PROJECT 2

NAME- ANURAG MISHRA SIC- 20BCED17

a) By using Logistic Regression Algorithm

Part A: Data Preprocessing

Step1 : importing the libraries

In [3]:

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

step2: import data set

In [4]:

```
dataset=pd.read_csv('Logistic Data.csv')
```

In [5]:

```
dataset
```

Out[5]:

	Age	Salary	Purchased Plot
0	22	22990	0
1	38	24200	0
2	29	52030	0
3	30	68970	0
4	22	91960	0
...	...	...	...
395	49	49610	1
396	54	27830	1
397	53	24200	1
398	39	39930	0
399	52	43560	1

400 rows × 3 columns

step3: to create feature matrix and dependent variable vector

In [6]:

```
a=dataset.iloc[: ,:-1].values
```

```
b=dataset.iloc[:,-1].values
```

In [7]:

```
a
```

Out[7]:

```
array([[    22,  22990],
       [    38,  24200],
       [    29,  52030],
       [    30,  68970],
       [    22,  91960],
       [    30,  70180],
       [    30, 101640],
       [    35, 181500],
       [    28,  39930],
       [    38,  78650],
       [    29,  96800],
       [    29,  62920],
       [    23, 104060],
       [    35,  21780],
       [    21,  99220],
       [    32,  96800],
       [    50,  30250],
       [    48,  31460],
       [    49,  33880],
       [    51,  35090],
       [    48,  26620],
       [    50,  59290],
       [    51,  49610],
       [    48,  26620],
       [    49,  27830],
       [    50,  24200],
       [    52,  33880],
       [    50,  36300],
       [    32,  52030],
       [    34,  21780],
       [    34,  89540],
       [    30, 165770],
       [    24,  19360],
       [    31,  53240],
       [    30, 108900],
       [    38,  32670],
       [    36,  33880],
       [    33,  59290],
       [    29,  87120],
       [    30,  37510],
       [    30,  20570],
       [    36,  61710],
       [    38, 130680],
       [    33,  18150],
       [    31, 101640],
       [    26,  24200],
       [    28,  95590],
       [    30,  65340],
       [    33, 163350],
       [    34, 107690],
       [    27,  38720],
       [    21,  53240],
       [    32, 100430],
       [    38,  27830],
       [    30,  70180],
       [    27,  66550],
       [    26,  58080],
       [    31,  95590],
       [    25,  21780],
       [    35, 141570],
       [    30,  24200],
       [    28, 105270],
       [    26,  79860],
       [    35, 145200],
```

```
[    62, 100430],
[    27,  70180],
[    27,  22990],
[    26,  99220],
[    25,  76230],
[    34,  82280],
[    28,  96800],
[    27,  32670],
[    23,  27830],
[    36, 136730],
[    35,  21780],
[    37, 135520],
[    21,  62920],
[    25,  32670],
[    31, 105270],
[    29,  20570],
[    33,  96800],
[    42,  50820],
[    23,  59290],
[    38, 106480],
[    33,  75020],
[    34, 142780],
[    27,  66550],
[    31, 102850],
[    29,  98010],
[    38,  60500],
[    25,  98010],
[    33, 140360],
[    29,  18150],
[    32,  33880],
[    32, 100430],
[    38,  53240],
[    38,  30250],
[    31, 148830],
[    38,  88330],
[    31,  44770],
[    30, 106480],
[    31,  71390],
[    35, 104060],
[    36, 180290],
[    22,  25410],
[    24,  87120],
[    29,  42350],
[    30, 107690],
[    29, 104060],
[    41,  96800],
[    42,  85910],
[    40,  85910],
[    41,  73810],
[    40,  66550],
[    45,  96800],
[    43,  68970],
[    38,  90750],
[    39,  62920],
[    43,  71390],
[    44,  71390],
[    39,  90750],
[    40,  87120],
[    43,  90750],
[    38,  64130],
[    44,  61710],
[    42,  73810],
[    45,  78650],
[    29,  38720],
[    33,  20570],
[    29, 101640],
[    34,  70180],
[    36,  37510],
[    33, 105270],
[    24,  82280],
[    31,  66550],
[    26,  76230],
```

```
[    23,  99220],
[    33, 129470],
[    31,  71390],
[    22,  30250],
[    22, 102850],
[    21,  82280],
[    38,  71390],
[    33, 107690],
[    37,  30250],
[    27, 107690],
[    30, 116160],
[    44,  36300],
[    32,  73810],
[    23,  89540],
[    29,  18150],
[    44,  54450],
[    34,  91960],
[    39,  60500],
[    43,  56870],
[    34,  18150],
[    49,  71390],
[    32,  90750],
[    29,  36300],
[    35, 163350],
[    35, 121000],
[    28, 108900],
[    40,  39930],
[    38,  45980],
[    36,  83490],
[    21, 104060],
[    25,  66550],
[    38,  85910],
[    32, 179080],
[    32,  56870],
[    24, 106480],
[    37, 139150],
[    29, 142780],
[    37,  52030],
[    37,  87120],
[    26,  33880],
[    38,  56870],
[    28,  26620],
[    27,  27830],
[    34,  41140],
[    29,  19360],
[    34,  85910],
[    35, 141570],
[    36,  52030],
[    36,  72600],
[    34,  79860],
[    23,  99220],
[    36,  49610],
[    38,  87120],
[    31,  38720],
[    27, 101640],
[    22,  31460],
[    32,  52030],
[    22,  84700],
[    31, 107690],
[    37,  52030],
[    33,  95590],
[    23,  43560],
[    29,  96800],
[    38,  26620],
[    38,  47190],
[    52,  89540],
[    42, 162140],
[    44,  85910],
[    61, 122210],
[    50,  56870],
[    58, 157300],
[    55, 137940],
```

```
[    43, 171820],
[    49,  26620],
[    51, 116160],
[    55, 181500],
[    62,  50820],
[    38,  70180],
[    50,  52030],
[    63, 130680],
[    52,  78650],
[    43,  94380],
[    49, 116160],
[    62, 173030],
[    44,  96800],
[    38, 110110],
[    40, 174240],
[    63, 123420],
[    38,  72600],
[    40,  64130],
[    39, 152460],
[    59, 160930],
[    43,  87120],
[    45,  96800],
[    38, 177870],
[    42,  50820],
[    43, 129470],
[    52, 104060],
[    41, 135520],
[    49,  95590],
[    43,  68970],
[    40,  96800],
[    49,  99220],
[    56, 173030],
[    45, 180290],
[    41,  71390],
[    53, 106480],
[    59, 125840],
[    44,  87120],
[    54, 176660],
[    38,  60500],
[    60, 147620],
[    44,  62920],
[    38, 117370],
[    47,  47190],
[    40,  62920],
[    51, 162140],
[    40, 176660],
[    53,  53240],
[    55, 108900],
[    44,  87120],
[    43,  68970],
[    61, 114950],
[    48, 158510],
[    38,  93170],
[    39, 174240],
[    58, 151250],
[    38,  87120],
[    51, 108900],
[    45, 130680],
[    43,  90750],
[    40,  89540],
[    50, 174240],
[    43,  73810],
[    46, 160930],
[    62,  91960],
[    63,  50820],
[    42, 128260],
[    60,  31460],
[    60,  89540],
[    41,  85910],
[    52, 106480],
[    55,  45980],
[    53,  43560],
```

```
[    62, 106480],
[    38,  73810],
[    40,  84700],
[    55,  25410],
[    51, 170610],
[    40, 112530],
[    40,  75020],
[    51, 166980],
[    44,  95590],
[    40,  94380],
[    42, 162140],
[    52, 107690],
[    58,  47190],
[    40,  93170],
[    38,  68970],
[    39,  76230],
[    45,  88330],
[    46, 135520],
[    48,  95590],
[    49, 141570],
[    61,  45980],
[    51,  89540],
[    40, 165770],
[    40,  95590],
[    43,  72600],
[    45,  65340],
[    54, 162140],
[    50, 136730],
[    39, 151250],
[    41,  60500],
[    45,  84700],
[    42, 116160],
[    41,  60500],
[    52, 170610],
[    42,  95590],
[    42,  90750],
[    57, 125840],
[    38,  66550],
[    48,  38720],
[    39,  72600],
[    55, 166980],
[    56,  99220],
[    44,  62920],
[    51,  36300],
[    51, 158510],
[    44,  72600],
[    44,  87120],
[    45,  90750],
[    39, 142780],
[    50, 129470],
[    41,  61710],
[    51, 143990],
[    45,  78650],
[    43,  78650],
[    60,  72600],
[    39,  65340],
[    61, 174240],
[    38,  95590],
[    41,  66550],
[    42, 147620],
[    56, 125840],
[    38,  90750],
[    41,  78650],
[    50,  61710],
[    50, 127050],
[    44,  76230],
[    56,  87120],
[    57, 130680],
[    42,  93170],
[    41,  73810],
[    41, 136730],
[    40,  90750],
```

```
       [    45, 108900],
       [    40,  68970],
       [    39, 119790],
       [    63,  41140],
       [    57,  84700],
       [    44,  87120],
       [    43,  85910],
       [    45,  65340],
       [    46, 156090],
       [    56,  41140],
       [    50,  60500],
       [    45,  95590],
       [    45, 125840],
       [    62,  35090],
       [    61,  56870],
       [    49, 106480],
       [    41,  85910],
       [    57,  31460],
       [    63,  55660],
       [    63, 100430],
       [    42,  88330],
       [    62, 157300],
       [    40,  96800],
       [    49,  38720],
       [    49,  89540],
       [    45,  64130],
       [    44, 105270],
       [    61,  27830],
       [    45,  77440],
       [    51,  39930],
       [    47, 168190],
       [    52,  33880],
       [    60,  39930],
       [    59,  72600],
       [    52,  47190],
       [    42,  85910],
       [    50,  41140],
       [    51,  42350],
       [    51,  39930],
       [    50,  27830],
       [    48,  54450],
       [    63,  50820],
       [    42,  71390],
       [    49,  49610],
       [    54,  27830],
       [    53,  24200],
       [    39,  39930],
       [    52,  43560]], dtype=int64)
```

In [8]:

```
b
```

Out[8]:

```
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)
```

step4: replace the missing data

In [9]:

```
from sklearn.impute import SimpleImputer
imputer=SimpleImputer(missing_values=np.nan,strategy='mean')
imputer.fit(a[:,:])
a[:,:]=imputer.transform(a[:,:])
```

In [10]:

```
a
```

Out[10]:

```
array([[    22,  22990],
       [    38,  24200],
       [    29,  52030],
       [    30,  68970],
       [    22,  91960],
       [    30,  70180],
       [    30, 101640],
       [    35, 181500],
       [    28,  39930],
       [    38,  78650],
       [    29,  96800],
       [    29,  62920],
       [    23, 104060],
       [    35,  21780],
       [    21,  99220],
       [    32,  96800],
       [    50,  30250],
       [    48,  31460],
       [    49,  33880],
       [    51,  35090],
       [    48,  26620],
       [    50,  59290],
       [    51,  49610],
       [    48,  26620],
       [    49,  27830],
       [    50,  24200],
       [    52,  33880],
       [    50,  36300],
       [    32,  52030],
       [    34,  21780],
       [    34,  89540],
       [    30, 165770],
       [    24,  19360],
       [    31,  53240],
       [    30, 108900],
       [    38,  32670],
       [    36,  33880],
       [    33,  59290],
       [    29,  87120],
       [    30,  37510],
       [    30,  20570],
       [    36,  61710],
       [    38, 130680],
       [    33,  18150],
       [    31, 101640],
       [    26,  24200],
       [    28,  95590],
       [    30,  65340],
       [    33, 163350],
       [    34, 107690],
       [    27,  38720],
       [    21,  53240],
       [    32, 100430],
```

```
[    32, 100430],
[    38,  27830],
[    30,  70180],
[    27,  66550],
[    26,  58080],
[    31,  95590],
[    25,  21780],
[    35, 141570],
[    30,  24200],
[    28, 105270],
[    26,  79860],
[    35, 145200],
[    62, 100430],
[    27,  70180],
[    27,  22990],
[    26,  99220],
[    25,  76230],
[    34,  82280],
[    28,  96800],
[    27,  32670],
[    23,  27830],
[    36, 136730],
[    35,  21780],
[    37, 135520],
[    21,  62920],
[    25,  32670],
[    31, 105270],
[    29,  20570],
[    33,  96800],
[    42,  50820],
[    23,  59290],
[    38, 106480],
[    33,  75020],
[    34, 142780],
[    27,  66550],
[    31, 102850],
[    29,  98010],
[    38,  60500],
[    25,  98010],
[    33, 140360],
[    29,  18150],
[    32,  33880],
[    32, 100430],
[    38,  53240],
[    38,  30250],
[    31, 148830],
[    38,  88330],
[    31,  44770],
[    30, 106480],
[    31,  71390],
[    35, 104060],
[    36, 180290],
[    22,  25410],
[    24,  87120],
[    29,  42350],
[    30, 107690],
[    29, 104060],
[    41,  96800],
[    42,  85910],
[    40,  85910],
[    41,  73810],
[    40,  66550],
[    45,  96800],
[    43,  68970],
[    38,  90750],
[    39,  62920],
[    43,  71390],
[    44,  71390],
[    39,  90750],
[    40,  87120],
[    43,  90750],
[    38,  64130],
[    44,  61710],
```

```
[    41,  81710],
[    42,  73810],
[    45,  78650],
[    29,  38720],
[    33,  20570],
[    29, 101640],
[    34,  70180],
[    36,  37510],
[    33, 105270],
[    24,  82280],
[    31,  66550],
[    26,  76230],
[    23,  99220],
[    33, 129470],
[    31,  71390],
[    22,  30250],
[    22, 102850],
[    21,  82280],
[    38,  71390],
[    33, 107690],
[    37,  30250],
[    27, 107690],
[    30, 116160],
[    44,  36300],
[    32,  73810],
[    23,  89540],
[    29,  18150],
[    44,  54450],
[    34,  91960],
[    39,  60500],
[    43,  56870],
[    34,  18150],
[    49,  71390],
[    32,  90750],
[    29,  36300],
[    35, 163350],
[    35, 121000],
[    28, 108900],
[    40,  39930],
[    38,  45980],
[    36,  83490],
[    21, 104060],
[    25,  66550],
[    38,  85910],
[    32, 179080],
[    32,  56870],
[    24, 106480],
[    37, 139150],
[    29, 142780],
[    37,  52030],
[    37,  87120],
[    26,  33880],
[    38,  56870],
[    28,  26620],
[    27,  27830],
[    34,  41140],
[    29,  19360],
[    34,  85910],
[    35, 141570],
[    36,  52030],
[    36,  72600],
[    34,  79860],
[    23,  99220],
[    36,  49610],
[    38,  87120],
[    31,  38720],
[    27, 101640],
[    22,  31460],
[    32,  52030],
[    22,  84700],
[    31, 107690],
[    37,  52030],
[    33,  95590],
```

```
[    55,  95590],
[    23,  43560],
[    29,  96800],
[    38,  26620],
[    38,  47190],
[    52,  89540],
[    42, 162140],
[    44,  85910],
[    61, 122210],
[    50,  56870],
[    58, 157300],
[    55, 137940],
[    43, 171820],
[    49,  26620],
[    51, 116160],
[    55, 181500],
[    62,  50820],
[    38,  70180],
[    50,  52030],
[    63, 130680],
[    52,  78650],
[    43,  94380],
[    49, 116160],
[    62, 173030],
[    44,  96800],
[    38, 110110],
[    40, 174240],
[    63, 123420],
[    38,  72600],
[    40,  64130],
[    39, 152460],
[    59, 160930],
[    43,  87120],
[    45,  96800],
[    38, 177870],
[    42,  50820],
[    43, 129470],
[    52, 104060],
[    41, 135520],
[    49,  95590],
[    43,  68970],
[    40,  96800],
[    49,  99220],
[    56, 173030],
[    45, 180290],
[    41,  71390],
[    53, 106480],
[    59, 125840],
[    44,  87120],
[    54, 176660],
[    38,  60500],
[    60, 147620],
[    44,  62920],
[    38, 117370],
[    47,  47190],
[    40,  62920],
[    51, 162140],
[    40, 176660],
[    53,  53240],
[    55, 108900],
[    44,  87120],
[    43,  68970],
[    61, 114950],
[    48, 158510],
[    38,  93170],
[    39, 174240],
[    58, 151250],
[    38,  87120],
[    51, 108900],
[    45, 130680],
[    43,  90750],
[    40,  89540],
[    50, 174240]
```

```
[    43,  73810],
[    46, 160930],
[    62,  91960],
[    63,  50820],
[    42, 128260],
[    60,  31460],
[    60,  89540],
[    41,  85910],
[    52, 106480],
[    55,  45980],
[    53,  43560],
[    62, 106480],
[    38,  73810],
[    40,  84700],
[    55,  25410],
[    51, 170610],
[    40, 112530],
[    40,  75020],
[    51, 166980],
[    44,  95590],
[    40,  94380],
[    42, 162140],
[    52, 107690],
[    58,  47190],
[    40,  93170],
[    38,  68970],
[    39,  76230],
[    45,  88330],
[    46, 135520],
[    48,  95590],
[    49, 141570],
[    61,  45980],
[    51,  89540],
[    40, 165770],
[    40,  95590],
[    43,  72600],
[    45,  65340],
[    54, 162140],
[    50, 136730],
[    39, 151250],
[    41,  60500],
[    45,  84700],
[    42, 116160],
[    41,  60500],
[    52, 170610],
[    42,  95590],
[    42,  90750],
[    57, 125840],
[    38,  66550],
[    48,  38720],
[    39,  72600],
[    55, 166980],
[    56,  99220],
[    44,  62920],
[    51,  36300],
[    51, 158510],
[    44,  72600],
[    44,  87120],
[    45,  90750],
[    39, 142780],
[    50, 129470],
[    41,  61710],
[    51, 143990],
[    45,  78650],
[    43,  78650],
[    60,  72600],
[    39,  65340],
[    61, 174240],
[    38,  95590],
[    41,  66550],
[    42, 147620],
[    56, 125840]
```

```
[    38,  90750],
[    41,  78650],
[    50,  61710],
[    50, 127050],
[    44,  76230],
[    56,  87120],
[    57, 130680],
[    42,  93170],
[    41,  73810],
[    41, 136730],
[    40,  90750],
[    45, 108900],
[    40,  68970],
[    39, 119790],
[    63,  41140],
[    57,  84700],
[    44,  87120],
[    43,  85910],
[    45,  65340],
[    46, 156090],
[    56,  41140],
[    50,  60500],
[    45,  95590],
[    45, 125840],
[    62,  35090],
[    61,  56870],
[    49, 106480],
[    41,  85910],
[    57,  31460],
[    63,  55660],
[    63, 100430],
[    42,  88330],
[    62, 157300],
[    40,  96800],
[    49,  38720],
[    49,  89540],
[    45,  64130],
[    44, 105270],
[    61,  27830],
[    45,  77440],
[    51,  39930],
[    47, 168190],
[    52,  33880],
[    60,  39930],
[    59,  72600],
[    52,  47190],
[    42,  85910],
[    50,  41140],
[    51,  42350],
[    51,  39930],
[    50,  27830],
[    48,  54450],
[    63,  50820],
[    42,  71390],
[    49,  49610],
[    54,  27830],
[    53,  24200],
[    39,  39930],
[    52,  43560]], dtype=int64)
```

In [11]:

```
b
```

Out[11]:

```
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, 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)
```

Step5: Encoding(not required)

step6 : spiliting of data set into training and testing set

In [12]:

```
from sklearn.model_selection import train_test_split
atrain,atest,btrain,btest=train_test_split(a,b,test_size=0.2,random_state=1)
```

In [13]:

```
atrain
```

Out[13]:

```
array([[    32,  33880],
       [    48,  26620],
       [    49, 141570],
       [    35,  21780],
       [    25,  98010],
       [    28, 105270],
       [    51,  39930],
       [    38,  70180],
       [    50,  27830],
       [    29,  38720],
       [    35, 163350],
       [    63,  41140],
       [    55,  25410],
       [    41,  66550],
       [    28, 108900],
       [    61,  45980],
       [    52, 106480],
       [    40,  94380],
       [    38,  93170],
       [    37,  52030],
       [    23,  59290],
       [    49, 106480],
       [    34,  41140],
       [    50,  36300],
       [    38,  60500],
       [    42, 116160],
       [    36, 136730],
       [    52, 104060],
       [    48,  95590],
       [    47,  47190],
       [    44,  71390],
       [    45,  64130],
       [    38,  88330],
       [    44,  87120],
       [    30, 116160],
       [    33, 140360],
       [    44,  62920],
       [    44,  62920],
       [    23,  99220],
       [    49,  49610],
```

```
[    49,  49610],
[    30,  37510],
[    38,  85910],
[    52,  33880],
[    38, 110110],
[    40,  90750],
[    35, 141570],
[    39,  90750],
[    23, 104060],
[    41,  60500],
[    52,  43560],
[    43,  78650],
[    40,  93170],
[    63,  55660],
[    51, 166980],
[    43,  85910],
[    39,  76230],
[    29,  98010],
[    36,  37510],
[    44,  61710],
[    49,  89540],
[    21,  99220],
[    40,  39930],
[    38,  64130],
[    31,  71390],
[    40,  85910],
[    21,  53240],
[    41,  73810],
[    38,  78650],
[    34,  79860],
[    50,  30250],
[    45, 125840],
[    22,  22990],
[    24,  87120],
[    31,  38720],
[    55, 181500],
[    56, 125840],
[    28,  96800],
[    29,  87120],
[    29,  18150],
[    43, 171820],
[    38,  45980],
[    60, 147620],
[    44, 105270],
[    27, 107690],
[    40,  62920],
[    38, 130680],
[    35, 121000],
[    38, 177870],
[    22,  31460],
[    49,  99220],
[    50, 127050],
[    48,  54450],
[    41,  61710],
[    40, 112530],
[    32,  52030],
[    44,  36300],
[    43,  72600],
[    30, 106480],
[    30, 108900],
[    42,  85910],
[    26,  33880],
[    54, 176660],
[    26,  76230],
[    40,  96800],
[    43,  56870],
[    51,  35090],
[    38,  71390],
[    29,  36300],
[    42, 128260],
[    31, 101640],
[    62,  91960],
[    43,  68970],
```

```
[    45,  68970],
[    38, 117370],
[    29, 104060],
[    44,  76230],
[    32,  56870],
[    29,  20570],
[    61,  27830],
[    33,  75020],
[    28,  39930],
[    62,  35090],
[    31,  44770],
[    42,  93170],
[    50, 129470],
[    55, 166980],
[    32,  52030],
[    30, 165770],
[    27,  66550],
[    43,  87120],
[    24,  19360],
[    60,  31460],
[    33, 163350],
[    51, 143990],
[    43,  68970],
[    38,  90750],
[    31,  53240],
[    40,  89540],
[    38,  32670],
[    40,  95590],
[    35, 145200],
[    32,  90750],
[    33,  20570],
[    28,  95590],
[    43, 129470],
[    27,  22990],
[    44,  87120],
[    41,  73810],
[    39, 174240],
[    45,  77440],
[    51,  39930],
[    51,  36300],
[    26,  24200],
[    58, 157300],
[    32, 179080],
[    45,  78650],
[    30,  65340],
[    40,  66550],
[    28,  26620],
[    62, 173030],
[    45,  65340],
[    30,  20570],
[    50,  59290],
[    31,  71390],
[    36,  83490],
[    34,  82280],
[    38,  27830],
[    38,  26620],
[    60,  39930],
[    33, 129470],
[    49,  27830],
[    44,  72600],
[    36,  49610],
[    36,  72600],
[    50,  56870],
[    31,  66550],
[    48,  38720],
[    38,  90750],
[    62,  50820],
[    50,  60500],
[    50,  61710],
[    38,  95590],
[    42,  50820],
[    43,  94380],
[    26,  58080],
```

```
[    26,  38080],
[    42,  90750],
[    38,  68970],
[    58,  47190],
[    34,  85910],
[    31, 148830],
[    45,  96800],
[    43,  71390],
[    51,  89540],
[    24, 106480],
[    56,  87120],
[    30,  70180],
[    38,  56870],
[    31, 107690],
[    29,  96800],
[    35, 141570],
[    45,  90750],
[    36, 180290],
[    44,  96800],
[    34,  70180],
[    42, 162140],
[    30,  24200],
[    32, 100430],
[    22,  84700],
[    22, 102850],
[    32,  73810],
[    42, 162140],
[    34,  91960],
[    44,  87120],
[    29,  96800],
[    43,  73810],
[    38,  30250],
[    51, 116160],
[    45, 180290],
[    31,  95590],
[    54, 162140],
[    36,  33880],
[    45,  65340],
[    48,  26620],
[    40,  68970],
[    37, 135520],
[    38,  47190],
[    25,  32670],
[    38,  87120],
[    42,  71390],
[    23,  89540],
[    49,  38720],
[    29,  52030],
[    32, 100430],
[    58, 151250],
[    40, 176660],
[    48, 158510],
[    36,  52030],
[    44,  54450],
[    45,  95590],
[    40, 165770],
[    27, 101640],
[    35,  21780],
[    59, 125840],
[    52,  47190],
[    31, 102850],
[    56, 173030],
[    33, 107690],
[    60,  72600],
[    43,  90750],
[    44,  95590],
[    23,  99220],
[    25,  66550],
[    38, 106480],
[    57,  84700],
[    34,  18150],
[    53,  43560],
[    45,  78650],
```

```
[    45,  78650],
[    37,  52030],
[    45, 130680],
[    57,  31460],
[    22,  25410],
[    39,  60500],
[    40,  84700],
[    39, 152460],
[    50,  24200],
[    33,  95590],
[    62, 100430],
[    32,  96800],
[    46, 135520],
[    41,  96800],
[    61, 174240],
[    39, 151250],
[    52,  33880],
[    21,  62920],
[    33,  18150],
[    62, 106480],
[    30,  68970],
[    41,  85910],
[    34, 107690],
[    50,  41140],
[    34,  89540],
[    40,  87120],
[    43,  68970],
[    62, 157300],
[    52,  78650],
[    51, 108900],
[    49,  26620],
[    57, 125840],
[    38,  24200],
[    52, 170610],
[    51,  49610],
[    38,  66550],
[    39,  72600],
[    35, 181500],
[    21,  82280],
[    27,  66550],
[    45, 108900],
[    41,  71390],
[    63, 130680],
[    25,  76230],
[    27,  38720],
[    49,  71390],
[    51, 162140],
[    44,  87120],
[    53,  53240],
[    41,  85910],
[    27,  27830],
[    38,  73810],
[    40,  96800],
[    27,  32670],
[    29, 101640],
[    37,  30250],
[    39,  65340],
[    24,  82280],
[    44,  85910],
[    63,  50820],
[    55, 108900],
[    23,  27830],
[    54,  27830],
[    49,  95590],
[    33,  59290]], dtype=int64)
```

step7 : Feature scaling

In [14]:

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

```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
atrain=sc.fit_transform(atrain)
atest=sc.fit_transform(atest)
```

In [15]:

```
atrain
```

Out[15]:

```
array([[-0.80330081, -1.19121795],
       [ 0.75697997, -1.36859801],
       [ 0.85449752,  1.43991958],
       [-0.51074816, -1.48685138],
       [-1.48592365,  0.37563923],
       [-1.19337101,  0.55301929],
       [ 1.04953262, -1.04340124],
       [-0.21819552, -0.30431766],
       [ 0.95201507, -1.33903467],
       [-1.09585346, -1.07296458],
       [-0.51074816,  1.97205975],
       [ 2.21974321, -1.0138379 ],
       [ 1.43960282, -1.39816136],
       [ 0.07435713, -0.39300769],
       [-1.19337101,  0.64170932],
       [ 2.02470811, -0.89558452],
       [ 1.14705017,  0.58258263],
       [-0.02316042,  0.2869492 ],
       [-0.21819552,  0.25738586],
       [-0.31571307, -0.74776781],
       [-1.68095875, -0.57038775],
       [ 0.85449752,  0.58258263],
       [-0.60826571, -1.0138379 ],
       [ 0.95201507, -1.13209127],
       [-0.21819552, -0.54082441],
       [ 0.17187468,  0.81908937],
       [-0.41323061,  1.32166621],
       [ 1.14705017,  0.52345594],
       [ 0.75697997,  0.31651254],
       [ 0.65946243, -0.86602118],
       [ 0.36690978, -0.27475432],
       [ 0.46442733, -0.45213438],
       [-0.21819552,  0.13913248],
       [ 0.36690978,  0.10956914],
       [-0.99833591,  0.81908937],
       [-0.70578326,  1.41035623],
       [ 0.36690978, -0.48169772],
       [ 0.36690978, -0.48169772],
       [-1.68095875,  0.40520257],
       [ 0.85449752, -0.80689449],
       [-0.99833591, -1.10252793],
       [-0.21819552,  0.0800058 ],
       [ 1.14705017, -1.19121795],
       [-0.21819552,  0.67127266],
       [-0.02316042,  0.19825917],
       [-0.51074816,  1.43991958],
       [-0.12067797,  0.19825917],
       [-1.68095875,  0.52345594],
       [ 0.07435713, -0.54082441],
       [ 1.14705017, -0.95471121],
       [ 0.26939223, -0.09737426],
       [-0.02316042,  0.25738586],
       [ 2.21974321, -0.65907778],
       [ 1.04953262,  2.06074978],
       [ 0.26939223,  0.0800058 ],
       [-0.12067797, -0.15650095],
       [-1.09585346,  0.37563923],
       [-0.41323061, -1.10252793],
       [ 0.36690978, -0.51126106],
       [ 0.85449752,  0.16869583],
       [-1.87599385,  0.40520257],
       [-0.02316042, -1.04340124],
```

```
[-0.21819552, -0.45213438],
[-0.90081836, -0.27475432],
[-0.02316042,  0.0800058 ],
[-1.87599385, -0.71820447],
[ 0.07435713, -0.21562763],
[-0.21819552, -0.09737426],
[-0.60826571, -0.06781092],
[ 0.95201507, -1.27990798],
[ 0.46442733,  1.05559612],
[-1.7784763 , -1.45728804],
[-1.5834412 ,  0.10956914],
[-0.90081836, -1.07296458],
[ 1.43960282,  2.4155099 ],
[ 1.53712037,  1.05559612],
[-1.19337101,  0.34607588],
[-1.09585346,  0.10956914],
[-1.09585346, -1.57554141],
[ 0.26939223,  2.17900315],
[-0.21819552, -0.89558452],
[ 1.92719056,  1.58773629],
[ 0.36690978,  0.55301929],
[-1.29088856,  0.61214597],
[-0.02316042, -0.48169772],
[-0.21819552,  1.17384949],
[-0.51074816,  0.93734275],
[-0.21819552,  2.32681987],
[-1.7784763 , -1.25034464],
[ 0.85449752,  0.40520257],
[ 0.95201507,  1.08515946],
[ 0.75697997, -0.68864112],
[ 0.07435713, -0.51126106],
[-0.02316042,  0.73039934],
[-0.80330081, -0.74776781],
[ 0.36690978, -1.13209127],
[ 0.26939223, -0.24519098],
[-0.99833591,  0.58258263],
[-0.99833591,  0.64170932],
[ 0.17187468,  0.0800058 ],
[-1.3884061 , -1.19121795],
[ 1.34208527,  2.29725653],
[-1.3884061 , -0.15650095],
[-0.02316042,  0.34607588],
[ 0.26939223, -0.62951444],
[ 1.04953262, -1.16165461],
[-0.21819552, -0.27475432],
[-1.09585346, -1.13209127],
[ 0.17187468,  1.1147228 ],
[-0.90081836,  0.46432926],
[ 2.12222566,  0.22782251],
[ 0.26939223, -0.33388101],
[-0.21819552,  0.84865272],
[-1.09585346,  0.52345594],
[ 0.36690978, -0.15650095],
[-0.80330081, -0.62951444],
[-1.09585346, -1.51641473],
[ 2.02470811, -1.33903467],
[-0.70578326, -0.18606429],
[-1.19337101, -1.04340124],
[ 2.12222566, -1.16165461],
[-0.90081836, -0.92514787],
[ 0.17187468,  0.25738586],
[ 0.95201507,  1.14428615],
[ 1.43960282,  2.06074978],
[-0.80330081, -0.74776781],
[-0.99833591,  2.03118644],
[-1.29088856, -0.39300769],
[ 0.26939223,  0.10956914],
[-1.5834412 , -1.54597807],
[ 1.92719056, -1.25034464],
[-0.70578326,  1.97205975],
[ 1.04953262,  1.49904626],
[ 0.26939223, -0.33388101],
```

```
[-0.21819552,  0.19825917],
[-0.90081836, -0.71820447],
[-0.02316042,  0.16869583],
[-0.21819552, -1.2207813 ],
[-0.02316042,  0.31651254],
[-0.51074816,  1.52860961],
[-0.80330081,  0.19825917],
[-0.70578326, -1.51641473],
[-1.19337101,  0.31651254],
[ 0.26939223,  1.14428615],
[-1.29088856, -1.45728804],
[ 0.36690978,  0.10956914],
[ 0.07435713, -0.21562763],
[-0.12067797,  2.23812984],
[ 0.46442733, -0.1269376 ],
[ 1.04953262, -1.04340124],
[ 1.04953262, -1.13209127],
[-1.3884061 , -1.4277247 ],
[ 1.73215547,  1.82424304],
[-0.80330081,  2.35638321],
[ 0.46442733, -0.09737426],
[-0.99833591, -0.42257103],
[-0.02316042, -0.39300769],
[-1.19337101, -1.36859801],
[ 2.12222566,  2.2085665 ],
[ 0.46442733, -0.42257103],
[-0.99833591, -1.51641473],
[ 0.95201507, -0.57038775],
[-0.90081836, -0.27475432],
[-0.41323061,  0.02087911],
[-0.60826571, -0.00868423],
[-0.21819552, -1.33903467],
[-0.21819552, -1.36859801],
[ 1.92719056, -1.04340124],
[-0.70578326,  1.14428615],
[ 0.85449752, -1.33903467],
[ 0.36690978, -0.24519098],
[-0.41323061, -0.80689449],
[-0.41323061, -0.24519098],
[ 0.95201507, -0.62951444],
[-0.90081836, -0.39300769],
[ 0.75697997, -1.07296458],
[-0.21819552,  0.19825917],
[ 2.12222566, -0.77733115],
[ 0.95201507, -0.54082441],
[ 0.95201507, -0.51126106],
[-0.21819552,  0.31651254],
[ 0.17187468, -0.77733115],
[ 0.26939223,  0.2869492 ],
[-1.3884061 , -0.59995109],
[ 0.17187468,  0.19825917],
[-0.21819552, -0.33388101],
[ 1.73215547, -0.86602118],
[-0.60826571,  0.0800058 ],
[-0.90081836,  1.61729964],
[ 0.46442733,  0.34607588],
[ 0.26939223, -0.27475432],
[ 1.04953262,  0.16869583],
[-1.5834412 ,  0.58258263],
[ 1.53712037,  0.10956914],
[-0.99833591, -0.30431766],
[-0.21819552, -0.62951444],
[-0.90081836,  0.61214597],
[-1.09585346,  0.34607588],
[-0.51074816,  1.43991958],
[ 0.46442733,  0.19825917],
[-0.41323061,  2.38594656],
[ 0.36690978,  0.34607588],
[-0.60826571, -0.30431766],
[ 0.17187468,  1.94249641],
[-0.99833591, -1.4277247 ],
[-0.80330081,  0.43476591],
```

```
[-1.7784763 ,  0.05044245],
[-1.7784763 ,  0.4938926 ],
[-0.80330081, -0.21562763],
[ 0.17187468,  1.94249641],
[-0.60826571,  0.22782251],
[ 0.36690978,  0.10956914],
[-1.09585346,  0.34607588],
[ 0.26939223, -0.21562763],
[-0.21819552, -1.27990798],
[ 1.04953262,  0.81908937],
[ 0.46442733,  2.38594656],
[-0.90081836,  0.31651254],
[ 1.34208527,  1.94249641],
[-0.41323061, -1.19121795],
[ 0.46442733, -0.42257103],
[ 0.75697997, -1.36859801],
[-0.02316042, -0.33388101],
[-0.31571307,  1.29210286],
[-0.21819552, -0.86602118],
[-1.48592365, -1.2207813 ],
[-0.21819552,  0.10956914],
[ 0.17187468, -0.27475432],
[-1.68095875,  0.16869583],
[ 0.85449752, -1.07296458],
[-1.09585346, -0.74776781],
[-0.80330081,  0.43476591],
[ 1.73215547,  1.67642632],
[-0.02316042,  2.29725653],
[ 0.75697997,  1.85380638],
[-0.41323061, -0.74776781],
[ 0.36690978, -0.68864112],
[ 0.46442733,  0.31651254],
[-0.02316042,  2.03118644],
[-1.29088856,  0.46432926],
[-0.51074816, -1.48685138],
[ 1.82967301,  1.05559612],
[ 1.14705017, -0.86602118],
[-0.90081836,  0.4938926 ],
[ 1.53712037,  2.2085665 ],
[-0.70578326,  0.61214597],
[ 1.92719056, -0.24519098],
[ 0.26939223,  0.19825917],
[ 0.36690978,  0.31651254],
[-1.68095875,  0.40520257],
[-1.48592365, -0.39300769],
[-0.21819552,  0.58258263],
[ 1.63463792,  0.05044245],
[-0.60826571, -1.57554141],
[ 1.24456772, -0.95471121],
[ 0.46442733, -0.09737426],
[-0.31571307, -0.74776781],
[ 0.46442733,  1.17384949],
[ 1.63463792, -1.25034464],
[-1.7784763 , -1.39816136],
[-0.12067797, -0.54082441],
[-0.02316042,  0.05044245],
[-0.12067797,  1.70598967],
[ 0.95201507, -1.4277247 ],
[-0.70578326,  0.31651254],
[ 2.12222566,  0.43476591],
[-0.80330081,  0.34607588],
[ 0.56194488,  1.29210286],
[ 0.07435713,  0.34607588],
[ 2.02470811,  2.23812984],
[-0.12067797,  1.67642632],
[ 1.14705017, -1.19121795],
[-1.87599385, -0.48169772],
[-0.70578326, -1.57554141],
[ 2.12222566,  0.58258263],
[-0.99833591, -0.33388101],
[ 0.07435713,  0.0800058 ],
[-0.60826571,  0.61214597],
```

```
       [ 0.95201507, -1.0138379 ],
       [-0.60826571,  0.16869583],
       [-0.02316042,  0.10956914],
       [ 0.26939223, -0.33388101],
       [ 2.12222566,  1.82424304],
       [ 1.14705017, -0.09737426],
       [ 1.04953262,  0.64170932],
       [ 0.85449752, -1.36859801],
       [ 1.63463792,  1.05559612],
       [-0.21819552, -1.4277247 ],
       [ 1.14705017,  2.14943981],
       [ 1.04953262, -0.80689449],
       [-0.21819552, -0.39300769],
       [-0.12067797, -0.24519098],
       [-0.51074816,  2.4155099 ],
       [-1.87599385, -0.00868423],
       [-1.29088856, -0.39300769],
       [ 0.46442733,  0.64170932],
       [ 0.07435713, -0.27475432],
       [ 2.21974321,  1.17384949],
       [-1.48592365, -0.15650095],
       [-1.29088856, -1.07296458],
       [ 0.85449752, -0.27475432],
       [ 1.04953262,  1.94249641],
       [ 0.36690978,  0.10956914],
       [ 1.24456772, -0.71820447],
       [ 0.07435713,  0.0800058 ],
       [-1.29088856, -1.33903467],
       [-0.21819552, -0.21562763],
       [-0.02316042,  0.34607588],
       [-1.29088856, -1.2207813 ],
       [-1.09585346,  0.46432926],
       [-0.31571307, -1.27990798],
       [-0.12067797, -0.42257103],
       [-1.5834412 , -0.00868423],
       [ 0.36690978,  0.0800058 ],
       [ 2.21974321, -0.77733115],
       [ 1.43960282,  0.64170932],
       [-1.68095875, -1.33903467],
       [ 1.34208527, -1.33903467],
       [ 0.85449752,  0.31651254],
       [-0.70578326, -0.57038775]])
```

Part B: build my first linear model

step 1: training the classification model

In [16]:

```
from sklearn.linear_model import LogisticRegression
LoR=LogisticRegression(random_state=0)
LoR.fit(atrain,btrain)
```

Out[16]:

```
LogisticRegression(random_state=0)
```

step 2: testing the linear model

In [17]:

```
bestimated=LoR.predict(atest)
print(np.concatenate((bestimated.reshape(len(bestimated),1),btest.reshape(len(btest),1))
,1))
```

```
[[0 0]
 [0 0]
```

[0 1]
[1 1]
[0 0]
[0 0]
[0 0]
[1 1]
[0 0]
[1 0]
[0 0]
[0 0]
[0 0]
[1 1]
[1 1]
[1 1]
[1 1]
[0 0]
[0 0]
[1 1]
[0 0]
[1 1]
[1 1]
[0 0]
[0 1]
[0 0]
[1 1]
[1 0]
[1 1]
[1 0]
[0 0]
[0 0]
[0 0]
[1 1]
[0 0]
[0 0]
[0 0]
[0 0]
[0 1]
[0 0]
[1 1]
[1 1]
[0 0]
[0 0]
[1 1]
[0 1]
[0 1]
[1 1]
[0 0]
[1 1]
[0 0]
[0 0]
[0 1]
[0 1]
[0 1]
[0 0]
[1 1]
[0 0]
[1 1]
[1 1]
[0 0]
[0 0]
[0 0]
[0 0]
[0 1]
[1 1]
[0 0]
[0 0]
[1 0]
[0 0]
[1 0]
[0 0]
[0 1]
[0 0]

```
[0 0]
[1 1]
[0 0]
[0 0]
[0 0]
[0 0]]
```

step C: performance matrix

In [19]:

```
from sklearn.metrics import confusion_matrix,accuracy_score,precision_score
cm=confusion_matrix(btest,bestimated)
print(cm)
print(accuracy_score(btest,bestimated))
print(precision_score(btest,bestimated))
```

```
[[43  5]
 [10 22]]
0.8125
0.8148148148148148
```

In [49]:

```
np.mean((True,True,False))
```

Out[49]:

```
0.6666666666666666
```

In [50]:

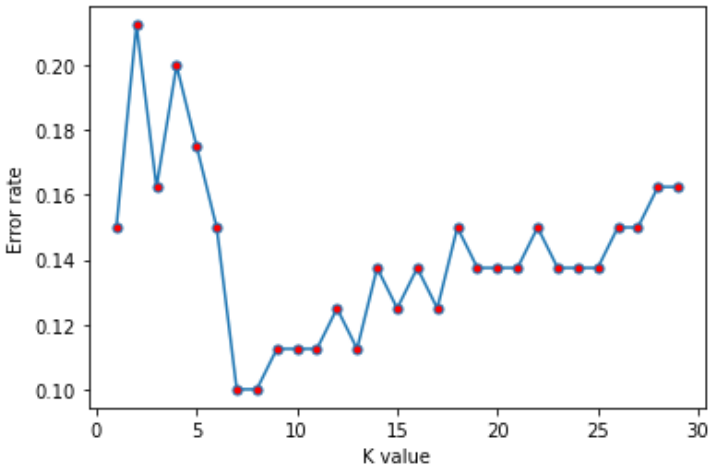
```
error_rate=[]
for i in range(1,30):
    KC=KNeighborsClassifier(n_neighbors=i)
    KC.fit(atrain,btrain)
    bpred_i=KC.predict(atest)
    error_rate.append(np.mean(bpred_i!=btest))
```

