In[1]:
```
import pandas as pd
import csv
file = "C:\\Users\\AMLEC\\Desktop\\LOAN.csv"
datax = pd.read_csv(file)
display(datax)
```

	name	age	salary	kids	loan
0	ravi	35	50000	3	yes
1	geetha	37	45000	1	no
2	raju	33	60000	2	yes
3	ronaldo	36	1000000	5	yes

In[2]:
```
import csv
file = "C:\\Users\\AMLEC\\Desktop\\LOAN.csv"
fd = csv.reader(open(file))
for line in fd:
    print(line)
```

```
['name', 'age', 'salary', 'kids', 'loan']
['ravi', '35', '50000', '3', 'yes']
['geetha', '37', '45000', '1', 'no']
['raju', '33', '60000', '2', 'yes']
['ronaldo', '36', '1000000', '5', 'yes']
```

2/10/22

Program - 7

Apply EM algorithm to cluster a set of data stored in a .CSV file. Use the same data set for clustering using k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering. You can add Java/Python ML library classes/API in the program.

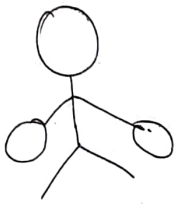
Light weight boxing

centroid

$w_1 = 55kg$

$h_1 = 160cm$

middle weight boxing

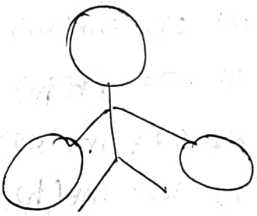


centroid

$w_2 = 70kg$

$h_2 = 170cm$

heavy weight boxing



centroid

$w_3 = 90kg$

$h_3 = 185cm$

New $w_2 = 75kg$

$h_2 = 172.5cm$

New $w_3 = 97.75kg$

$h_3 = 183.75cm$

New $w_1 = 52.5kg$

$h_1 = 157.5cm$

New $w_3 = 97.5kg$

$h_3 = 187.5cm$

New $w_2 = 68.75kg$

$h_2 = 168.125cm$

```
import sys
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

def kmeans(data):
    print("Enter light weight boxing centroid (w,h)")
    w1,h1 = input().split()
    print("Enter middle weight boxing centroid (w,h)")
    w2,h2 = input().split()
    print("Enter heavy weight boxing centroid (w,h)")
    w3,h3 = input().split()

    w1 = int(w1)
    h1 = int(h1)
    w2 = int(w2)
    h2 = int(h2)
    w3 = int(w3)
    h3 = int(h3)

    for line in data:
        x = int(line[0])
        y = int(line[1])
        d1 = np.sqrt((x-w1)**2 + (y-h1)**2)
        d2 = np.sqrt((x-w2)**2 + (y-h2)**2)
        d3 = np.sqrt((x-w3)**2 + (y-h3)**2)
        if(d1<d2 and d1<d3):
            print("boxer having w=", x, "h=", y, "belongs to cluster 1")
            w1 = (x+w1)/2
            h1 = (y+h1)/2
            plt.scatter(x,y, c="red")

        elif(d2<d1 and d2<d3):
            print("boxer having w=", x "h=", y, "belongs to cluster 2")
            w2 = (x+w2)/2
            h2 = (y+h2)/2
            plt.scatter(x,y, c="blue")
```

```
else :
    print ("boxer having w:", x, "h:", y, "belongs to cluster 3")
    plt. w3= (x+w3)/2
         h3= (y+h3)/2
         plt.scatter (x,y, c="green")

plt.scatter
plt.xlabel('weight')
plt.ylabel('height')
plt.title('VTU boxing 2021-2022')
plt.legend()
plt.show()

def main():
    file = "C:\\User\\Desktop\\Boxing.csv"
    datax = pd.read_csv(file)
    display (datax)
    data= []
    fd= csv.reader (open(file))
    for line in fd:
        data.append(line)
    kmean(data[1:])
main()
```

19/9/22

Program 6

weight	height	$d_1=\sqrt{(x-w_1)^2+(y-h_1)^2}$	$d_2=\sqrt{(x-w_2)^2+(y-h_1)^2}$	$d_3=\sqrt{(x-w_3)^2+(y-h_3)^2}$	comments
80	175	29.15	(11.18)	14.1	belongs to cluster 2 $w_2=(70+80)/2=75$ kg $h_2=(170+175)/2=172.5$ cm
50	155	(7.07)	26.08	50	belongs to cluster 1 $w_1=52.5$ kg $h_1=157.5$ cm
105	190	61.75	34.73	(15.81)	belongs to cluster 3 $w_3=97.5$ kg $h_3=187.5$ cm
70	170	21.50	(5.59)	32.59	belongs to cluster 2 $w_2=72.5$ kg. $h_2=171.25$ cm
65	165	14.57	(9.76)	39.52	belongs to cluster 2 $w_2=68.75$ kg $h_2=168.12$ cm
98	180	50.75	31.56	(8.76)	belongs to cluster 3 $w_3=97.75$ kg $h_3=183.75$ cm

Program 8

K - Nearest Neighbour Algorithm

```
import csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

def KNN (datax):
    print(" Enter your weight , height and chest size to buy right size")
    a,b,c = input().split()
    a=int(a)
    b=int(b)
    c=int(c)
    for line in datax:
        x=int (line[0])
        y=int (line[1])
        z=int(line[2])
        dist = np.sqrt((x-a)**2+(y-b)**2+(z-c)**2)
        line.append(dist)
    print(" How many nearest survey k = ")
    k=int(input())
    datax.sort(key=lambda i:i[4])
# print(" K sorted distance")
    scount=0
    mcount=0
    lcount=0
    xlcount=0
    for j in range(k):
        print(datax[j])
        if datax[j][3]=='S':
            scount+=1
        if datax[j][3]=='M':
            mcount+=1
```

```
        if datax[j][3]=='L':
            lcount+=1;
        if datax[j][3]=='XL':
            xlcount+=1
    if(scount>mcount and scount>lcount and scount>xlcount):
        print("---- GO SMALL SIZE ----")
    elif(mcount>scount and mcount>lcount and mcount>xlcount):
        print("---- GO MEDIUM SIZE ----")
    elif(lcount>scount and lcount>mcount and lcount>xlcount):
        print("---- GO LARGE SIZE ----")
    else:
        print("---- GO XL SIZE ----")

def main():
    file=r"C:\Users\AMLEC\Desktop\T-shirt1.csv"
    data=pd.read_csv(file)
    display(data)
    fd=csv.reader(open(file))
    datax=[]
    for line in fd:
        datax.append(line)
    KNN(datax[1:])

main()
```

Output :

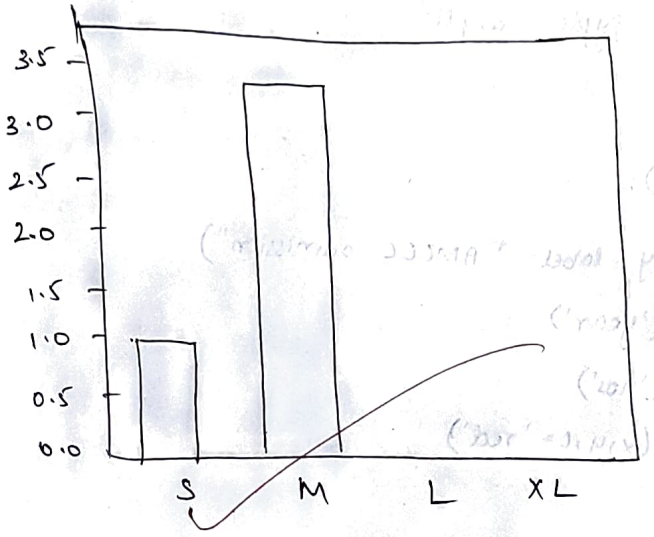
```
Enter your weight, height and chest size to buy right size:
60 176 38
How many nearest survey K=
4
K shortest distance
```

['55', '170', '45', 'S', 10.488...]

['67', '171', '44', 'M', 10.488...]

['66', '168', '46', 'M', 12.80...]

['67', '172', '49', 'M', 13.63...]



----- GO FOR MEDIUM SIZE -----

22/9/22

Program 9

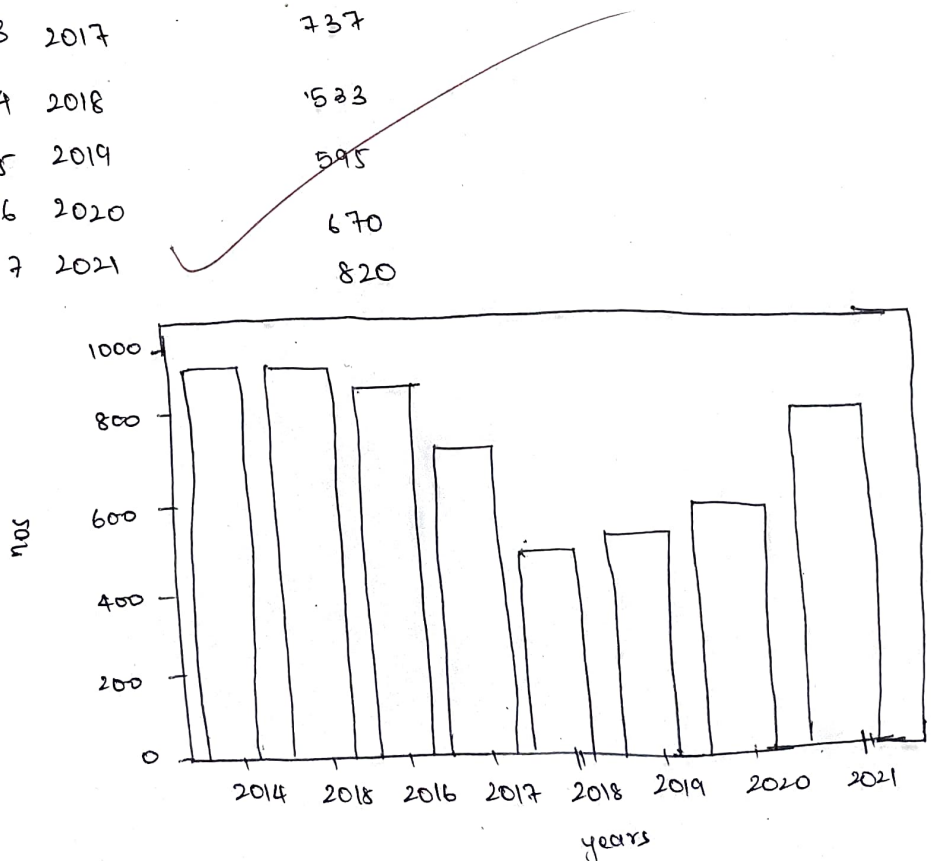
Linear Regression

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import csv

def regline(x,y):
    plt.bar(x,y, label="AMCEC admission")
    plt.xlabel('year')
    plt.ylabel('no')
    plt.scatter(x,y,c='red')
    plt.show()
    mx = np.mean(x)
    my = np.mean(y)
    n = len(x)
    up = 0
    M = 0
    dw = 0
    for i in range(n):
        up += (x[i]-mx)*(y[i]-my)
        dw += (x[i]-mx)**2

    M = up/dw
    c = my-(M*mx)
    print("linear regression slope =", M)
    print("linear regression constant =", c)
    max_x = np.max(x)+1
    min_x = np.min(x)-1
    x1 = 0
    x1 = np.linspace(min_x, max_x, 6)
    y1 = M*1+c
    plt.plot(x1,y1, color='blue')
```

```
plt.show()
print("Enter which year admission prediction")
year= int(input())
adm= m*year+c
print("Predicted admission=", adm)

def main():
    file= r"C:\Users\AMC college\Desktop\DATA SET\AMCEC.csv"
    data= pd.read_csv(file)
    display(data)
    x= data['year'].values
    y= data['nos'].values
    regline(x,y)

main()
```

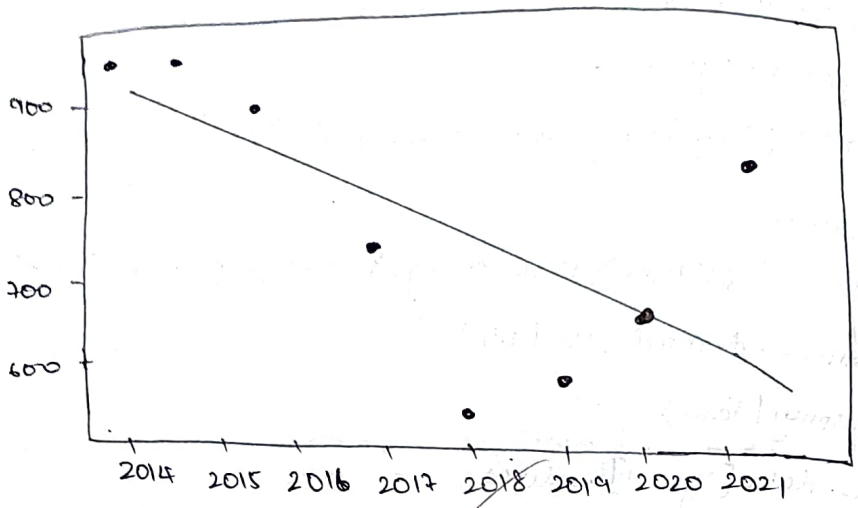
Output:

	year	nos
0	2014	955
1	2015	957
2	2016	914
3	2017	737
4	2018	533
5	2019	595
6	2020	670
7	2021	820



linear regression slope = -48.15476190476190 5

linear regression constant = 85819.85714285714



Enter which year admission predection

~~2020~~ 2019

Predicted admission = 709.3928571428551

27/9/22.

Problem

8/10/22

Program 6

```
import numpy as np
import pandas as pd
import csv
import matplotlib.pyplot as plt

def Bayes(DATA, x, col, yescount, nocount):
    xyes = 0
    xno = 0
    for line in DATA:
        if line[col] == x:
            if line[-1] == 'yes':
                xyes += 1
            else:
                xno += 1
    pxyes = xyes/yescount
    pxno = xno/nocount
    return pxyes, pxno

def main():
    file = r"C:\Users\AMC College\Desktop\DATASET\car(2).csv"
    temp = pd.read_csv(file)
    display(temp)
    data = []
    fd = csv.reader(open(file))
    for line in fd:
        data.append(line)
    DATA = data[1:]
    n = len(DATA)
    yescount = 0
    nocount = 0
    for line in DATA:
        if line[-1] == 'yes':
            yescount += 1
```

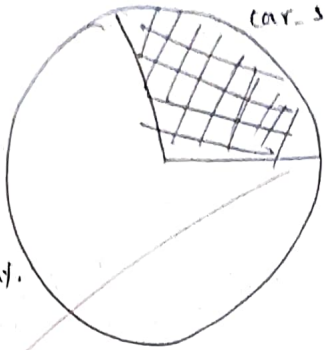
```
    else :
        y.nocount + = 1

pyes = yescount / n

pno = nocount / n

print (" enter your new car feature color, type, origin")

x, y, z = input().split()

pxyes , pxno = Bayes( DATA, x, 0, yescount, nocount)

pyyes, pyno = Bayes( DATA, y, 1, yescount, nocount)

pzyes, pzno = Bayes( DATA, z, 2, yescount, nocount)

resyes = pxyes * pyyes * pzyes * pyes

resno = pxno * pyno * pzno * pno

Percentageyes = (resyes / (resyes + resno)) * 100

Percentageno = (resno / (resno + resyes)) * 100

Pex = [Percentageyes, Percentageno]

label = ["car_stolen %.", "car_not_stolen %."]

plt.pie(Pex, labels = label)

plt.show()

print ("Percentages_yes =", Percentageyes, "percentage_no =",
                                    Percentageno)

main()
```

Output :

	color	type	origin	stolen
0	red	sports	domestic	yes
1	red	sports	imported	no
2	yellow	SUV	imported	yes
3	red	sports	domestic	yes
4	red	sports	imported	no
5	yellow	SUV	imported	yes
6	yellow	SUV	imported	yes
7	yellow	sports	imported	no
8	red	sports	domestic	no
9	red	sports	imported	no

Enter your new car feature color, type, origin

red sports domestic



Percentages-yes = 28.571428571428577

Percentage_no = 71.42857142857143.

Problems

18/10/22

Problem :

n = 9

n(yes) = 5 n(no) = 4 P(yes) = 5/9 P(no) = 4/9

$P(yes|(red, sports, domestic) = P(red/yes) * P(sports/yes) * P(domestic/yes) * P(yes)$

$P(no|(red, sports, domestic) = P(red/no) * P(sports/no) * P(domestic/no) * P(no)$

Attribute = red

n(red/yes) = 3 P(red/yes) = 3/5

n(red/no) = 2 P(red/no) = 2/4

Attribute = sports

n(sports/yes) = 3 P(sports/yes) = 3/5

n(sports/no) = 1 P(sports/no) = 1/4

Attribute : domestic

n(domestic/yes) = 3 $\quad$ P(domestic/yes) = 3/5

n(domestic/no) = 2 $\quad$ P(domestic/no) = 2/4

P(yes) = 5/9

P(no) = 4/9

$$P(\text{yes}/(\text{red}, \text{sports}, \text{domestic})) = P\left(\frac{\text{red}}{\text{yes}}\right) * P\left(\frac{\text{sports}}{\text{yes}}\right) * P\left(\frac{\text{domestic}}{\text{yes}}\right) * P(\text{yes})$$

$$= \frac{3}{5} * \frac{3}{5} * \frac{3}{5} * \frac{5}{9}$$

$$= \frac{3}{25}$$

$$x = 0.12$$

$$P(\text{no}/(\text{red}, \text{sports}, \text{domestic})) = P\left(\frac{\text{red}}{\text{no}}\right) * P\left(\frac{\text{sports}}{\text{no}}\right) * P\left(\frac{\text{domestic}}{\text{no}}\right) * P(\text{no})$$

$$= \frac{2}{4} \times \frac{1}{4} \times \frac{2}{4} \times \frac{4}{9}$$

$$= \frac{1}{36}$$

$$y = 0.027$$

$$\% \text{ car stolen} = \frac{x}{(x+y)} * 100 = \frac{0.12}{(0.12 + 0.027)} * 100$$

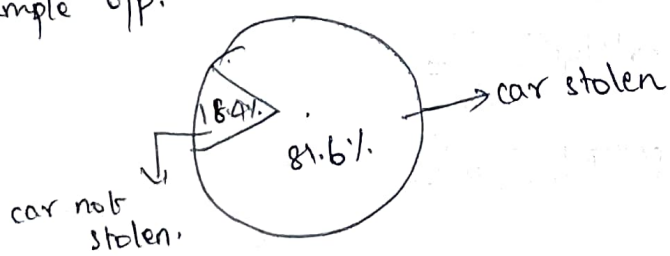
$$= 81.6\%.$$

$$\% \text{ car not stolen} = \frac{y}{(x+y)} * 100$$

$$= \frac{0.027}{(0.12 + 0.027)} * 100$$

$$= 18.4\%$$

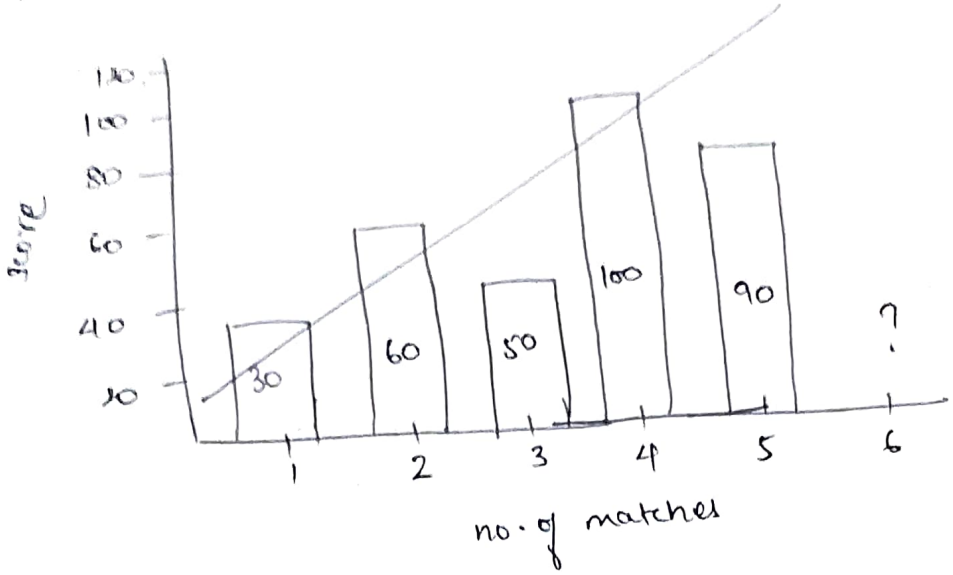
Sample o/p.



Program 9

Problem 1

Cricketer Preethi



X	Y
1	30
2	60
3	50
4	100
5	90
6	
15	340

$$M = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

$\bar{y} = m\bar{x} + c$

$c = \bar{y} - m\bar{x}$

$mx = \frac{\Sigma x}{n} = \frac{15}{5} = 3$ $\quad my = \frac{\Sigma y}{n} = \frac{340}{5} = 68.$

x	y	$(x_i - mx)$	$(y_i - my)$	$(x_i - mx)*(y_i - my)$	$(x_i - mx)^2$
1	30	(1-3) = 2	38	76	4
2	60	-1	-8	8	1
3	50	0	-18	0	0
4	100	1	42	42	1
5	90	2	22	44	4
				160	10

$M = \frac{170}{10} = 17$ $\quad c = my - m * mx \Rightarrow c = 17.$

$y = m * x + c \Rightarrow y = 17 * 6 + 17$

$y = 119$

$x = 6$

$y = ?$