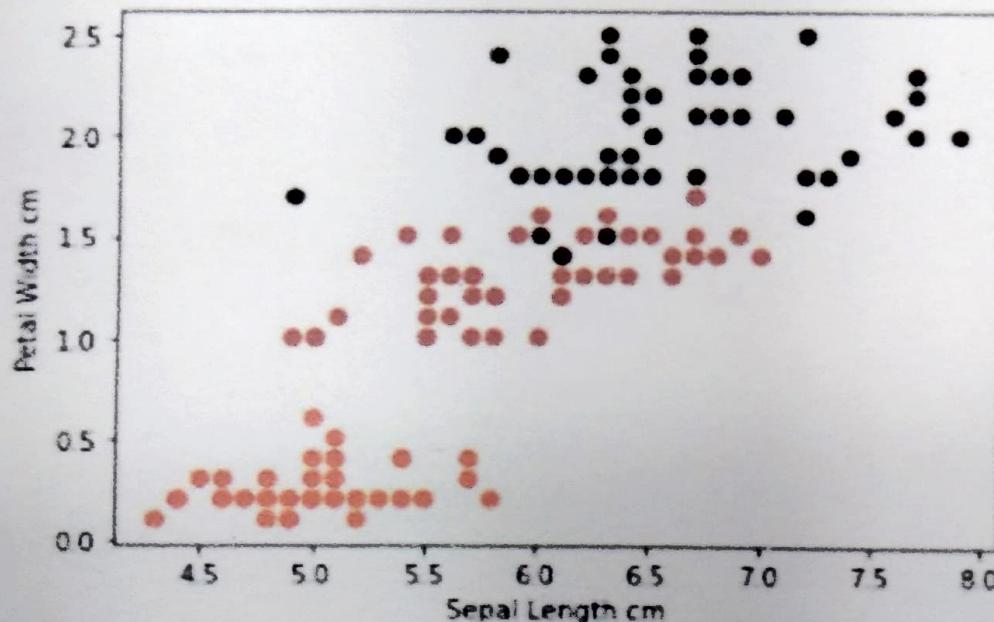
```
[ ] x_test_std = sc.transform(x_test)
x_test[0:5]
```

```
array([[ 6.1,  2.8,  4.7,  1.2],
       [ 5.7,  3.8,  1.7,  0.3],
       [ 7.7,  2.6,  6.9,  2.3],
       [ 6. ,  2.9,  4.5,  1.5],
       [ 6.8,  2.8,  4.8,  1.4]])
```

```
[ ] x_test_std[0:5]
```

```
array([[ 0.3100623 , -0.50256349,  0.484213 , -0.05282593],
       [-0.17225683,  1.89603497, -1.26695916, -1.27039917],
       [ 2.23933883, -0.98228318,  1.76840592,  1.43531914],
       [ 0.18948252, -0.26270364,  0.36746819,  0.35303182],
       [ 1.15412078, -0.50256349,  0.54258541,  0.2177459 ]])
```

```
#x = iris.iloc[:, :-1].values
#y = iris.iloc[:, 4].values
plt.scatter(x[:,0], x[:,3], c = y, cmap='flag')
plt.xlabel('Sepal Length cm')
plt.ylabel('Petal Width cm')
plt.show()
```



Aim:

To write a python program for Data Preprocessing, Building Good Training set.

Algorithm:

step 1: Start

step 2: import the necessary libraries

step 3: import the iris file from sklearn

step 4: initialize iris.data to x and iris.target to y

step 5: Split the data into train and test, test to 30%.

step 6: Using standard scaler, fit the x train

step 7: x train std to standard scaler transform x test

step 8: print the x test first five columns

step 9: print the x test std first five columns

step 10: plot the scatter plot of x-axis as sepal length and y-axis on petal width

step 11: stop

Program:

```
import pandas as pd
```

```
import numpy as np
```

```
from sklearn.model_selection import train_test_split
```

```
from sklearn.preprocessing import StandardScaler
```

```
from sklearn.datasets import load_iris
```

```
import matplotlib.pyplot as plt
```

```
iris = load_iris()
```

~~```
x = iris.data
```~~~~```
y = iris.target
```~~

Expt. No. _____

Page No. 18

Expt. Name. _____

Date : _____

$x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3,$
 $random_state = 42)$

$sc = \text{StandardScaler}()$

$sc.fit(x_train)$

$x_train_std = sc.transform(x_test)$

$\text{print}(x_test_std[0:5])$

$\text{print}(x_test[0:5])$

$\text{plt.scatter}(x[:, 0], x[:, 3], c=y, cmap='flag')$

$\text{plt.xlabel}("Sepal\ length\ cm")$

$\text{plt.ylabel}("Petal\ width\ cm")$

$\text{plt.show}()$

Result:

The above program of Data pre-processing is successfully completed.

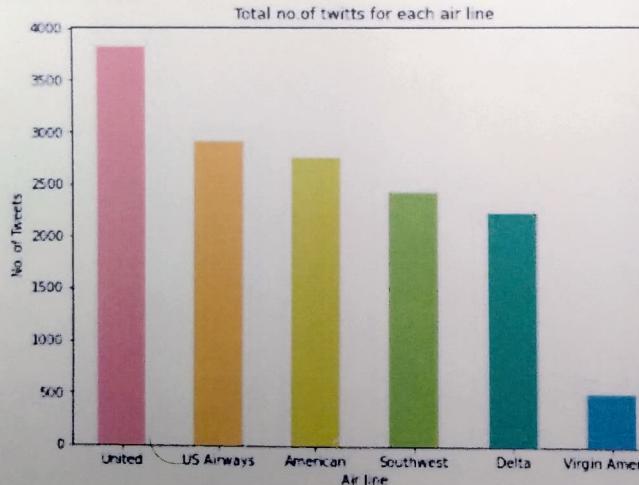
✓
Jyoti 22/3/22

```
[5] tweet.tail()  
tweet.shape  
tweet.airline.value_counts()
```

```
United      3822  
US Airways  2913  
American    2759  
Southwest   2420  
Delta       2222  
Virgin America 504  
Name: airline, dtype: int64
```

```
pd.Series(tweet["airline"].value_counts()).plot(kind="bar", color=colors, figsize=(8,6), fontsize=10, rot=0, title="Total no.of twitts for each air line")  
plt.xlabel("Air line", fontsize=10)  
plt.ylabel("No. of Tweets", fontsize=10)  
tweet.airline_sentiment.value_counts()  
colors = sns.color_palette("husl",10)  
pd.Series(colors)
```

```
0    (0.9677975592919913, 0.44127456009157356, 0.53...  
1    (0.8616090647292522, 0.536495730113334, 0.1954...  
2    (0.6804189127793346, 0.6151497514677574, 0.194...  
3    (0.46810256823426105, 0.6699492535792404, 0.19...  
4    (0.20125317221201128, 0.6907928815379025, 0.47...  
5    (0.21844753832183283, 0.6773105080456748, 0.64...  
6    (0.2197995660828324, 0.6625157876858336, 0.773...  
7    (0.433280341176423, 0.6065273407962815, 0.9585...  
8    (0.8004936186423958, 0.47703363533737203, 0.95...  
9    (0.962272393509669, 0.3976451968965351, 0.8008...  
dtype: object
```



Aim?

To write a python program manipulate the twitter Data set.

Algorithm:

- step 1: start
 - step 2: import necessary libraries
 - step 3: import the twitter dataset
 - step 4: initialize the twitter dataset
 - step 5: print the twitter dataset to tweet
 - step 6: print the bar graph using airline value counts with x-axis as airline and y-axis as No. of tweets
 - step 7: stop

Program :

```
import sklearn  
import matplotlib.pyplot as plt  
%matplotlib inline  
import pandas as pd  
from sklearn.model_selection import train_test_split  
import numpy as np  
import seaborn as sns  
from nltk.corpus import stopwords  
from wordcloud import wordcloud, STOPWORDS  
from google.colab import files  
uploaded = files.upload()  
tweet = pd.read_csv("Tweets.csv")  
tweet.tail()  
tweet.shape()
```

Expt. No. _____

Page No. 20

Expt. Name. _____

Date : _____

tweet.airline.value_counts()

```
pd.Series(tweet['airline'].value_counts(), plot(kind='bar', color=colors,  
figsize=(8,6), fontsize=10, rot=0, title="Total no. of tweets for each airline")  
plt.xlabel("Airline", fontsize=10)  
plt.ylabel("No. of tweets", fontsize=10)  
tweet.airline.sentiment.value_counts()  
colors = sns.color_palette("husl", 10)  
pd.Series(colors)
```

Result :

The above program of manipulating twitter data is successfully executed.

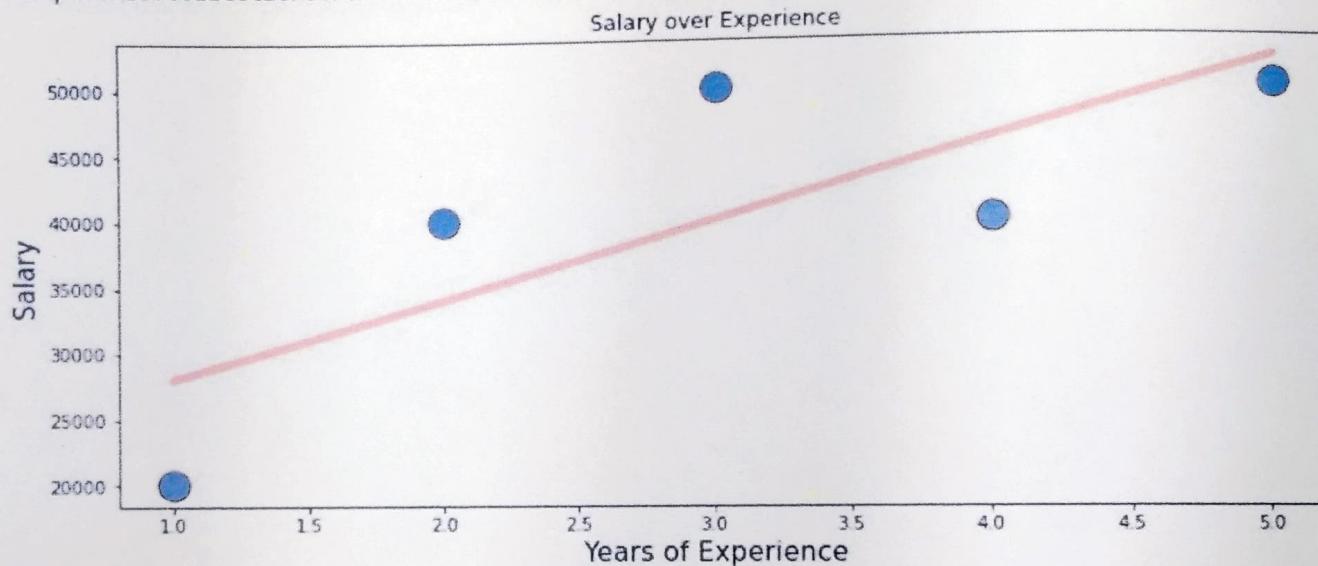
Jan 22/3/22

+ Code + Text

```
plt.text(x=1, y=10, s=text, fontsize=12, bbox={ 'facecolor': 'grey', 'alpha':0.2, 'pad':10})  
plt.title("Salary over Experience")  
plt.xlabel('Years of Experience', fontsize=15)  
plt.ylabel('Salary', fontsize=15)  
plt.plot(x,B0 + B1*x, c='r', linewidth=5, alpha=.5, solid_capstyle='round')  
plt.scatter(x=x.mean(), y=y.mean(), marker ='', s=10*2.5, c='r')
```

{x}

↳ <matplotlib.collections.PathCollection at 0x7fee67aa7210>



X : 3.0 Years
Y : \$40000.0
 $y = 22000.0 + 6000.0X$

Expt. No. 3 a

Page No. 21

Expt. Name. Implementing Regression and Correlation Technique Date :

Aim:

To write a python program Implementing Regression and Correlation Technique.

Algorithm:

step 1: start

step 2: import the necessary libraries

step 3: Using dictionary input the job title

step 4: Calculate the frame of the dictionary

step 5: define the class linear regression and find the x mean and y mean

step 6: define the class predict and find the prediction, reg-line, B_0, B_1

step 7: Using the library matplotlib, plot the graph with attributes x, $B_0 + B_1 * x$.

step 8: stop

Program:

import pandas as pd

import numpy as np

dict = {"Experience": [1, 2, 3, 4, 5], "Salary": [20000, 40000, 50000, 40000, 50000]}

df = pd.DataFrame(dict)

df

x = df.iloc[:, 0].values

y = df.iloc[:, 1].values

Expt. No. _____

Page No. 22

Expt. Name. _____

Date : _____

def Linear Regression(x, y):

N = len(x)

x_mean = x.mean()

y_mean = y.mean()

B1_num = ((x - x_mean) * (y - y_mean)).sum()

B1_den = ((x - x_mean) ** 2).sum()

B1 = B1_num / B1_den

B0 = y_mean - (B1 * x_mean)

reg_line = 'y = {} + {} x'.format(B0, round(B1, 3))

return (B0, B1, reg_line)

def Predict(B0, B1, new_x):

y = B0 + B1 * new_x

return y

pred = predict(B0, B1, 8)

pred

B0, B1, regline = Linear Regression(x, y)

import matplotlib.pyplot as plt

plt.figure(figsize=(12, 5))

plt.scatter(x, y, s=300, linewidth=1, edgecolor='black')

text = "x: {} Years Y: {} y = {} + {} x".format(round(x.mean(), 2),
round(y.mean(), 2), round(B0, 3), round(B1, 3))

plt.text(x=1, y=10, s=text, fontsize=12, bbox={'facecolor': 'grey', 'alpha': 0.2, 'pad': 10})

plt.title("Salary over Experience")

plt.xlabel("Years of Experience", fontsize=15)

plt.ylabel("Salary", fontsize=15)

plt.plot(x, B0 + B1 * x, c='r', linewidth=5, alpha=.5, solid_capstyle='round')

Expt. No. _____

Page No. 23

Expt. Name. _____

Date : _____

plt.scatter(x=x.mean(), y=y.mean(), marker='^', s=s-10*2.5, c=r')

Result:

The above program of evaluating the results of machine learning algorithm is executed successfully.

Geo
22/3/22

```
['I1', 'I2', 'I3', 'I4', 'I5']
```

C1:

```
['I1']: 6  
['I2']: 7  
['I3']: 6  
['I4']: 2  
['I5']: 2
```

L1:

```
['I1']: 6  
['I2']: 7  
['I3']: 6  
['I4']: 2  
['I5']: 2
```

C2:

```
['I4', 'I3']: 0  
['I5', 'I1']: 2  
['I1', 'I3']: 4  
['I2', 'I4']: 2  
['I2', 'I1']: 4  
['I5', 'I4']: 0  
['I2', 'I3']: 4  
['I2', 'I5']: 2  
['I4', 'I1']: 1  
['I5', 'I3']: 1
```

L2:

```
['I5', 'I1']: 2  
['I1', 'I3']: 4  
['I2', 'I4']: 2  
['I2', 'I1']: 4  
['I2', 'I3']: 4  
['I2', 'I5']: 2
```

C3:

```
['I2', 'I5', 'I1']: 2  
['I2', 'I1', 'I3']: 2  
['I2', 'I4', 'I3']: 0  
['I2', 'I5', 'I3']: 1  
['I5', 'I1', 'I3']: 1  
['I2', 'I4', 'I1']: 1  
['I2', 'I5', 'I4']: 0
```

L3:

```
['I2', 'I5', 'I1']: 2  
['I2', 'I1', 'I3']: 2
```

C4:

```
['I5', 'I1', 'I3', 'I2']: 1
```

L4:

Result:

L3:

```
['I2', 'I5', 'I1']: 2  
['I2', 'I1', 'I3']: 2
```

Expt. No. 3b

Page No. 24

Expt. Name. Implementing Regression and Correlation Techniques (Aproni) Date: _____

Aim:

To write a python program Implementing Regression and Correlation Techniques. (Aproni).

Algorithm:

step 1: start

step 2: import necessary libraries

step 3: initialize the data by giving the values

step 4: sort the data

step 5: Calculate the len and initialize it to s

step 6: print the c1 & l1 with respect to the item

step 7: Similarly, using loop combine the items and display c2 & l2

step 8: Continue, the loop until maximum items match

step 9: Calculate the support & Confidence

step 10: Display support & confidence

step 11: stop

Program:

```
data = [
    ['T100', ['I1', 'I2', 'I5']],
    ['T200', ['I2', 'I4']],
    ['T300', ['I2', 'I3']],
    ['T400', ['I1', 'I2', 'I4']],
    ['T500', ['I1', 'I3']],
    ['T600', ['I2', 'I3']],
    ['T700', ['I1', 'I3']],
    ['T800', ['I1', 'I2', 'I3', 'I5']],
    ['T900', ['I1', 'I2', 'I3']] ]
```

['I2', 'I5'] -> ['I1'] = 100.0%

['I1'] -> ['I2', 'I5'] = 33.33333333333333%

['I2', 'I1'] -> ['I5'] = 50.0%

['I5'] -> ['I2', 'I1'] = 100.0%

['I5', 'I1'] -> ['I2'] = 100.0%

['I2'] -> ['I5', 'I1'] = 28.57142857142857%

choosing: 1 4 5

['I2', 'I1'] -> ['I3'] = 50.0%

['I3'] -> ['I2', 'I1'] = 33.33333333333333%

['I2', 'I3'] -> ['I1'] = 50.0%

['I1'] -> ['I2', 'I3'] = 33.33333333333333%

['I1', 'I3'] -> ['I2'] = 50.0%

['I2'] -> ['I1', 'I3'] = 28.57142857142857%

choosing: 1 3 5

Expt. No. _____

Page No. 25

Expt. Name. _____

Date : _____

```
init = []
for i in data:
    for q in i[1]:
        if (q not in init):
            init.append(q)
init = sorted(init)
print(init)
sp = 0.4
s = int(sp * len(init))
s
from collections import Counter
c = Counter()
for i in init:
    for d in data:
        if (i in d[1]):
            c[i] += 1
print("C1:")
for i in c:
    print(str(i) + ":" + str(c[i]))
print()
l = Counter()
for i in c:
    if (c[i] >= s):
        l[frozenset([i])] += c[i]
print("L1:")
for i in l:
    print(str(list(i)) + ":" + str(l[i]))
print()
pl = l
```

pos = 1

for count in range(2, 1000):

nc = set()

temp = list(l)

for i in range(0, len(temp)):

for j in range(i+1, len(temp)):

t = temp[i].union(temp[j])

if (len(t) == count):

nc.add(temp[i].union(temp[j]))

nc = list(nc)

c = Counter()

for i in nc:

c[i] = 0

for g in data:

temp = sd(g[i])

if (i.issubset(temp)):

c[i] += 1

print(f"C{str(count)}")

for i in c:

print(str(list(c)) + ":" + str(c[i]))

print()

l = Counter()

for i in c:

if (c[i] >= s):

l[i] += c[i]

print("L" + str(count) + ":" + "

for i in l:

print(str(list(i)) + ":" + str(l[i]))

~~print()~~

Expt. No. _____

Page No. 27

Expt. Name. _____

Date : _____

if (len(l) == 0):

 break

pl = l

pos = count

print ("Result: ")

print ("L" + str(pos) + ":")

for i in pl:

 print (str(i) + ":" + str(pl[i]))

print()

from itertools import combination

for l in pl:

c = [frozenset(g) for g in combination (l, len(l)-1)]

mmax = 0

for a in c:

 b = l-a

 ab = l

 sab = 0

 sa = 0

 sb = 0

 for q in data:

 temp = set (q[i])

 if (a.issubset(temp)):

 sa += 1

 if (b.issubset(temp)):

 sb += 1

 if (ab.issubset(temp)):

 sab += 1

 temp = Sab / sa * 100

 if (temp > mmax)

 mmax = temp

Expt. No. _____

Page No. 28

Expt. Name. _____ Date : _____

$$\text{temp} = \frac{sab}{sb} * 100$$

if ($\text{temp} > \text{mmax}$)

$$\text{mmax} = \text{temp}$$

print(str(list(a)) + " → " + str(list(b)) + " = " + str(sab / sa * 100) + "%")

print(str(list(b)) + " → " + str(list(a)) + " = " + str(sab / sb * 100) + "%")

$$\text{curr} = 1$$

print("Choosing:", end=" ")

for a in c:

$$b = l - a$$

$$ab = l$$

$$sa = 0$$

$$sa = 0$$

$$sb = 0$$

for g in data:

$$\text{temp} = \text{set}(g[i])$$

if (a.issubset(temp)):

$$sa += 1$$

if (b.issubset(temp)):

$$sb += 1$$

if (ab.issubset(temp)):

$$sab += 1$$

$$\text{temp} = \frac{sab}{sa} * 100$$

if ($\text{temp} \geq \text{mmax}$)

~~mmax = temp~~ print(curr, end=" ")

$$\text{curr} += 1$$

$$\text{temp} = \frac{sab}{sb} * 100$$

if ($\text{temp} \geq \text{mmax}$):

~~print(curr, end=" ")~~

$$\text{curr} += 1$$

Expt. No. _____

Page No. 29

Expt. Name. _____

Date : _____

print()

print()

Result:

The Above program to evaluating the results of machine Learning algorithm
is executed successfully.

✓
10/10
22/3/22

Confusion Matrix: [6, 7, 5, 2]

Accuracy: 0.65

Precision: 0.5454545454545454

Recall: 0.75

Sensitivity: 0.75

Specificity: 0.5833333333333334

Missclassification Error: 0.35

Expt. No. 4

Page No. 30
algorithm Date:

Expt. Name. Evaluating the results of machine Learning

Aim:

To Evaluate the results of machine Learning algorithm

Algorithm:

step 1: start

step 2: Input the file Y into program

step 3: predict the Y

step 4: Initialize the TP, TN, FP, FN as '0'.

step 5: Using for loop Calculate the confusion matrix

step 6: Calculate the Accuracy, precision, Recall Misclassification error and specificity

step 7: print the result Acc, PRE, REC, SN, SP & MCB.

step 8: stop

Program:

```
y = ['0', '1', '0', '1', '1', '0', '1', '0', '1', '1', '0', '1', '0', '1', '1', '0', '1', '1', '0']
```

```
y-pred = ['0', '0', '0', '0', '1', '0', '1', '1', '1', '1', '0', '0', '1', '0', '0', '0', '1', '0']
```

j = 0

TP, TN, FP, FN = 0, 0, 0, 0

for i in y:

~~if (i == '1' and y-pred[j] == '1'):~~

~~TP += 1~~

~~elif (i == '0' and y-pred[j] == '0'):~~

~~TN += 1~~

~~elif (i == '1' and y-pred[j] == '0'):~~

~~FP += 1~~

Expt. No. _____

Page No. 31

Expt. Name. _____

Date : _____

if ($i == '0'$ and $y_pred[j] == '1'$):
 $FN += 1$

$j+1$

Confusion-matrix = [TP, TN, FP, FN]

print('Confusion-matrix', Confusion-matrix)

ACC = $(TP + TN) / (TP + TN + FP + FN)$

print("Accuracy: ", ACC)

PREC = $(TP) / (TP + FP)$

print("Precision: ", PREC)

REC = $TP / (TP + FN)$

print("Recall: ", REC)

SN = $TP / (TP + FN)$

print("Sensitivity: ", SN)

SP = $TN / (TN + FP)$

print("Specificity: ", SP)

print("Misclassification Error: ", MCE)

MCE = 1 - ACC

Result:

The above program evaluating the results of machine learning
is executed successfully.

✓

QFC 22/3/22

```
[2] df = pd.DataFrame()
df['refund'] = ['yes', 'no', 'no', 'yes', 'no', 'yes', 'no', 'no', 'no']
df['marital_status'] = ['single', 'married', 'single', 'married', 'divorced', 'married', 'divorced', 'single', 'single']
df['taxable_income'] = [125000, 100000, 70000, 120000, 95000, 60000, 220000, 85000, 75000, 90000]
df['evade'] = [ 'no', 'no', 'no', 'no', 'yes', 'no', 'yes', 'no', 'yes' ]
df
```

| | refund | marital_status | taxable_income | evade |
|---|--------|----------------|----------------|-------|
| 0 | yes | single | 125000 | no |
| 1 | no | married | 100000 | no |
| 2 | no | single | 70000 | no |
| 3 | yes | married | 120000 | no |
| 4 | no | divorced | 95000 | yes |
| 5 | no | married | 60000 | no |
| 6 | yes | divorced | 220000 | no |
| 7 | no | single | 85000 | yes |
| 8 | no | married | 75000 | no |
| 9 | no | single | 90000 | yes |

```
for i in range(len(df)):
    df.loc[i,'taxable_income']=str(ceil(df.loc[i,'taxable_income']/100000))
df
```

| | refund | marital_status | taxable_income | evade |
|---|--------|----------------|----------------|-------|
| 0 | yes | single | 2 | no |
| 1 | no | married | 1 | no |
| 2 | no | single | 1 | no |
| 3 | yes | married | 2 | no |
| 4 | no | divorced | 1 | yes |
| 5 | no | married | 1 | no |
| 6 | yes | divorced | 3 | no |
| 7 | no | single | 1 | yes |
| 8 | no | married | 1 | no |
| 9 | no | single | 1 | yes |

Expt. No. 5

Expt. Name. Implement Classification Algorithm

Page No. 32

Date: _____

Aim:

To write a python program implementing classification algorithm

Algorithm:

step 1: start

step 2: import the necessary packages

step 3: Calculate the dataframe using df define the dataframe
refund, flexible income , evade.

step 4: Using for loop find the length of df

step 5: using ceil function calculate 'flexible-income'

step 6: Calculate the strna , strnb , prby [] , prbn []

step 7: Calculate the dummy values in df

step 8: print the evade of x

step 9: print the prob + na(data)

step 10: stop

Program:

```
import pandas as pd
import numpy as np
from math import *
df = pd.DataFrame()
df['refund'] = ['Yes', 'no', 'no', 'yes', 'no', 'no', 'no', 'no', 'no']
df['marital-status'] = ['single', 'married', 'single', 'married', 'divorced',
                       'married', 'divorced', 'single', 'married', 'single']
df['flexible-income'] = [125000, 100000, 70000, 120000, 95000, 60000,
                        220000, 85000, 75000, 90000]
df['evade'] = ['no', 'no', 'no', 'no', 'yes', 'no', 'yes', 'no', 'yes']
```

df

```

data = pd.get_dummies(df[df.columns])
data

```

| | refund_no | refund_yes | martial_status_divorced | martial_status_married | martial_status_single | taxable_income_1 | taxable_income_2 | taxable_income_3 | evade_no | evade_yes |
|---|-----------|------------|-------------------------|------------------------|-----------------------|------------------|------------------|------------------|----------|-----------|
| 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 2 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 3 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 4 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 5 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 6 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 7 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 8 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 9 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |

```

x=['no', 'married', 140000]
x[2]=str(ceil(x[2]/100000))
x

```

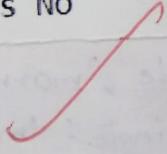
['no', 'married', '2']

```

print('no : ',prb*pa(data,'evade','no'))
print('Evade of X is No')

```

yes : 0.0
 no : 0.08163265306122447
 Evade of X is No



Expt. No. _____

Page No. 83

Expt. Name. _____

Date : _____

```
for i in range(len(df)):
    df.loc[i, 'Taxable_income'] = str(ceil(df.loc[i, 'Taxable_income']) / 100000))
df
data = pd.get_dummies(df[df.columns])
data
for i in range(1, 4):
    if ('taxable_income' + str(i)) not in data.columns:
        data['taxable_income' + str(i)] = [0 in range(10)]
x = ['no', 'married', 14000]
x[2] = str(ceil(x[2] / 100000))
x
def pa(data, cls, sbcls):
    strn = str(cls) + '-' + str(sbcls)
    return sum(data[strn]) / len(strn))
def pa-b(data, clsa, sbcls, clsb, sbclsb):
    strna = str(clsa) + '-' + str(sbcls / sa)
    strnb = str(csb) + '-' + str(sbcls / sb)
    amb = 0
    for i in range(len(data)):
        if (data.loc[i, strna] & data.loc[i, strnb]):
            amb += 1
    return (amb / sum(data[strnb]))
prby = []
prbn = []
col = df.columns
for i in range(len(x)):
    prby.append(pa-b(data, col[i], x[i], 'evade', 'yes'))
for i in prby:
    prb *= i
```

Expt. No. _____

Expt. Name. _____

Page No. 34

Date: _____

```
print("Yes:", prb * pa(data, 'evade', 'Yes'))  
for i in range(len(x)):  
    prb.append(pa_b(data, col[i], x[i], 'evade', 'no'))  
prb = 1  
for i in prbn:  
    prb *= i  
print("no:", prb * pa(data, 'evade', 'no'))  
print('Evade of x is No')
```

Result:

The above program of Implementing Classification Algorithm is successfully executed.

✓
Date 22/3/22

```
[ ] y=[]
for i in p1:
    x.append(i[0])
    y.append(i[1])
plt.scatter(x,y,c='b',label='Cluster 1')
x=[]
y=[]
for i in p2:
    x.append(i[0])
    y.append(i[1])
plt.scatter(x,y,c='r',label='Cluster 2')
plt.legend()
plt.show()
```

After Epoch 1 ERROR: 7.5

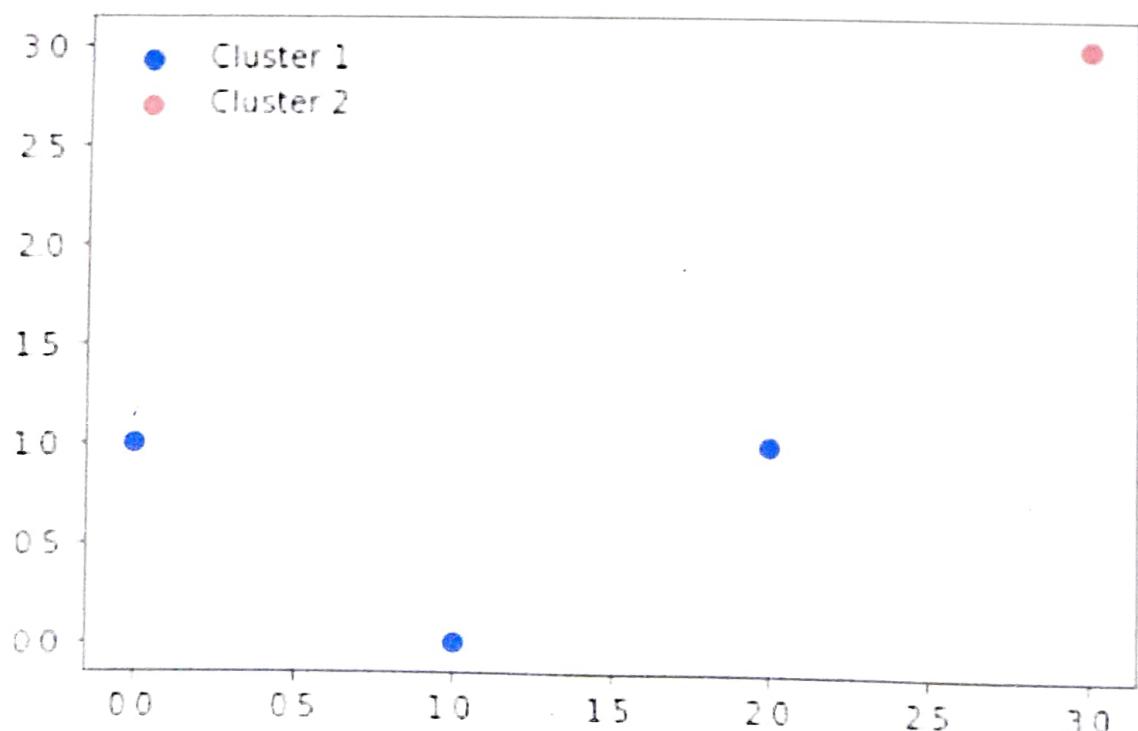
After Epoch 2 ERROR: 2.666666666666667

Data Objects:

- ('x1', [1, 0])
- ('x2', [0, 1])
- ('x3', [2, 1])
- ('x4', [3, 3])

Cluster 1: ['x1', 'x2', 'x3']

Cluster 2: ['x4']



Expt. No. 6

Expt. Name. K-means algorithm

Page No. 35

Date :

Aim:

To write a python program implementing k-means algorithm

Algorithm:

- step 1: start
- step 2: select the number k to decide the number of cluster
- step 3: select random k points or centroid
- step 4: calculate the variance and place a new centroid of each cluster
- step 5: Assign each data point to their closest centroid will assign k cluster
- step 6: Repeat step 3 until it nears to centroid of cluster
- step 7: if any reassignment occurs, then go to step 4, goto finish
- step 8: stop

Program:

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

class Kmeanalg:

```
    def __init__(self):
```

```
        self.m1 = None
```

```
        self.m2 = None
```

```
        self.c1, self.c2, self.ncl, self.nc2 = [], [], [], []
```

```
        self.pts = {}
```

```
        self.err = 0
```

```
    def error(self):
```

```
        e1 = 0
```

```
        e2 = 0
```

```
        for i in self.c1:
```

Expt. No. _____

Page No. 36

Expt. Name. _____

Date : _____

$$e1t = ((i[0] - \text{self}.m1[0])^{**2} + (i[1] - \text{self}.m1[1])^{**2})$$

for i in self.c2:

$$e2t = ((i[0] - \text{self}.m2[0])^{**2} + (i[1] - \text{self}.m2[1])^{**2})$$

return (e1t + e2t)

def fitting(self, c1, c2, *args):

def mean(c1):

n = len(c1)

a = 0

b = 0

for i in c1:

$$at = i[0]/n$$

$$bt = i[1]/n$$

return [a, b]

def dist(a, b):

$$\text{return } ((a[0] - b[0])^{**2} + (a[1] - b[1])^{**2})^{**(1/2)}$$

self.c1 = c1

self.c2 = c2

- con = 1

for i in args:

self pts ['x' + str(-con)] = i

- con += 1

epoch = 1

while (True):

self.m1 = mean(self.c1)

self.m2 = mean(self.c2)

for i in args:

p = dist(i, self.m1)

q = dist(i, self.m2)

if (p > q):

Expt. No. _____

Expt. Name. _____

Page No. 37

Date : _____

```
self.nc2.append(i)
else:
    self.nc1.append(i)
print('After Epoch', epoch, 'Error:', self.error())
epoch += 1
if (self.c1 == self.nc1 and self.c2 == self.nc2):
    break;
else:
    self.c1, self.c2 = self.nc1.copy(), self.nc2.copy()
    self.nc1, self.nc2 = [], []
for i, j in sorted(self.ptc.items()):
    for k in range(len(self.c1)):
        if (j == self.c1[k]):
            self.c1[k] = i
    for l in range(len(self.c2)):
        if (j == self.c2[l]):
            self.c2[l] = i
def display(self):
    print("Cluster 1:", self.c1, "In Cluster 2:", self.c2)
```

$$x_1 = [1, 0]$$

$$x_2 = [0, 1]$$

$$x_3 = [2, 1]$$

$$x_4 = [3, 3]$$

$$c_1 = [x_1, x_2]$$

$$c_2 = [x_2, x_4]$$

model = k_meanalg()

model.fitting(c1, c2, x1, x2, x3, x4)

print("Data Objects: ")

```
for i in sorted(model pts.items()):
```

```
    print(i)
```

```
model.display()
```

```
p1 = [model pts[i] for i in model.c1]
```

```
p2 = [model pts[i] for i in model.c2]
```

```
x = []
```

```
y = []
```

```
for i in p1:
```

```
    x.append(i[0])
```

```
    y.append(i[1])
```

```
plt.scatter(x, y, c='b', label='cluster 1')
```

```
x = []
```

```
y = []
```

```
for i in p2:
```

```
    x.append(i[0])
```

```
    y.append(i[1])
```

```
plt.scatter(x, y, c='r', label='cluster 2')
```

```
plt.legend()
```

```
plt.show()
```

Result:

The above program implementing of k-means algorithm is executed successfully.

```
Command Prompt - mongo
which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
=====
The server generated these startup warnings when booting:
2022-03-16T21:15:34.165+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
...
...
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
>
> show dbs
admin 0.000GB
chandra 0.000GB
config 0.000GB
local 0.000GB
> use chandra
switched to db chandra
> db.chandra.insert({"reg number":"39110640","sec":"B5"})
WriteResult({ "nInserted" : 1 })
>
```

Set

```
C:\Program Files\MongoDB\Server\5.0\bin\mongo.exe
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
>
> show dbs
admin 0.000GB
chandra 0.000GB
config 0.000GB
local 0.000GB
> use chandra
switched to db chandra
> use chandra39110640
switched to db chandra39110640
> show dbs
admin 0.000GB
chandra 0.000GB
config 0.000GB
local 0.000GB
> db.createCollection("Student1")
{ "ok" : 1 }
> show collections
Student1
> db.chandra39110640.insert({"name":"chandra","reg number":"39110640"})
WriteResult({ "nInserted" : 1 })
> show collections
Student1
chandra39110640
>
```

Expt. No. 01

CYCLE - II

Page No. 39

Expt. Name. Python with Mongo DB

Date: 08-03-2022

Aim:

To implement Mongo DB using python

Algorithm:

step 1: start

step 2: install the Mongo DB

step 3: set the path of Mongo DB in command prompt

step 4: show the dbs in command prompt

step 5: change the db to ~~username~~ of us.

step 6: insert the details into the db of us

step 7: write result of db inserted

step 8: stop

Program:

> Set path = "C:\Program Files\MongoDB\Server\5.0\bin"

> mongo

> show dbs

admin 0.0000 GB

Mohnish 0.0000 GB

config 0.0000 GB

local 0.0000 GB

> use ~~db~~ mohnish

switched to db mohnish

> db.mohnish.insert({ "reg number": "39110636", "sec": "B5" })

Result:

The program executed and got the output successfully.

Registration Options

INSERT

DISPLAY

DELETE

Exit

Registration Form

Name

Register no.

Email Id.

Phone No.

10th CGPA

12th CGPA

Address

Database

DISPLAY FAILED

INSERT

DISPLAY

DELETE

Exit

Registration Options

INSERT

DISPLAY

DELETE

Exit

DISPLAY Window

-

□

×

Database

```
{  
  '_id': ObjectId('623edcaecfa7f31aa7c04a17'),  
  'name': 'Anugraha. J',  
  'Regno': 39110073,  
  'Email': 'test@gmail.com',  
  'Phoneno': 12345678,  
  '10thCGPA': 10.0,  
  '12thcgpa': 9.3,  
  'Address': 'xyz street'  
}
```

Registration Options

INSERT

DISPLAY

DELETE

Exit

Update Window

Delete Form

Register no.

Submit

DELETED Successfully

Aim:

To implement the Tkinter module with MS Access.

Algorithm:

step 1: start

step 2: import the necessary packages

step 3: Insert the Details' boxes like Name, Mail id, regno, etc.

step 4: Create the table & databases which contain details

step 5: Display the created database details

step 6: Delete the unnecessary details in database.

step 7: Finally, exit the screen of the database

step 8: Stop

Algorithm:

```
import pyodbc as md
```

```
from tkinter import *
```

```
from tkinter import ttk
```

```
ws = TK()
```

```
ws.title("student Registration Form")
```

```
ws.geometry("400x400")
```

```
ws['bg'] = "#0ff"
```

```
DBpath = r'DBQ=D:\COLLEGE\SEM 6\ML and DA (LAB)\h\MS Access\database1.accdb'
```

```
def insert():
```

```
ins = Toplevel()
```

```
name = StringVar()
```

```
regno = IntVar()
```

```
mail = StringVar()
```

```
phone = StringVar()
```

Expt. No. _____

Page No. 41

Expt. Name. _____

Date : _____

tencgpa = Double Var()

twnicgpa = Double Var()

addr = String Var()

ins.title ("update Window")

ins.geometry ("1000x900")

ins['bg'] = "#0A9"

def run():

addr = hl.get('1.D', 'end-1c')

conn_str =

r'DRIVER=Microsoft Access Driver (*.mdb, *.accdb);'

r'DBQ=D:\COLLEGE\SEM 6\ML and DA (L&B)\MsAccess\Database1.accdb'

)

mydb = md.connect(conn_str)

db = mydb.cursor()

try:

db.execute("CREATE TABLE details(

Name VARCHAR(255),

Reg no int,

Email-ID VARCHAR(225),

Phone NO VARCHAR(10),

10th-CGPA FLOAT,

12th-CGPA FLOAT,

Address TEXT);")

except:

pass

sql =

sql = "INSERT INTO details VALUES (?, ?, ?, ?, ?, ?, ?, ?)"

```
val = (name.get(), regno.get(), mail.get(), phone.get(), tencgpa.get(),
twncgpa.get(), addr)
```

```
db.execute(sql, val)
```

```
xyz = db.execute("select * from details").fetchall()
```

```
print(xyz)
```

```
mydb.commit()
```

```
mydb.close()
```

```
print(name.get(), regno.get(), mail.get(), phone.get(), tencgpa.get(), twncgpa.get(),
addr)
```

```
label1 = Label(inc, text='Registration Form', font=('Arial', 20)).grid(row=0, column=1, padx=20, pady=10)
```

```
a = Label(inc, text="Name", font=('Arial', 12), width=10).grid(row=1, column=0, padx=20, pady=30)
```

```
a1 = Entry(inc, width=100, textvariable=name).grid(row=1, column=1, padx=20, pady=30)
```

```
b = Label(inc, text='Regno', font=('Arial', 12), width=10).grid(row=2, column=0, padx=20, pady=30)
```

```
b1 = Entry(inc, width=100, textvariable=regno).grid(row=2, column=1, padx=20, pady=30)
```

```
c = Label(inc, text="Email Id.", font=('Arial', 12), width=10).grid(row=2, column=0, padx=20, pady=30)
```

```
c1 = Entry(inc, width=100, textvariable=mail).grid(row=3, column=1, padx=20, pady=20, ipady=3)
```

```
d = Label(inc, text="Phone no", font=('Arial', 12), width=10).grid(row=4, column=0, padx=20, pady=30)
```

```
d1 = Entry(inc, width=100, textvariable=phone).grid(row=4, column=1, padx=20, pady=30, ipady=3)
```

```
e = Label(inc, text="10th CGPA", font=('Arial', 12), width=10).grid(row=5, column=0, padx=20, pady=30)
```

```
e1 = Entry(inc, width=100, textvariable=tencgpa).grid(row=5, column=1, padx=20, pady=20, ipady=3)
```

```
f = Label(inc, text="12th CGPA", font=('Arial', 12), width=10).grid(row=6, column=0, padx=20, pady=30)
```

```
f1 = Entry(inc, width=100, textvariable=twncgpa).grid(row=6, column=1, padx=20, pady=30, ipady=3)
```

```
h1 = Text(inc, width=50, height=5)
```

```
h1.grid(row=7, column=1, padx=0, pady=30)
```

```
tk = Button(inc, text='Submit', command=run).grid(row=8, column=1, padx=10, ipadx=5, ipady=5)
```

```
inc.mainloop()
```

```
def display():
```

```
det = Toplevel()
```

```
det.title("DISPLAY window")
```

Expt. No. _____

Page No. 43

Expt. Name. _____

Date: _____

```
det.geometry('500x600')
det['bg'] = '#009'
label(det, text="Database", font=('Arial', 20)).grid(row=0, column=1, padx=20)
tb = Text(det, width=50, height=30, background="#ffff")
tb.grid(row=1, column=1, padx=10, pady=20)
try:
    conn_str = (
        r'DRIVER={Microsoft Access Driver (*.mdb, *.accdb)};'
        r'DBQ=D:\COLLEGE\SEM6\ML and DA(LAB)\MSaccess\Database.accdb'
    )
    my_db = mdb.connect(conn_str)
    db = my_db.cursor()
    for i in xyz:
        tb.insert(INSERT, str(i) + '\n')
    my_db.commit()
    my_db.close()
except:
    tb.insert(INSERT, "DISPLAY FAILED\n")
    try:
        my_db.close()
    except:
        pass
det.mainloop()

def delete():
    det = Toplevel()
    regne = IntVar()
    def run():
        try:
```

Expt. No. _____

Page No. 44

Expt. Name. _____

Date : _____

conn_str = (

r'DRIVER = Microsoft Access Driver (*.mdb, *.accdb);'
r'DBQ = D:\COLLEGE\SEM 6\m2 and DB\ Database.accdb';
)

db.execute ("DELETE FROM Details where Regno = ? ", (regno.get(), F))

tb.insert (INSERT, "Deleted successfully");

except:

tb.insert (INSERT, "DELETION FAILED")

det.title ("Update Window")

det['bg'] = "#DAA"

a = Label (det, text = "Reg No", font = ('Arial', 12), width=10).grid (row=1, column=0, padx=20)

a1 = Entry (det, width=30, textvariable=regno).grid (row=1, column=1, padx=20, pady=20)

tb = Text (det, width=30, height=8, background="#fff").grid (row=9, column=1, padx=10)

det.mainloop()

label tit = Label (text = "Reg options", font=20)

label tit.pack (side = TOP, padx=30)

Result:

The above program is executed and the output is verified.

Sample I/O:

Input:

cat > /home /cloudera/Procen file.txt

Hey Cassie
Hey Anu
Hey Nate
Hey Masy
Hey Anu
Hi Nate

Output:

hdfs dfs -cat /out/part-r-000

Anu 2
Cassie 1
Hey 1
Hi 1
Maddy 1
Nate 2

Ques:

To write a word count Java program in ~~hadoop~~ hadoop to implement the map reduce concept.

Program:

```
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
```

Public class WordCount {

```
    public static class TokenizerMapper extends
```

```
        Mapper<Object, Text, Text, IntWritable> {
```

```
    private final static IntWritable one = new IntWritable(1);
```

```
    private Text word = new Text();
```

```
    public void map(Object key, Text values, Context context)
```

```
        throws IOException, InterruptedException {
```

```
        StringTokenizer itr = new StringTokenizer(values.toString());
```

```
        while (itr.hasMoreTokens()) {
```

```
            word.set(itr.nextToken());
```

```
word.set(itr.nextToken());
```

```
context.write(word, one);
```

{

}

```
}
```

```
public static class IntSumReducer extends  
Reducer<Text, IntWritable, Text, IntWritable> {  
private IntWritable result = new IntWritable();  
public void reduce(Text key, Iterable<IntWritable>  
values, Context context) throws IOException, InterruptedException {
```

```
int sum = 0;
```

```
for (IntWritable val: values) {  
sum += val.get();
```

{

```
result.set(sum);
```

```
context.write(key, result)
```

{

{

```
Public static void main(String[] args) throws Exception {
```

```
Configuration conf = new Configuration();
```

```
Job job = job.getInstance(conf, "WordCount");
```

```
job.setJarByClass(WordCount.class);
```

```
job.setMapperClass(TokenizerMapper.class);
```

```
job.setCombinerClass(IntSumReducer.class);
```

```
job.setReducerClass(IntSumReducer.class);
```

```
job.setOutputKeyClass(Text.class);
```

```
job.setOutputValueClass(IntWritable.class);
```

Expt. No. _____

Expt. Name. _____

Page No. 44

Date : _____

FileInputFormat. addInputPath(job, new path (args [0]));

FileOutputFormat. setOutputPath(job, new path (args [1]))

System. exit (job, wait for completion (true) : 0; 1);

3

Result:

The program was created successfully.