In [51]:

```
plt.plot(range(1,30),error_rate,marker='o',markerfacecolor='red',markersize=5)
plt.xlabel('K value')
plt.ylabel('Error rate')
```

Out[51]:

```
Text(0, 0.5, 'Error rate')
```



b) By using KNN Algorithm

Part A: Data Preprocessing

Step1 : importing the libraries

In [20]:

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

step2: import data set

In [22]:

```
dataset=pd.read_csv('Logistic Data.csv')
```

In [23]:

```
dataset
```

Out[23]:

	Age	Salary	Purchased Plot
0	22	22990	0
1	38	24200	0
2	29	52030	0
3	30	68970	0
4	22	91960	0
...	...	...	...
395	49	49610	1
396	54	27830	1
397	53	24200	1
398	39	39930	0
399	52	43560	1

400 rows × 3 columns

step3: to create feature matrix and dependent variable vector

In [24]:

```
a=dataset.iloc[:,:-1].values
b=dataset.iloc[:,-1].values
```

In [25]:

```
a
```

Out[25]:

```
array([[    22,  22990],
       [    38,  24200],
       [    29,  52030],
       [    30,  68970],
       [    22,  91960],
       [    30,  70180],
       [    30, 101640],
```

```
[    35, 181500],
[    28,  39930],
[    38,  78650],
[    29,  96800],
[    29,  62920],
[    23, 104060],
[    35,  21780],
[    21,  99220],
[    32,  96800],
[    50,  30250],
[    48,  31460],
[    49,  33880],
[    51,  35090],
[    48,  26620],
[    50,  59290],
[    51,  49610],
[    48,  26620],
[    49,  27830],
[    50,  24200],
[    52,  33880],
[    50,  36300],
[    32,  52030],
[    34,  21780],
[    34,  89540],
[    30, 165770],
[    24,  19360],
[    31,  53240],
[    30, 108900],
[    38,  32670],
[    36,  33880],
[    33,  59290],
[    29,  87120],
[    30,  37510],
[    30,  20570],
[    36,  61710],
[    38, 130680],
[    33,  18150],
[    31, 101640],
[    26,  24200],
[    28,  95590],
[    30,  65340],
[    33, 163350],
[    34, 107690],
[    27,  38720],
[    21,  53240],
[    32, 100430],
[    38,  27830],
[    30,  70180],
[    27,  66550],
[    26,  58080],
[    31,  95590],
[    25,  21780],
[    35, 141570],
[    30,  24200],
[    28, 105270],
[    26,  79860],
[    35, 145200],
[    62, 100430],
[    27,  70180],
[    27,  22990],
[    26,  99220],
[    25,  76230],
[    34,  82280],
[    28,  96800],
[    27,  32670],
[    23,  27830],
[    36, 136730],
[    35,  21780],
[    37, 135520],
[    21,  62920],
[    25,  32670],
[    31, 105270],
```

```
[    29,  20570],
[    33,  96800],
[    42,  50820],
[    23,  59290],
[    38, 106480],
[    33,  75020],
[    34, 142780],
[    27,  66550],
[    31, 102850],
[    29,  98010],
[    38,  60500],
[    25,  98010],
[    33, 140360],
[    29,  18150],
[    32,  33880],
[    32, 100430],
[    38,  53240],
[    38,  30250],
[    31, 148830],
[    38,  88330],
[    31,  44770],
[    30, 106480],
[    31,  71390],
[    35, 104060],
[    36, 180290],
[    22,  25410],
[    24,  87120],
[    29,  42350],
[    30, 107690],
[    29, 104060],
[    41,  96800],
[    42,  85910],
[    40,  85910],
[    41,  73810],
[    40,  66550],
[    45,  96800],
[    43,  68970],
[    38,  90750],
[    39,  62920],
[    43,  71390],
[    44,  71390],
[    39,  90750],
[    40,  87120],
[    43,  90750],
[    38,  64130],
[    44,  61710],
[    42,  73810],
[    45,  78650],
[    29,  38720],
[    33,  20570],
[    29, 101640],
[    34,  70180],
[    36,  37510],
[    33, 105270],
[    24,  82280],
[    31,  66550],
[    26,  76230],
[    23,  99220],
[    33, 129470],
[    31,  71390],
[    22,  30250],
[    22, 102850],
[    21,  82280],
[    38,  71390],
[    33, 107690],
[    37,  30250],
[    27, 107690],
[    30, 116160],
[    44,  36300],
[    32,  73810],
[    23,  89540],
[    29,  18150],
```

```
[    44,  54450],
[    34,  91960],
[    39,  60500],
[    43,  56870],
[    34,  18150],
[    49,  71390],
[    32,  90750],
[    29,  36300],
[    35, 163350],
[    35, 121000],
[    28, 108900],
[    40,  39930],
[    38,  45980],
[    36,  83490],
[    21, 104060],
[    25,  66550],
[    38,  85910],
[    32, 179080],
[    32,  56870],
[    24, 106480],
[    37, 139150],
[    29, 142780],
[    37,  52030],
[    37,  87120],
[    26,  33880],
[    38,  56870],
[    28,  26620],
[    27,  27830],
[    34,  41140],
[    29,  19360],
[    34,  85910],
[    35, 141570],
[    36,  52030],
[    36,  72600],
[    34,  79860],
[    23,  99220],
[    36,  49610],
[    38,  87120],
[    31,  38720],
[    27, 101640],
[    22,  31460],
[    32,  52030],
[    22,  84700],
[    31, 107690],
[    37,  52030],
[    33,  95590],
[    23,  43560],
[    29,  96800],
[    38,  26620],
[    38,  47190],
[    52,  89540],
[    42, 162140],
[    44,  85910],
[    61, 122210],
[    50,  56870],
[    58, 157300],
[    55, 137940],
[    43, 171820],
[    49,  26620],
[    51, 116160],
[    55, 181500],
[    62,  50820],
[    38,  70180],
[    50,  52030],
[    63, 130680],
[    52,  78650],
[    43,  94380],
[    49, 116160],
[    62, 173030],
[    44,  96800],
[    38, 110110],
[    40, 174240],
```

```
[    63, 123420],
[    38,  72600],
[    40,  64130],
[    39, 152460],
[    59, 160930],
[    43,  87120],
[    45,  96800],
[    38, 177870],
[    42,  50820],
[    43, 129470],
[    52, 104060],
[    41, 135520],
[    49,  95590],
[    43,  68970],
[    40,  96800],
[    49,  99220],
[    56, 173030],
[    45, 180290],
[    41,  71390],
[    53, 106480],
[    59, 125840],
[    44,  87120],
[    54, 176660],
[    38,  60500],
[    60, 147620],
[    44,  62920],
[    38, 117370],
[    47,  47190],
[    40,  62920],
[    51, 162140],
[    40, 176660],
[    53,  53240],
[    55, 108900],
[    44,  87120],
[    43,  68970],
[    61, 114950],
[    48, 158510],
[    38,  93170],
[    39, 174240],
[    58, 151250],
[    38,  87120],
[    51, 108900],
[    45, 130680],
[    43,  90750],
[    40,  89540],
[    50, 174240],
[    43,  73810],
[    46, 160930],
[    62,  91960],
[    63,  50820],
[    42, 128260],
[    60,  31460],
[    60,  89540],
[    41,  85910],
[    52, 106480],
[    55,  45980],
[    53,  43560],
[    62, 106480],
[    38,  73810],
[    40,  84700],
[    55,  25410],
[    51, 170610],
[    40, 112530],
[    40,  75020],
[    51, 166980],
[    44,  95590],
[    40,  94380],
[    42, 162140],
[    52, 107690],
[    58,  47190],
[    40,  93170],
[    38,  68970],
```

```
[    39,  76230],
[    45,  88330],
[    46, 135520],
[    48,  95590],
[    49, 141570],
[    61,  45980],
[    51,  89540],
[    40, 165770],
[    40,  95590],
[    43,  72600],
[    45,  65340],
[    54, 162140],
[    50, 136730],
[    39, 151250],
[    41,  60500],
[    45,  84700],
[    42, 116160],
[    41,  60500],
[    52, 170610],
[    42,  95590],
[    42,  90750],
[    57, 125840],
[    38,  66550],
[    48,  38720],
[    39,  72600],
[    55, 166980],
[    56,  99220],
[    44,  62920],
[    51,  36300],
[    51, 158510],
[    44,  72600],
[    44,  87120],
[    45,  90750],
[    39, 142780],
[    50, 129470],
[    41,  61710],
[    51, 143990],
[    45,  78650],
[    43,  78650],
[    60,  72600],
[    39,  65340],
[    61, 174240],
[    38,  95590],
[    41,  66550],
[    42, 147620],
[    56, 125840],
[    38,  90750],
[    41,  78650],
[    50,  61710],
[    50, 127050],
[    44,  76230],
[    56,  87120],
[    57, 130680],
[    42,  93170],
[    41,  73810],
[    41, 136730],
[    40,  90750],
[    45, 108900],
[    40,  68970],
[    39, 119790],
[    63,  41140],
[    57,  84700],
[    44,  87120],
[    43,  85910],
[    45,  65340],
[    46, 156090],
[    56,  41140],
[    50,  60500],
[    45,  95590],
[    45, 125840],
[    62,  35090],
[    61,  56870],
```

```
       [    49, 106480],
       [    41,  85910],
       [    57,  31460],
       [    63,  55660],
       [    63, 100430],
       [    42,  88330],
       [    62, 157300],
       [    40,  96800],
       [    49,  38720],
       [    49,  89540],
       [    45,  64130],
       [    44, 105270],
       [    61,  27830],
       [    45,  77440],
       [    51,  39930],
       [    47, 168190],
       [    52,  33880],
       [    60,  39930],
       [    59,  72600],
       [    52,  47190],
       [    42,  85910],
       [    50,  41140],
       [    51,  42350],
       [    51,  39930],
       [    50,  27830],
       [    48,  54450],
       [    63,  50820],
       [    42,  71390],
       [    49,  49610],
       [    54,  27830],
       [    53,  24200],
       [    39,  39930],
       [    52,  43560]], dtype=int64)
```

In [26]:

```
b
```

Out[26]:

```
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)
```

step4: replace the missing data

In [33]:

```
from sklearn.impute import SimpleImputer
imputer=SimpleImputer(missing_values=np.nan,strategy='mean')
imputer.fit(a[:,:])
a[:,:]=imputer.transform(a[:,:])
```

In [34]:

```
a
```

Out[34]:

```
array([[    22,  22990],
       [    38,  24200],
       [    29,  52030],
       [    30,  68970],
       [    22,  91960],
       [    30,  70180],
       [    30, 101640],
       [    35, 181500],
       [    28,  39930],
       [    38,  78650],
       [    29,  96800],
       [    29,  62920],
       [    23, 104060],
       [    35,  21780],
       [    21,  99220],
       [    32,  96800],
       [    50,  30250],
       [    48,  31460],
       [    49,  33880],
       [    51,  35090],
       [    48,  26620],
       [    50,  59290],
       [    51,  49610],
       [    48,  26620],
       [    49,  27830],
       [    50,  24200],
       [    52,  33880],
       [    50,  36300],
       [    32,  52030],
       [    34,  21780],
       [    34,  89540],
       [    30, 165770],
       [    24,  19360],
       [    31,  53240],
       [    30, 108900],
       [    38,  32670],
       [    36,  33880],
       [    33,  59290],
       [    29,  87120],
       [    30,  37510],
       [    30,  20570],
       [    36,  61710],
       [    38, 130680],
       [    33,  18150],
       [    31, 101640],
       [    26,  24200],
       [    28,  95590],
       [    30,  65340],
       [    33, 163350],
       [    34, 107690],
       [    27,  38720],
       [    21,  53240],
       [    32, 100430],
       [    38,  27830],
       [    30,  70180],
       [    27,  66550],
       [    26,  58080],
       [    31,  95590],
       [    25,  21780],
       [    35, 141570],
       [    30,  24200],
       [    28, 105270],
       [    26,  79860],
       [    35, 145200],
       [    62, 100430],
       [    27,  70180],
       [    27,  22990],
       [    26,  99220],
```

```
[    26,  99220],
[    25,  76230],
[    34,  82280],
[    28,  96800],
[    27,  32670],
[    23,  27830],
[    36, 136730],
[    35,  21780],
[    37, 135520],
[    21,  62920],
[    25,  32670],
[    31, 105270],
[    29,  20570],
[    33,  96800],
[    42,  50820],
[    23,  59290],
[    38, 106480],
[    33,  75020],
[    34, 142780],
[    27,  66550],
[    31, 102850],
[    29,  98010],
[    38,  60500],
[    25,  98010],
[    33, 140360],
[    29,  18150],
[    32,  33880],
[    32, 100430],
[    38,  53240],
[    38,  30250],
[    31, 148830],
[    38,  88330],
[    31,  44770],
[    30, 106480],
[    31,  71390],
[    35, 104060],
[    36, 180290],
[    22,  25410],
[    24,  87120],
[    29,  42350],
[    30, 107690],
[    29, 104060],
[    41,  96800],
[    42,  85910],
[    40,  85910],
[    41,  73810],
[    40,  66550],
[    45,  96800],
[    43,  68970],
[    38,  90750],
[    39,  62920],
[    43,  71390],
[    44,  71390],
[    39,  90750],
[    40,  87120],
[    43,  90750],
[    38,  64130],
[    44,  61710],
[    42,  73810],
[    45,  78650],
[    29,  38720],
[    33,  20570],
[    29, 101640],
[    34,  70180],
[    36,  37510],
[    33, 105270],
[    24,  82280],
[    31,  66550],
[    26,  76230],
[    23,  99220],
[    33, 129470],
[    31,  71390],
[    22,  30250]
```

```
[    22,  30250],
[    22, 102850],
[    21,  82280],
[    38,  71390],
[    33, 107690],
[    37,  30250],
[    27, 107690],
[    30, 116160],
[    44,  36300],
[    32,  73810],
[    23,  89540],
[    29,  18150],
[    44,  54450],
[    34,  91960],
[    39,  60500],
[    43,  56870],
[    34,  18150],
[    49,  71390],
[    32,  90750],
[    29,  36300],
[    35, 163350],
[    35, 121000],
[    28, 108900],
[    40,  39930],
[    38,  45980],
[    36,  83490],
[    21, 104060],
[    25,  66550],
[    38,  85910],
[    32, 179080],
[    32,  56870],
[    24, 106480],
[    37, 139150],
[    29, 142780],
[    37,  52030],
[    37,  87120],
[    26,  33880],
[    38,  56870],
[    28,  26620],
[    27,  27830],
[    34,  41140],
[    29,  19360],
[    34,  85910],
[    35, 141570],
[    36,  52030],
[    36,  72600],
[    34,  79860],
[    23,  99220],
[    36,  49610],
[    38,  87120],
[    31,  38720],
[    27, 101640],
[    22,  31460],
[    32,  52030],
[    22,  84700],
[    31, 107690],
[    37,  52030],
[    33,  95590],
[    23,  43560],
[    29,  96800],
[    38,  26620],
[    38,  47190],
[    52,  89540],
[    42, 162140],
[    44,  85910],
[    61, 122210],
[    50,  56870],
[    58, 157300],
[    55, 137940],
[    43, 171820],
[    49,  26620],
[    51, 116160],
[    55, 181500]
```

```
[    55, 181500],
[    62,  50820],
[    38,  70180],
[    50,  52030],
[    63, 130680],
[    52,  78650],
[    43,  94380],
[    49, 116160],
[    62, 173030],
[    44,  96800],
[    38, 110110],
[    40, 174240],
[    63, 123420],
[    38,  72600],
[    40,  64130],
[    39, 152460],
[    59, 160930],
[    43,  87120],
[    45,  96800],
[    38, 177870],
[    42,  50820],
[    43, 129470],
[    52, 104060],
[    41, 135520],
[    49,  95590],
[    43,  68970],
[    40,  96800],
[    49,  99220],
[    56, 173030],
[    45, 180290],
[    41,  71390],
[    53, 106480],
[    59, 125840],
[    44,  87120],
[    54, 176660],
[    38,  60500],
[    60, 147620],
[    44,  62920],
[    38, 117370],
[    47,  47190],
[    40,  62920],
[    51, 162140],
[    40, 176660],
[    53,  53240],
[    55, 108900],
[    44,  87120],
[    43,  68970],
[    61, 114950],
[    48, 158510],
[    38,  93170],
[    39, 174240],
[    58, 151250],
[    38,  87120],
[    51, 108900],
[    45, 130680],
[    43,  90750],
[    40,  89540],
[    50, 174240],
[    43,  73810],
[    46, 160930],
[    62,  91960],
[    63,  50820],
[    42, 128260],
[    60,  31460],
[    60,  89540],
[    41,  85910],
[    52, 106480],
[    55,  45980],
[    53,  43560],
[    62, 106480],
[    38,  73810],
[    40,  84700],
[    55,  25410],
```

```
[    55,  25410],
[    51, 170610],
[    40, 112530],
[    40,  75020],
[    51, 166980],
[    44,  95590],
[    40,  94380],
[    42, 162140],
[    52, 107690],
[    58,  47190],
[    40,  93170],
[    38,  68970],
[    39,  76230],
[    45,  88330],
[    46, 135520],
[    48,  95590],
[    49, 141570],
[    61,  45980],
[    51,  89540],
[    40, 165770],
[    40,  95590],
[    43,  72600],
[    45,  65340],
[    54, 162140],
[    50, 136730],
[    39, 151250],
[    41,  60500],
[    45,  84700],
[    42, 116160],
[    41,  60500],
[    52, 170610],
[    42,  95590],
[    42,  90750],
[    57, 125840],
[    38,  66550],
[    48,  38720],
[    39,  72600],
[    55, 166980],
[    56,  99220],
[    44,  62920],
[    51,  36300],
[    51, 158510],
[    44,  72600],
[    44,  87120],
[    45,  90750],
[    39, 142780],
[    50, 129470],
[    41,  61710],
[    51, 143990],
[    45,  78650],
[    43,  78650],
[    60,  72600],
[    39,  65340],
[    61, 174240],
[    38,  95590],
[    41,  66550],
[    42, 147620],
[    56, 125840],
[    38,  90750],
[    41,  78650],
[    50,  61710],
[    50, 127050],
[    44,  76230],
[    56,  87120],
[    57, 130680],
[    42,  93170],
[    41,  73810],
[    41, 136730],
[    40,  90750],
[    45, 108900],
[    40,  68970],
[    39, 119790],
[    63,  41140],
```

```
       [    57,  84700],
       [    44,  87120],
       [    43,  85910],
       [    45,  65340],
       [    46, 156090],
       [    56,  41140],
       [    50,  60500],
       [    45,  95590],
       [    45, 125840],
       [    62,  35090],
       [    61,  56870],
       [    49, 106480],
       [    41,  85910],
       [    57,  31460],
       [    63,  55660],
       [    63, 100430],
       [    42,  88330],
       [    62, 157300],
       [    40,  96800],
       [    49,  38720],
       [    49,  89540],
       [    45,  64130],
       [    44, 105270],
       [    61,  27830],
       [    45,  77440],
       [    51,  39930],
       [    47, 168190],
       [    52,  33880],
       [    60,  39930],
       [    59,  72600],
       [    52,  47190],
       [    42,  85910],
       [    50,  41140],
       [    51,  42350],
       [    51,  39930],
       [    50,  27830],
       [    48,  54450],
       [    63,  50820],
       [    42,  71390],
       [    49,  49610],
       [    54,  27830],
       [    53,  24200],
       [    39,  39930],
       [    52,  43560]], dtype=int64)
```

Step5: Encoding(not required)

step6 : spiliting of data set into training and testing set

In [35]:

```
from sklearn.model_selection import train_test_split
atrain,atest,btrain,btest=train_test_split(a,b,test_size=0.2,random_state=1)
```

In [36]:

```
atrain
```

Out[36]:

```
array([[    32,  33880],
       [    48,  26620],
       [    49, 141570],
       [    35,  21780],
       [    25,  98010],
       [    28, 105270],
       [    51,  39930],
       [    38,  70180],
```

```
[    50,  27830],
[    29,  38720],
[    35, 163350],
[    63,  41140],
[    55,  25410],
[    41,  66550],
[    28, 108900],
[    61,  45980],
[    52, 106480],
[    40,  94380],
[    38,  93170],
[    37,  52030],
[    23,  59290],
[    49, 106480],
[    34,  41140],
[    50,  36300],
[    38,  60500],
[    42, 116160],
[    36, 136730],
[    52, 104060],
[    48,  95590],
[    47,  47190],
[    44,  71390],
[    45,  64130],
[    38,  88330],
[    44,  87120],
[    30, 116160],
[    33, 140360],
[    44,  62920],
[    44,  62920],
[    23,  99220],
[    49,  49610],
[    30,  37510],
[    38,  85910],
[    52,  33880],
[    38, 110110],
[    40,  90750],
[    35, 141570],
[    39,  90750],
[    23, 104060],
[    41,  60500],
[    52,  43560],
[    43,  78650],
[    40,  93170],
[    63,  55660],
[    51, 166980],
[    43,  85910],
[    39,  76230],
[    29,  98010],
[    36,  37510],
[    44,  61710],
[    49,  89540],
[    21,  99220],
[    40,  39930],
[    38,  64130],
[    31,  71390],
[    40,  85910],
[    21,  53240],
[    41,  73810],
[    38,  78650],
[    34,  79860],
[    50,  30250],
[    45, 125840],
[    22,  22990],
[    24,  87120],
[    31,  38720],
[    55, 181500],
[    56, 125840],
[    28,  96800],
[    29,  87120],
[    29,  18150],
[    43, 171820],
```

```
[    38,  45980],
[    60, 147620],
[    44, 105270],
[    27, 107690],
[    40,  62920],
[    38, 130680],
[    35, 121000],
[    38, 177870],
[    22,  31460],
[    49,  99220],
[    50, 127050],
[    48,  54450],
[    41,  61710],
[    40, 112530],
[    32,  52030],
[    44,  36300],
[    43,  72600],
[    30, 106480],
[    30, 108900],
[    42,  85910],
[    26,  33880],
[    54, 176660],
[    26,  76230],
[    40,  96800],
[    43,  56870],
[    51,  35090],
[    38,  71390],
[    29,  36300],
[    42, 128260],
[    31, 101640],
[    62,  91960],
[    43,  68970],
[    38, 117370],
[    29, 104060],
[    44,  76230],
[    32,  56870],
[    29,  20570],
[    61,  27830],
[    33,  75020],
[    28,  39930],
[    62,  35090],
[    31,  44770],
[    42,  93170],
[    50, 129470],
[    55, 166980],
[    32,  52030],
[    30, 165770],
[    27,  66550],
[    43,  87120],
[    24,  19360],
[    60,  31460],
[    33, 163350],
[    51, 143990],
[    43,  68970],
[    38,  90750],
[    31,  53240],
[    40,  89540],
[    38,  32670],
[    40,  95590],
[    35, 145200],
[    32,  90750],
[    33,  20570],
[    28,  95590],
[    43, 129470],
[    27,  22990],
[    44,  87120],
[    41,  73810],
[    39, 174240],
[    45,  77440],
[    51,  39930],
[    51,  36300],
[    26,  24200],
```

```
[    58, 157300],
[    32, 179080],
[    45,  78650],
[    30,  65340],
[    40,  66550],
[    28,  26620],
[    62, 173030],
[    45,  65340],
[    30,  20570],
[    50,  59290],
[    31,  71390],
[    36,  83490],
[    34,  82280],
[    38,  27830],
[    38,  26620],
[    60,  39930],
[    33, 129470],
[    49,  27830],
[    44,  72600],
[    36,  49610],
[    36,  72600],
[    50,  56870],
[    31,  66550],
[    48,  38720],
[    38,  90750],
[    62,  50820],
[    50,  60500],
[    50,  61710],
[    38,  95590],
[    42,  50820],
[    43,  94380],
[    26,  58080],
[    42,  90750],
[    38,  68970],
[    58,  47190],
[    34,  85910],
[    31, 148830],
[    45,  96800],
[    43,  71390],
[    51,  89540],
[    24, 106480],
[    56,  87120],
[    30,  70180],
[    38,  56870],
[    31, 107690],
[    29,  96800],
[    35, 141570],
[    45,  90750],
[    36, 180290],
[    44,  96800],
[    34,  70180],
[    42, 162140],
[    30,  24200],
[    32, 100430],
[    22,  84700],
[    22, 102850],
[    32,  73810],
[    42, 162140],
[    34,  91960],
[    44,  87120],
[    29,  96800],
[    43,  73810],
[    38,  30250],
[    51, 116160],
[    45, 180290],
[    31,  95590],
[    54, 162140],
[    36,  33880],
[    45,  65340],
[    48,  26620],
[    40,  68970],
[    37, 135520],
```

```
[    38,  47190],
[    25,  32670],
[    38,  87120],
[    42,  71390],
[    23,  89540],
[    49,  38720],
[    29,  52030],
[    32, 100430],
[    58, 151250],
[    40, 176660],
[    48, 158510],
[    36,  52030],
[    44,  54450],
[    45,  95590],
[    40, 165770],
[    27, 101640],
[    35,  21780],
[    59, 125840],
[    52,  47190],
[    31, 102850],
[    56, 173030],
[    33, 107690],
[    60,  72600],
[    43,  90750],
[    44,  95590],
[    23,  99220],
[    25,  66550],
[    38, 106480],
[    57,  84700],
[    34,  18150],
[    53,  43560],
[    45,  78650],
[    37,  52030],
[    45, 130680],
[    57,  31460],
[    22,  25410],
[    39,  60500],
[    40,  84700],
[    39, 152460],
[    50,  24200],
[    33,  95590],
[    62, 100430],
[    32,  96800],
[    46, 135520],
[    41,  96800],
[    61, 174240],
[    39, 151250],
[    52,  33880],
[    21,  62920],
[    33,  18150],
[    62, 106480],
[    30,  68970],
[    41,  85910],
[    34, 107690],
[    50,  41140],
[    34,  89540],
[    40,  87120],
[    43,  68970],
[    62, 157300],
[    52,  78650],
[    51, 108900],
[    49,  26620],
[    57, 125840],
[    38,  24200],
[    52, 170610],
[    51,  49610],
[    38,  66550],
[    39,  72600],
[    35, 181500],
[    21,  82280],
[    27,  66550],
[    45, 108900],
```

```
       [    41,  71390],
       [    63, 130680],
       [    25,  76230],
       [    27,  38720],
       [    49,  71390],
       [    51, 162140],
       [    44,  87120],
       [    53,  53240],
       [    41,  85910],
       [    27,  27830],
       [    38,  73810],
       [    40,  96800],
       [    27,  32670],
       [    29, 101640],
       [    37,  30250],
       [    39,  65340],
       [    24,  82280],
       [    44,  85910],
       [    63,  50820],
       [    55, 108900],
       [    23,  27830],
       [    54,  27830],
       [    49,  95590],
       [    33,  59290]], dtype=int64)
```

step7 : Feature scaling

In [37]:

```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
atrain=sc.fit_transform(atrain)
atest=sc.fit_transform(atest)
```

In [38]:

```
atrain
```

Out[38]:

```
array([[-0.80330081, -1.19121795],
       [ 0.75697997, -1.36859801],
       [ 0.85449752,  1.43991958],
       [-0.51074816, -1.48685138],
       [-1.48592365,  0.37563923],
       [-1.19337101,  0.55301929],
       [ 1.04953262, -1.04340124],
       [-0.21819552, -0.30431766],
       [ 0.95201507, -1.33903467],
       [-1.09585346, -1.07296458],
       [-0.51074816,  1.97205975],
       [ 2.21974321, -1.0138379 ],
       [ 1.43960282, -1.39816136],
       [ 0.07435713, -0.39300769],
       [-1.19337101,  0.64170932],
       [ 2.02470811, -0.89558452],
       [ 1.14705017,  0.58258263],
       [-0.02316042,  0.2869492 ],
       [-0.21819552,  0.25738586],
       [-0.31571307, -0.74776781],
       [-1.68095875, -0.57038775],
       [ 0.85449752,  0.58258263],
       [-0.60826571, -1.0138379 ],
       [ 0.95201507, -1.13209127],
       [-0.21819552, -0.54082441],
       [ 0.17187468,  0.81908937],
       [-0.41323061,  1.32166621],
       [ 1.14705017,  0.52345594],
       [ 0.75697997,  0.31651254],
       [ 0.65946243, -0.86602118],
```

```
[ 0.36690978, -0.27475432],
[ 0.46442733, -0.45213438],
[-0.21819552,  0.13913248],
[ 0.36690978,  0.10956914],
[-0.99833591,  0.81908937],
[-0.70578326,  1.41035623],
[ 0.36690978, -0.48169772],
[ 0.36690978, -0.48169772],
[-1.68095875,  0.40520257],
[ 0.85449752, -0.80689449],
[-0.99833591, -1.10252793],
[-0.21819552,  0.0800058 ],
[ 1.14705017, -1.19121795],
[-0.21819552,  0.67127266],
[-0.02316042,  0.19825917],
[-0.51074816,  1.43991958],
[-0.12067797,  0.19825917],
[-1.68095875,  0.52345594],
[ 0.07435713, -0.54082441],
[ 1.14705017, -0.95471121],
[ 0.26939223, -0.09737426],
[-0.02316042,  0.25738586],
[ 2.21974321, -0.65907778],
[ 1.04953262,  2.06074978],
[ 0.26939223,  0.0800058 ],
[-0.12067797, -0.15650095],
[-1.09585346,  0.37563923],
[-0.41323061, -1.10252793],
[ 0.36690978, -0.51126106],
[ 0.85449752,  0.16869583],
[-1.87599385,  0.40520257],
[-0.02316042, -1.04340124],
[-0.21819552, -0.45213438],
[-0.90081836, -0.27475432],
[-0.02316042,  0.0800058 ],
[-1.87599385, -0.71820447],
[ 0.07435713, -0.21562763],
[-0.21819552, -0.09737426],
[-0.60826571, -0.06781092],
[ 0.95201507, -1.27990798],
[ 0.46442733,  1.05559612],
[-1.7784763 , -1.45728804],
[-1.5834412 ,  0.10956914],
[-0.90081836, -1.07296458],
[ 1.43960282,  2.4155099 ],
[ 1.53712037,  1.05559612],
[-1.19337101,  0.34607588],
[-1.09585346,  0.10956914],
[-1.09585346, -1.57554141],
[ 0.26939223,  2.17900315],
[-0.21819552, -0.89558452],
[ 1.92719056,  1.58773629],
[ 0.36690978,  0.55301929],
[-1.29088856,  0.61214597],
[-0.02316042, -0.48169772],
[-0.21819552,  1.17384949],
[-0.51074816,  0.93734275],
[-0.21819552,  2.32681987],
[-1.7784763 , -1.25034464],
[ 0.85449752,  0.40520257],
[ 0.95201507,  1.08515946],
[ 0.75697997, -0.68864112],
[ 0.07435713, -0.51126106],
[-0.02316042,  0.73039934],
[-0.80330081, -0.74776781],
[ 0.36690978, -1.13209127],
[ 0.26939223, -0.24519098],
[-0.99833591,  0.58258263],
[-0.99833591,  0.64170932],
[ 0.17187468,  0.0800058 ],
[-1.3884061 , -1.19121795],
[ 1.34208527,  2.29725653],
```

```
[-1.3884061 , -0.15650095],
[-0.02316042,  0.34607588],
[ 0.26939223, -0.62951444],
[ 1.04953262, -1.16165461],
[-0.21819552, -0.27475432],
[-1.09585346, -1.13209127],
[ 0.17187468,  1.1147228 ],
[-0.90081836,  0.46432926],
[ 2.12222566,  0.22782251],
[ 0.26939223, -0.33388101],
[-0.21819552,  0.84865272],
[-1.09585346,  0.52345594],
[ 0.36690978, -0.15650095],
[-0.80330081, -0.62951444],
[-1.09585346, -1.51641473],
[ 2.02470811, -1.33903467],
[-0.70578326, -0.18606429],
[-1.19337101, -1.04340124],
[ 2.12222566, -1.16165461],
[-0.90081836, -0.92514787],
[ 0.17187468,  0.25738586],
[ 0.95201507,  1.14428615],
[ 1.43960282,  2.06074978],
[-0.80330081, -0.74776781],
[-0.99833591,  2.03118644],
[-1.29088856, -0.39300769],
[ 0.26939223,  0.10956914],
[-1.5834412 , -1.54597807],
[ 1.92719056, -1.25034464],
[-0.70578326,  1.97205975],
[ 1.04953262,  1.49904626],
[ 0.26939223, -0.33388101],
[-0.21819552,  0.19825917],
[-0.90081836, -0.71820447],
[-0.02316042,  0.16869583],
[-0.21819552, -1.2207813 ],
[-0.02316042,  0.31651254],
[-0.51074816,  1.52860961],
[-0.80330081,  0.19825917],
[-0.70578326, -1.51641473],
[-1.19337101,  0.31651254],
[ 0.26939223,  1.14428615],
[-1.29088856, -1.45728804],
[ 0.36690978,  0.10956914],
[ 0.07435713, -0.21562763],
[-0.12067797,  2.23812984],
[ 0.46442733, -0.1269376 ],
[ 1.04953262, -1.04340124],
[ 1.04953262, -1.13209127],
[-1.3884061 , -1.4277247 ],
[ 1.73215547,  1.82424304],
[-0.80330081,  2.35638321],
[ 0.46442733, -0.09737426],
[-0.99833591, -0.42257103],
[-0.02316042, -0.39300769],
[-1.19337101, -1.36859801],
[ 2.12222566,  2.2085665 ],
[ 0.46442733, -0.42257103],
[-0.99833591, -1.51641473],
[ 0.95201507, -0.57038775],
[-0.90081836, -0.27475432],
[-0.41323061,  0.02087911],
[-0.60826571, -0.00868423],
[-0.21819552, -1.33903467],
[-0.21819552, -1.36859801],
[ 1.92719056, -1.04340124],
[-0.70578326,  1.14428615],
[ 0.85449752, -1.33903467],
[ 0.36690978, -0.24519098],
[-0.41323061, -0.80689449],
[-0.41323061, -0.24519098],
[ 0.95201507, -0.62951444],
```

```
[-0.90081836, -0.39300769],
[ 0.75697997, -1.07296458],
[-0.21819552,  0.19825917],
[ 2.12222566, -0.77733115],
[ 0.95201507, -0.54082441],
[ 0.95201507, -0.51126106],
[-0.21819552,  0.31651254],
[ 0.17187468, -0.77733115],
[ 0.26939223,  0.2869492 ],
[-1.3884061 , -0.59995109],
[ 0.17187468,  0.19825917],
[-0.21819552, -0.33388101],
[ 1.73215547, -0.86602118],
[-0.60826571,  0.0800058 ],
[-0.90081836,  1.61729964],
[ 0.46442733,  0.34607588],
[ 0.26939223, -0.27475432],
[ 1.04953262,  0.16869583],
[-1.5834412 ,  0.58258263],
[ 1.53712037,  0.10956914],
[-0.99833591, -0.30431766],
[-0.21819552, -0.62951444],
[-0.90081836,  0.61214597],
[-1.09585346,  0.34607588],
[-0.51074816,  1.43991958],
[ 0.46442733,  0.19825917],
[-0.41323061,  2.38594656],
[ 0.36690978,  0.34607588],
[-0.60826571, -0.30431766],
[ 0.17187468,  1.94249641],
[-0.99833591, -1.4277247 ],
[-0.80330081,  0.43476591],
[-1.7784763 ,  0.05044245],
[-1.7784763 ,  0.4938926 ],
[-0.80330081, -0.21562763],
[ 0.17187468,  1.94249641],
[-0.60826571,  0.22782251],
[ 0.36690978,  0.10956914],
[-1.09585346,  0.34607588],
[ 0.26939223, -0.21562763],
[-0.21819552, -1.27990798],
[ 1.04953262,  0.81908937],
[ 0.46442733,  2.38594656],
[-0.90081836,  0.31651254],
[ 1.34208527,  1.94249641],
[-0.41323061, -1.19121795],
[ 0.46442733, -0.42257103],
[ 0.75697997, -1.36859801],
[-0.02316042, -0.33388101],
[-0.31571307,  1.29210286],
[-0.21819552, -0.86602118],
[-1.48592365, -1.2207813 ],
[-0.21819552,  0.10956914],
[ 0.17187468, -0.27475432],
[-1.68095875,  0.16869583],
[ 0.85449752, -1.07296458],
[-1.09585346, -0.74776781],
[-0.80330081,  0.43476591],
[ 1.73215547,  1.67642632],
[-0.02316042,  2.29725653],
[ 0.75697997,  1.85380638],
[-0.41323061, -0.74776781],
[ 0.36690978, -0.68864112],
[ 0.46442733,  0.31651254],
[-0.02316042,  2.03118644],
[-1.29088856,  0.46432926],
[-0.51074816, -1.48685138],
[ 1.82967301,  1.05559612],
[ 1.14705017, -0.86602118],
[-0.90081836,  0.4938926 ],
[ 1.53712037,  2.2085665 ],
[-0.70578326,  0.61214597],
[-1.02710956, -0.24510008],
```

```
[ 1.92719056, -0.24519098],
[ 0.26939223,  0.19825917],
[ 0.36690978,  0.31651254],
[-1.68095875,  0.40520257],
[-1.48592365, -0.39300769],
[-0.21819552,  0.58258263],
[ 1.63463792,  0.05044245],
[-0.60826571, -1.57554141],
[ 1.24456772, -0.95471121],
[ 0.46442733, -0.09737426],
[-0.31571307, -0.74776781],
[ 0.46442733,  1.17384949],
[ 1.63463792, -1.25034464],
[-1.7784763 , -1.39816136],
[-0.12067797, -0.54082441],
[-0.02316042,  0.05044245],
[-0.12067797,  1.70598967],
[ 0.95201507, -1.4277247 ],
[-0.70578326,  0.31651254],
[ 2.12222566,  0.43476591],
[-0.80330081,  0.34607588],
[ 0.56194488,  1.29210286],
[ 0.07435713,  0.34607588],
[ 2.02470811,  2.23812984],
[-0.12067797,  1.67642632],
[ 1.14705017, -1.19121795],
[-1.87599385, -0.48169772],
[-0.70578326, -1.57554141],
[ 2.12222566,  0.58258263],
[-0.99833591, -0.33388101],
[ 0.07435713,  0.0800058 ],
[-0.60826571,  0.61214597],
[ 0.95201507, -1.0138379 ],
[-0.60826571,  0.16869583],
[-0.02316042,  0.10956914],
[ 0.26939223, -0.33388101],
[ 2.12222566,  1.82424304],
[ 1.14705017, -0.09737426],
[ 1.04953262,  0.64170932],
[ 0.85449752, -1.36859801],
[ 1.63463792,  1.05559612],
[-0.21819552, -1.4277247 ],
[ 1.14705017,  2.14943981],
[ 1.04953262, -0.80689449],
[-0.21819552, -0.39300769],
[-0.12067797, -0.24519098],
[-0.51074816,  2.4155099 ],
[-1.87599385, -0.00868423],
[-1.29088856, -0.39300769],
[ 0.46442733,  0.64170932],
[ 0.07435713, -0.27475432],
[ 2.21974321,  1.17384949],
[-1.48592365, -0.15650095],
[-1.29088856, -1.07296458],
[ 0.85449752, -0.27475432],
[ 1.04953262,  1.94249641],
[ 0.36690978,  0.10956914],
[ 1.24456772, -0.71820447],
[ 0.07435713,  0.0800058 ],
[-1.29088856, -1.33903467],
[-0.21819552, -0.21562763],
[-0.02316042,  0.34607588],
[-1.29088856, -1.2207813 ],
[-1.09585346,  0.46432926],
[-0.31571307, -1.27990798],
[-0.12067797, -0.42257103],
[-1.5834412 , -0.00868423],
[ 0.36690978,  0.0800058 ],
[ 2.21974321, -0.77733115],
[ 1.43960282,  0.64170932],
[-1.68095875, -1.33903467],
[ 1.34208527, -1.33903467],
[ 0.85449752, -0.21562754],
```

```
[ 0.85449752,  0.31651254],
[-0.70578326, -0.57038775]])
```

Part B: build my KNN classification model

step 1: training the classification model

In [42]:

```
from sklearn.neighbors import KNeighborsClassifier
KC=KNeighborsClassifier(n_neighbors=7,weights='uniform',p=2)
KC.fit(atrain,btrain)
```

Out[42]:

```
KNeighborsClassifier(n_neighbors=7)
```

step 2: testing the linear model

In [43]:

```
bestimated=KC.predict(atest)
```

step C: performance matrix

In [45]:

```
from sklearn.metrics import confusion_matrix,accuracy_score,precision_score
cm=confusion_matrix(btest,bestimated)
print(cm)
print(accuracy_score(btest,bestimated))
print(precision_score(btest,bestimated))
```

```
[[42  6]
 [ 2 30]]
0.9
0.8333333333333334
```

In [46]:

```
np.mean((True,True,False))
```

Out[46]:

```
0.6666666666666666
```

In [47]:

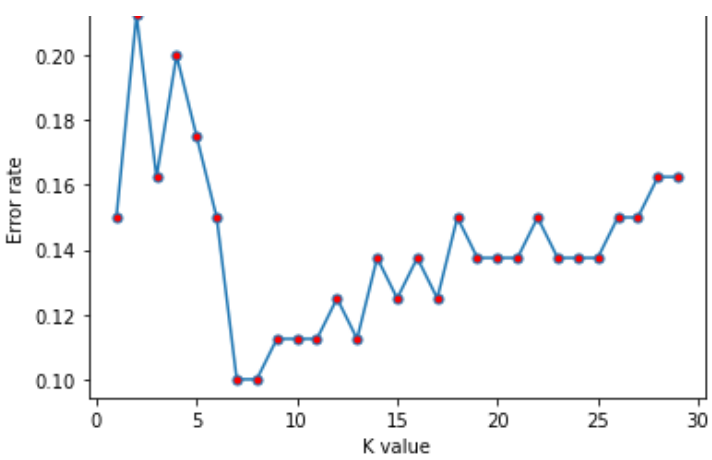
```
error_rate=[]
for i in range(1,30):
    KC=KNeighborsClassifier(n_neighbors=i)
    KC.fit(atrain,btrain)
    bpred_i=KC.predict(atest)
    error_rate.append(np.mean(bpred_i!=btest))
```

In [48]:

```
plt.plot(range(1,30),error_rate,marker='o',markerfacecolor='red',markersize=5)
plt.xlabel('K value')
plt.ylabel('Error rate')
```

Out[48]:

```
Text(0, 0.5, 'Error rate')
```



In [ ]: