In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
dataset=pd.read_csv(r"D:\Data Science with AI\4th-jan-2024\Social_Network_Ads.csv
```

In [3]:
```
dataset
```

Out[3]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0
...	...	...	...	...	...
395	15691863	Female	46	41000	1
396	15706071	Male	51	23000	1
397	15654296	Female	50	20000	1
398	15755018	Male	36	33000	0
399	15594041	Female	49	36000	1

400 rows × 5 columns

convert dataset into dependent and independent

In [4]:
```
x=dataset.iloc[:,[2,3]].values
y=dataset.iloc[:,-1].values
```

In [5]:
```
x
```

Out[5]:
```
array([[    19,  19000],
       [    35,  20000],
       [    26,  43000],
       [    27,  57000],
       [    19,  76000],
       [    27,  58000],
       [    27,  84000],
       [    32, 150000],
       [    25,  33000],
       [    35,  65000],
       [    26,  80000],
       [    26,  52000],
       [    20,  86000],
       [    32,  18000],
       [    18,  82000],
       [    29,  80000],
       [    47,  25000],
       [    45,  26000],
       [    46,  28000],
```

In [6]:
```
y
```

Out[6]:
```
array([0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1,
       0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0,
       1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0,
       1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1,
       0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1,
       1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1,
       0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0,
       1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1,
       0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1,
       1, 1, 0, 1], dtype=int64)
```

spliting the dataset into traning and testing set

In [9]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.20,random_state=0
```

In [10]:
```
x_train
```

Out[10]:
```
array([[    33,  31000],
       [    41,  72000],
       [    36,  33000],
       [    55, 125000],
       [    48, 131000],
       [    41,  71000],
       [    30,  62000],
       [    37,  72000],
       [    41,  63000],
       [    58,  47000],
       [    30, 116000],
       [    20,  49000],
       [    37,  74000],
       [    41,  59000],
       [    49,  89000],
       [    28,  79000],
       [    53,  82000],
       [    40,  57000],
       [    60,  34000],
       [    35, 108000],
       [    21,  72000],
       [    38,  71000],
       [    39, 106000],
       [    37,  57000],
       [    26,  72000],
       [    35,  23000],
       [    54, 108000],
       [    30,  17000],
       [    39, 134000],
       [    29,  43000],
       [    33,  43000],
       [    35,  38000],
       [    41,  45000],
       [    41,  72000],
       [    39, 134000],
       [    27, 137000],
       [    21,  16000],
       [    26,  32000],
       [    31,  66000],
       [    39,  73000],
       [    41,  79000],
       [    47,  50000],
       [    41,  30000],
       [    37,  93000],
       [    60,  46000],
       [    25,  22000],
       [    28,  37000],
       [    38,  55000],
       [    36,  54000],
       [    20,  36000],
       [    56, 104000],
       [    40,  57000],
       [    42, 108000],
       [    20,  23000],
       [    40,  65000],
       [    47,  20000],
       [    18,  86000],
```

```
       [    35,  79000],
       [    57,  33000],
       [    34,  72000],
       [    49,  39000],
       [    27,  31000],
       [    19,  70000],
       [    39,  79000],
       [    26,  81000],
       [    25,  80000],
       [    28,  85000],
       [    55,  39000],
       [    50,  88000],
       [    49,  88000],
       [    52, 150000],
       [    35,  65000],
       [    42,  54000],
       [    34,  43000],
       [    37,  52000],
       [    48,  30000],
       [    29,  43000],
       [    36,  52000],
       [    27,  54000],
       [    26, 118000]], dtype=int64)
```

In [11]:
```
y_train
```

Out[11]:
```
array([0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0,
       1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1,
       1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0,
       0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0], dtype=int64)
```

In [12]:
```
x_test
```

Out[12]:
```
array([[    30,  87000],
       [    38,  50000],
       [    35,  75000],
       [    30,  79000],
       [    35,  50000],
       [    27,  20000],
       [    31,  15000],
       [    36, 144000],
       [    18,  68000],
       [    47,  43000],
       [    30,  49000],
       [    28,  55000],
       [    37,  55000],
       [    39,  77000],
       [    20,  86000],
       [    32, 117000],
       [    37,  77000],
       [    19,  85000],
       [    55, 130000],
```

In [13]:
```
y_test
```

Out[13]:
```
array([0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1,
       0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1,
       0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1,
       1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0,
       0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1,
       0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0,
       1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1,
       0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0,
       0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1,
       1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0,
       0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0], dtype=int64)
```

feature scaling

In [16]:
```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
x_train=sc.fit_transform(x_train)
x_test=sc.transform(x_test)
```

In [18]:

```
print(x_train)
```

```
[[-0.41832001 -1.12664039]
 [ 0.34879978  0.18238564]
 [-0.13065009 -1.06278546]
 [ 1.69125941  1.87454123]
 [ 1.0200296   2.06610602]
 [ 0.34879978  0.15045817]
 [-0.70598993 -0.136889  ]
 [-0.03476012  0.18238564]
 [ 0.34879978 -0.10496154]
 [ 1.97892933 -0.61580096]
 [-0.70598993  1.58719406]
 [-1.66488967 -0.55194603]
 [-0.03476012  0.24624057]
 [ 0.34879978 -0.23267139]
 [ 1.11591957  0.72515253]
 [-0.89776988  0.40587789]
 [ 1.49947946  0.50166028]
 [ 0.25290981 -0.29652632]
 [ 2.17070928 -1.03085799]
 [-0.22654006  1.33177434]
 [-1.5689997   0.18238564]
 [ 0.06112986  0.15045817]
 [ 0.15701983  1.26791941]
 [-0.03476012 -0.29652632]
 [-1.08954983  0.18238564]
 [-0.22654006 -1.3820601 ]
 [ 1.59536944  1.33177434]
 [-0.70598993 -1.57362488]
 [ 0.15701983  2.16188841]
 [-0.80187991 -0.74351082]
 [-0.41832001 -0.74351082]
 [-0.22654006 -0.90314814]
 [ 0.34879978 -0.67965589]
 [ 0.34879978  0.18238564]
 [ 0.15701983  2.16188841]
 [-0.99365985  2.2576708 ]
 [-1.5689997  -1.60555235]
 [-1.08954983 -1.09471292]
 [-0.61009996 -0.00917915]
 [ 0.15701983  0.2143131 ]
 [ 0.34879978  0.40587789]
 [ 0.92413962 -0.52001857]
 [ 0.34879978 -1.15856785]
 [-0.03476012  0.85286238]
 [ 2.17070928 -0.64772843]
 [-1.1854398  -1.41398756]
 [-0.89776988 -0.9350756 ]
 [ 0.06112986 -0.36038125]
 [-0.13065009 -0.39230871]
 [-1.66488967 -0.96700307]
 [ 1.78714939  1.20406449]
 [ 0.25290981 -0.29652632]
 [ 0.44468975  1.33177434]
 [-1.66488967 -1.3820601 ]
 [ 0.25290981 -0.04110661]
 [ 0.92413962 -1.47784249]
 [-1.85666962  0.62937013]
```

```
 [-0.22654006  0.40587789]
 [ 1.88303936 -1.06278546]
 [-0.32243004  0.18238564]
 [ 1.11591957 -0.87122067]
 [-0.99365985 -1.12664039]
 [-1.76077964  0.11853071]
 [ 0.15701983  0.40587789]
 [-1.08954983  0.46973281]
 [-1.1854398   0.43780535]
 [-0.89776988  0.59744267]
 [ 1.69125941 -0.87122067]
 [ 1.21180954  0.69322506]
 [ 1.11591957  0.69322506]
 [ 1.40358949  2.67272783]
 [-0.22654006 -0.04110661]
 [ 0.44468975 -0.39230871]
 [-0.32243004 -0.74351082]
 [-0.03476012 -0.45616364]
 [ 1.0200296  -1.15856785]
 [-0.80187991 -0.74351082]
 [-0.13065009 -0.45616364]
 [-0.99365985 -0.39230871]
 [-1.08954983  1.65104898]]
```

In [19]:
```
print(x_test)
```

```
[[-0.70598993  0.6612976 ]
 [ 0.06112986 -0.52001857]
 [-0.22654006  0.27816803]
 [-0.70598993  0.40587789]
 [-0.22654006 -0.52001857]
 [-0.99365985 -1.47784249]
 [-0.61009996 -1.63747981]
 [-0.13065009  2.48116305]
 [-1.85666962  0.05467578]
 [ 0.92413962 -0.74351082]
 [-0.70598993 -0.55194603]
 [-0.89776988 -0.36038125]
 [-0.03476012 -0.36038125]
 [ 0.15701983  0.34202296]
 [-1.66488967  0.62937013]
 [-0.51420998  1.61912152]
 [-0.03476012  0.34202296]
 [-1.76077964  0.59744267]
 [ 1.69125941  2.03417855]
```

traning the knn model on traning set

In [27]:
```
from sklearn.neighbors import KNeighborsClassifier
classifier=KNeighborsClassifier()
classifier.fit(x_train,y_train)
```

Out[27]:
```
KNeighborsClassifier
KNeighborsClassifier()
```

predicting test set result

In [29]:
```
y_pred=classifier.predict(x_test)
```

In [30]:
```
y_pred
```

Out[30]:
```
array([0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1,
       0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1,
       0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1,
       1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0,
       0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1,
       0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0,
       1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1,
       0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1,
       0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1,
       1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0,
       1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1,
       1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1,
       0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0], dtype=int64)
```

making confusion matrix

In [44]:
```
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,y_pred)
print(cm)
```

```
[[185  16]
 [ 19 100]]
```

this is get model accuracy

In [45]:
```
from sklearn.metrics import accuracy_score
ac=accuracy_score(y_test,y_pred)
print(ac)
```

```
0.890625
```

this is get classification report

In [47]:
```
from sklearn.metrics import classification_report
cr=classification_report(y_test,y_pred)
print(cr)
```

```
              precision    recall  f1-score   support

           0       0.91      0.92      0.91       201
           1       0.86      0.84      0.85       119

    accuracy                           0.89       320
   macro avg       0.88      0.88      0.88       320
weighted avg       0.89      0.89      0.89       320
```

In [48]:
```
bias=classifier.score(x_train,y_train)
bias
```

Out[48]: 0.9625

In [50]:
```
variance=classifier.score(x_test,y_test)
variance
```

Out[50]: 0.890625

In [ ]: