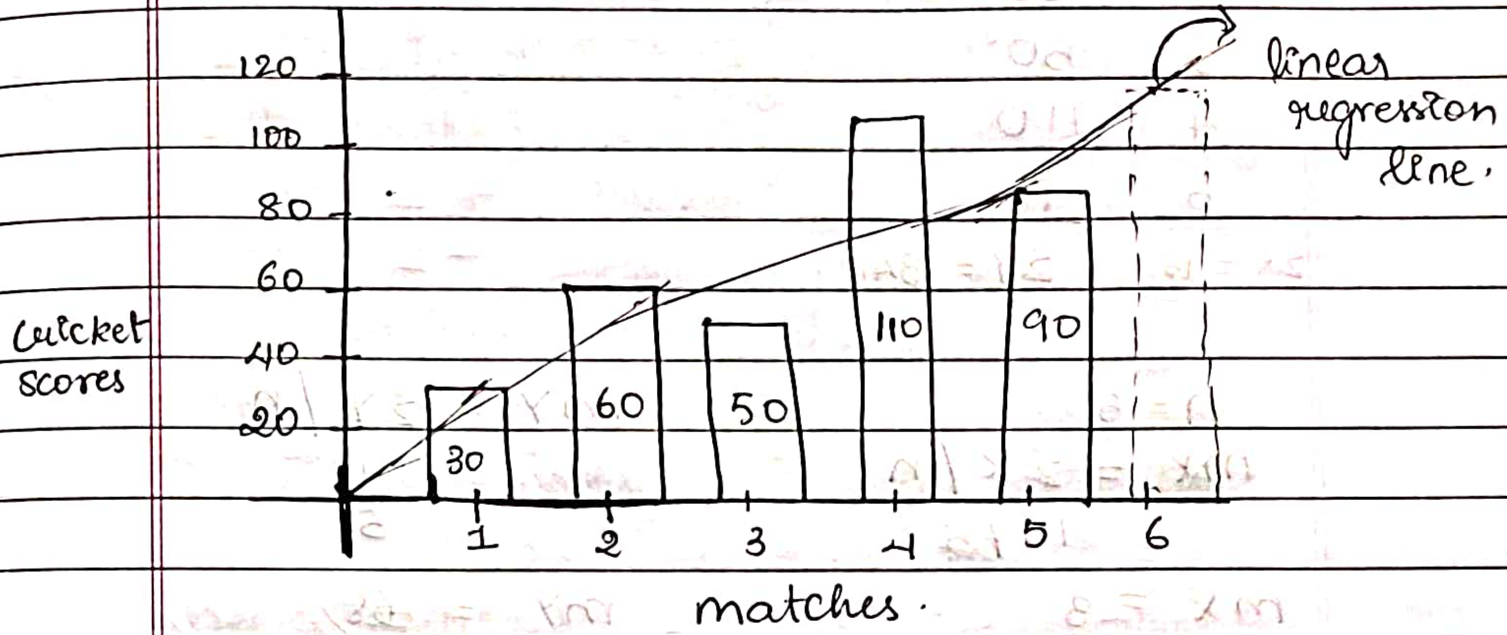
```
        plt.legend()
        plt.show()
def main():
      file = " C :\\user \\ Desktop \\ Boxing.csv"
      data x = pd. read csv (file)
      display (data x)
      data = []
    fd= csv. reader (open (file))
    for line  in fd :
         data . append (line)
    Kmean (data[1:])
main()
```

Linear Regression Algorithm



$$y = mx + c$$

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

formula for constant attribute :

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

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

X	Y
1	30
2	60
3	50
4	110
5	90
$\Sigma x = 15$	$\Sigma Y = 340$.

$n = 5$

$mx = \Sigma x / n$

$= 15/5$

$mx = 3$

$mY = \Sigma Y / n$

$= \frac{340}{5}$

$mY = 68$ //

X	y	(xi-mx)	(yi-my)	(xi-mx)* (yi-my)	(xi-mx)^2
1	30	-2	-38	76	4
2	60	-1	-8	8	1
3	50	0	-18	0	0
4	110	1	42	42	1
5	90	2	22	44	4
$\Sigma X = 15$	$\Sigma Y = 340$.			$\Sigma = 170$	$\Sigma = 10$

Origen point

$mY = M * mx + c$

$c = mY - M * mx$

$= 68 - 17 * 3$

$= 68 - 51$.

$c = 17$ //

$M = \frac{\Sigma(x - mx)(Y - mY)}{\Sigma(x - mx)^2}$

$= 170/10 = 17$ //.

$X = 6$ $Y = 2$

$y = M * x + c$

$= 17 * 6 + 17$

$y = 119$ //

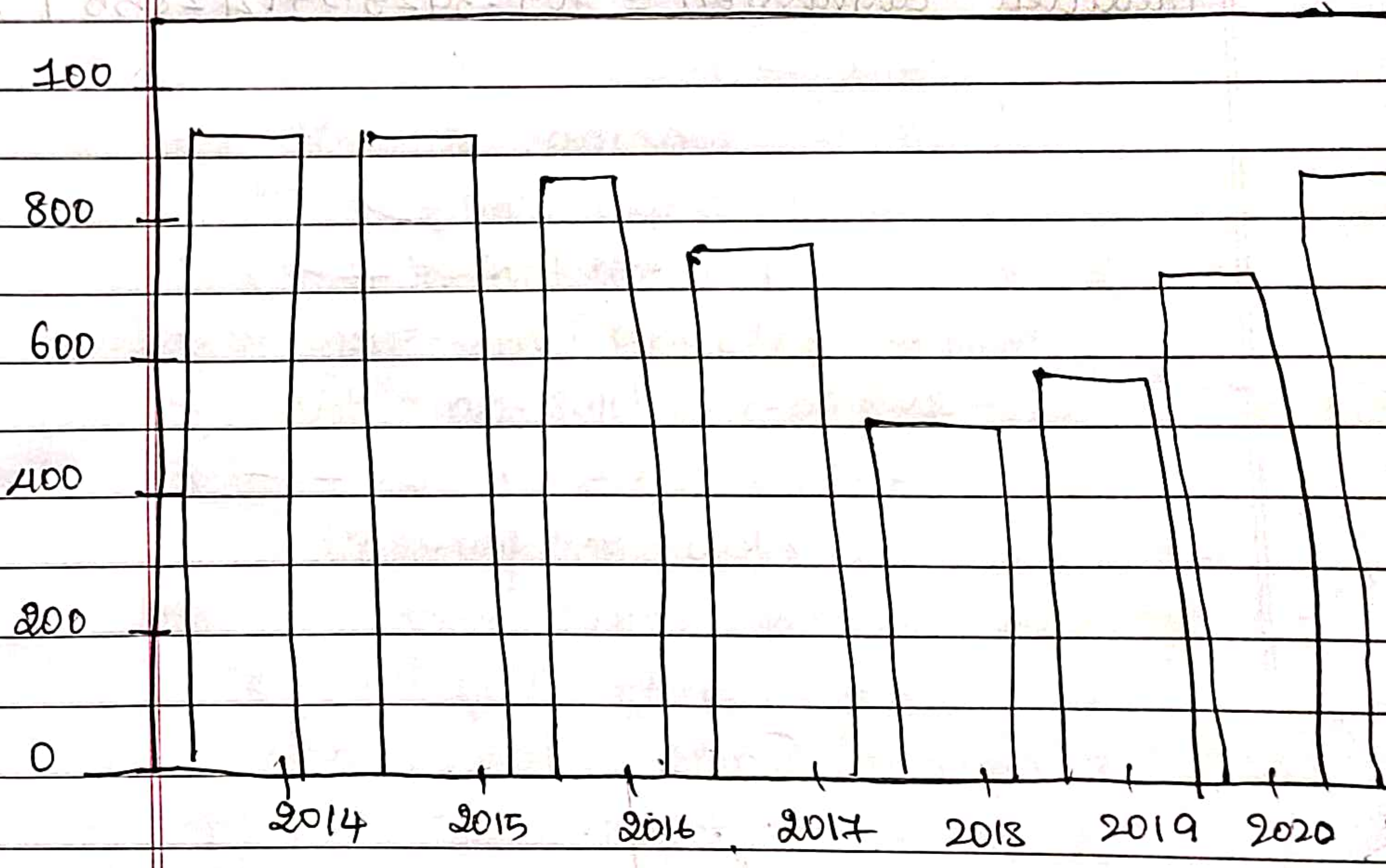
code :-

```
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 ('nos')
    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
    #print (max-x)
    X1 = 0
    X1 = np.linspace (min-x, max-x, 6)
    y1 = M * X1 + C
    plt.plot (X1, Y1, color = 'blue')
    print (x)
    plt.scatter (x, Y, c = 'r')
    plt.show()
    print ("Enter year prediction")
    year = int(input())
     adm = M* * 1 + c
     print (" Predicted admission =", adm)
def main():
file = r"C:\users\AMC college\Desktop\DATASE
                        \AMCEC.CV".
 data = pd.read_csv(file)
 display(data)
X = data['year'].values.
Y = data['nos'].values.
```

regline (x, y)
main()

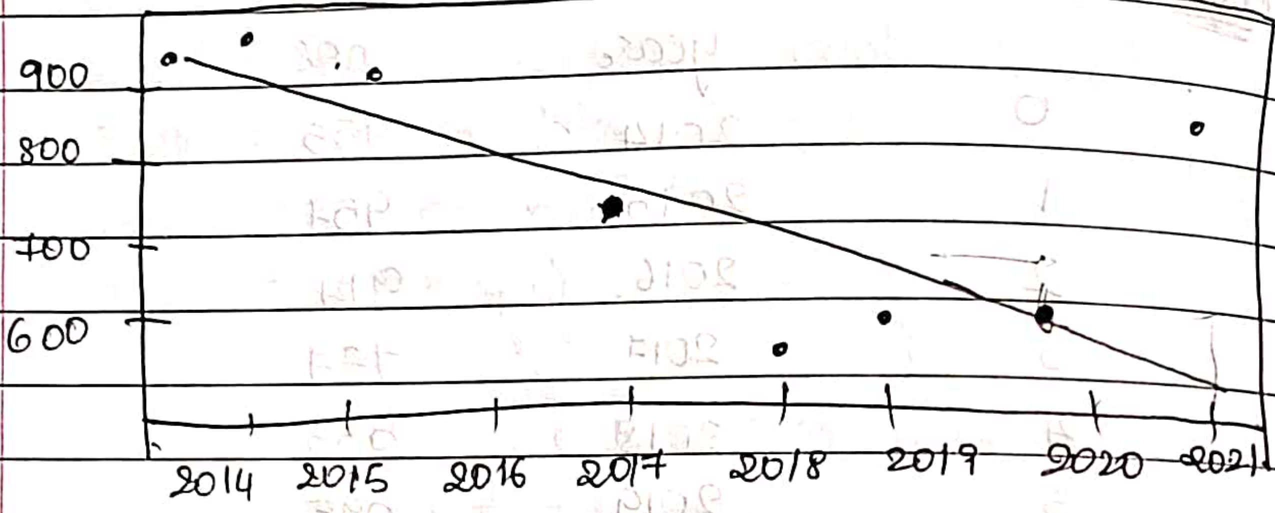
Output:

```
          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 = -42.154761904761905

Linear regression constant = 85814.85714285714



Enter which year admission prediction
2019

Predicted admission = 709.3928571428551.

PROGRAM - 8

K-Nearest Neighbour Algorithm.

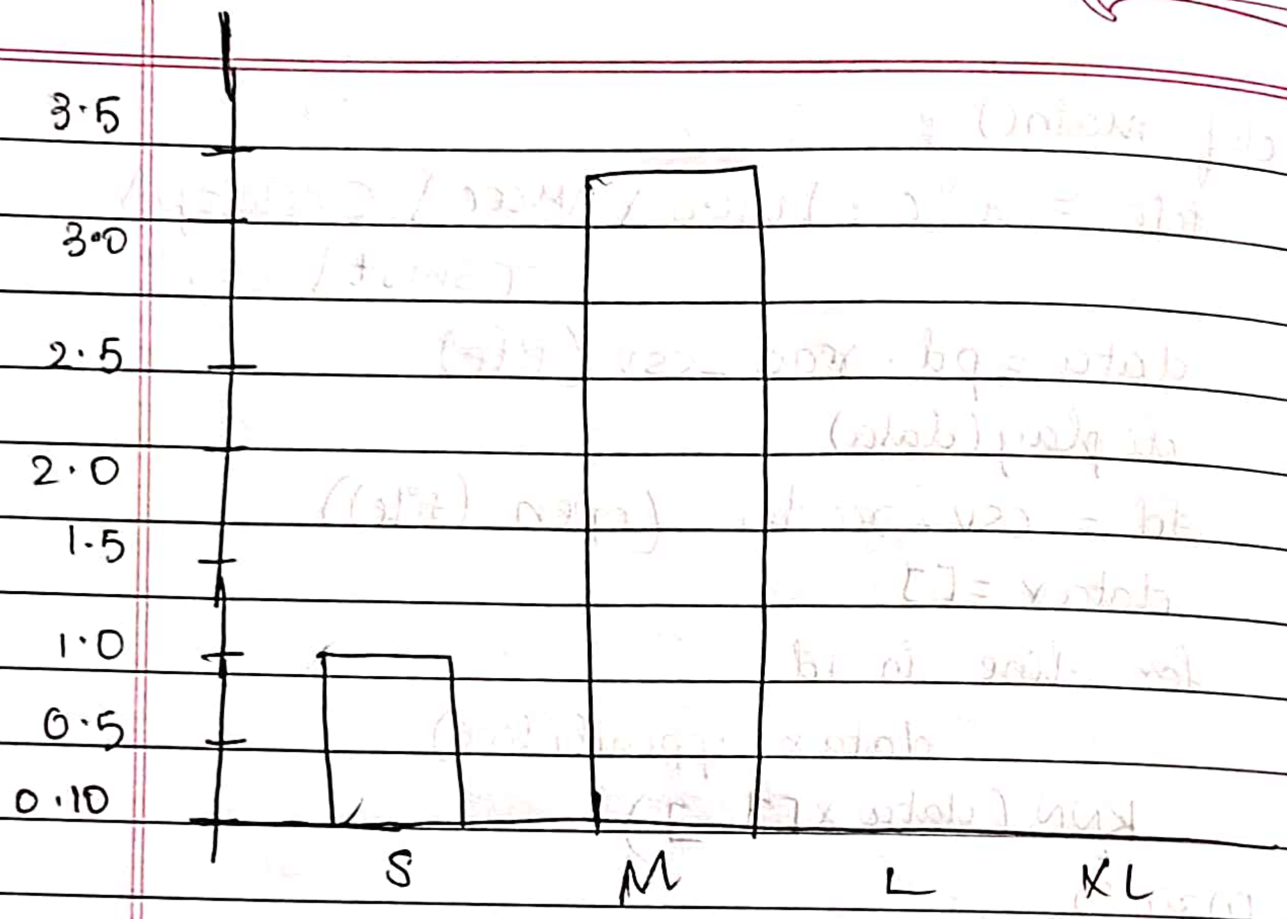
```
import csv
import numpy as np
import pandas as pd.
import matplot lib. pyplot as plt.
def KNN (data x):
print(" Enter your weight, height & chest
  size to buy right size")
 a, b, c = input(). split()
  a = int (a)
  b = int(b)
  c = int(c)
for line in data x:
        x = int( line [0])
        y = int( line [1])
        z = int ( line [2])
      dist = np. sqrt ( (x-a) **2 + (y-b) **2 +(z
             ** 2)
        line. append (dist)
print( "How many nearest survey k=")
  K = int (input())
   data x. sort (key = lambda i = i[4])
 #print ("k Sorted distance")
```

```
scount = 0
mcount =0
L count =0
xL count=0
for j in range (K):
    print (data x [j])
    if data x [j][3] == '3' :
        scount + =1
    if data x [j] [3] == 'M' :
        mcount + =1
    if data x [j][3] == 'L' :
        lcount + =1
    if datax [j][3] == 'XL' :
        XL Count+=1
if (scount >mcount and scount>Lcount and
        scount > xlcount ):
    print ("--- GO SMALL SIZE ---")
elif (mcount > scount and mcount > Lcount
    and mcount > xl count):
    print ("--- GO MEDIUM SIZE ---")
elif (lcount > scount and lcount > mcount
    and Lcount > xlcount):
    prent ("---GO LARGE SIZE ---")
else :
    print("---GO XL SIZE ---")
```

```
def main() :
    file = r "C : \users \ AMCEC \ Desktop \
                                T-shirt \ . 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 & chest size to
  buy right size :
60  176  38
How many nearest survey k :
 4
K shortest distance
['55', '170', '45', '3', 10.488--]
['67', '171', '44', 'M', 10.488--]
['66', '168', '46', 'M' 12.80--]
['67', '172', '49', 'M' 13.63--]
```



--- GO MEDIUM SIZE ---

Program 6

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(domestics /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.

Attributes = 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   p(domestic/yes) = 3/5

n(domestic/no) = 2   p(domestic/no) = 2/4

p(yes) = 5/9

p(no) = 4/9

$P(yes) (red, sports, domestics))$

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

$= 3/5 * 3/5 * 3/5 * 1/9 = 3/25$

$X = 0.12$

$P(no / (red, sports, domestic)))$

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

$= \frac{3}{5} * \frac{3}{5} * \frac{2}{4} * \frac{4}{9}$

$= \frac{1}{36}$

$y = 0.02\cancel{8}7$

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

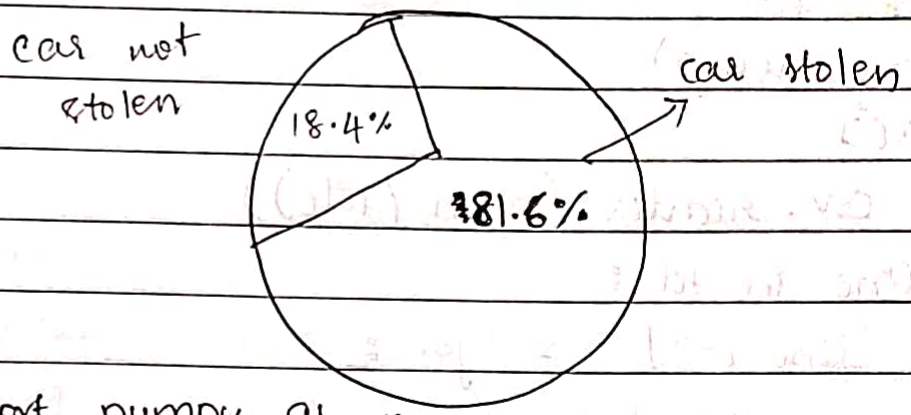
$= 81.6\%$

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

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

$= 18.4\%$

Sample o/p :



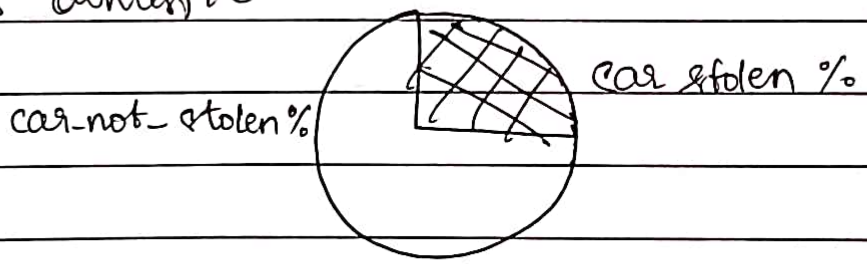
```
import numpy as np.
impost pandas as pd
 impost csv
import matplotlib . pyplot as plt
def Bayes ( DATA , X.COL , yes count , nocount):
      xyes = 0
      x no = 0
   for line in DATA :
         if line [COL] = = x :
         if line [-1] = 'yes' :
               X y + = 1
         else
               x no + = 1
      Px yes = xyes /yes count
      px no = xno / no count
      return pxyes , px no
 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 :
    if line [-1] = = 'yes' :
          yescount + = 1
    else :
          nocount + = 1
pyes = yescount /n
pno = nocount /n
print ("Enter your new car feature color, type,
                                         origen")
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 * pyyes * pzyes * pyes
resno = pxno * pyno * pyno * pzno * pno.
percentageyes = (resyes/ (resyes + resno)) *100
percentageno = (resno/(resno + resyes)) * 100
pex = [percentages_yes , Percentageno]
label = ["car_stolen % ", "car_not_stolen % "]
plt. pie (pex, label = lab el)
plt .show()
```

```
print ("Percentage_yes=", Percentageyes =,
        "percentage_no", percentageno).
main()
```

⇒ Output :

```
     color     type      origen       stolen
0    red       sports    domestic       yes
1    red       sports     imported      no
2    yellow    suv          ''          yes
3    red       sports     domestic      yes
4    red       sports     imported      no
5    yellow    suv          "           yes
6    yellow    suv          "           yes
7    yellow    sports     "              no
8    red         "        domestic       no
9    red         "        imported       no
Enter your new car feature color, type, origin
red sports domestic
```



```
percentage_yes = 28.571428571428577
percentage_no = 71.428571428571143.
```