k Nearest Neighbours

Import libraries

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

Import dataset

In [2]:
```
dataset = pd.read_csv("Social_Network_Ads.csv")
X = dataset.iloc[ : , : -1].values
y = dataset.iloc[ : , -1].values
```

Splitting dataset into training and testing

In [3]:
```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size = 0.1,
                                                    random_state = 0)
```

In [4]:
```
print(X_train, end = "\n\n\n")
print(X_test)
print(y_train)
print(y_test, end = "\n")
```

```
[[    27  88000]
 [    41  52000]
 [    27  84000]
 [    35  20000]
 [    43 112000]
 [    27  58000]
 [    37  80000]
 [    52  90000]
 [    26  30000]
 [    49  86000]
 [    57 122000]
 [    34  25000]
 [    35  57000]
 [    34 115000]
 [    59  88000]
 [    45  32000]
 [    29  83000]
 [    26  80000]
 [    49  28000]
 [    23  20000]
 [    32  18000]
 [    60  42000]
 [    19  76000]
 [    36  99000]
 [    19  26000]
 [    60  83000]
 [    24  89000]
 [    27  58000]
 [    40  47000]
 [    42  70000]
 [    32 150000]
 [    35  77000]
 [    22  63000]
 [    45  22000]
 [    27  89000]
 [    18  82000]
 [    42  79000]
 [    40  60000]
 [    53  34000]
 [    47 107000]
 [    58 144000]
 [    59  83000]
 [    24  55000]
 [    26  35000]
 [    58  38000]
 [    42  80000]
 [    40  75000]
 [    59 130000]
 [    46  41000]
 [    41  60000]
 [    42  64000]
 [    37 146000]
 [    23  48000]
 [    25  33000]
 [    24  84000]
 [    27  96000]
 [    23  63000]
 [    48  33000]
 [    48  90000]
 [    42 104000]
```

```
[    44  39000]
[    32 120000]
[    38  50000]
[    32 135000]
[    52  21000]
[    53 104000]
[    39  42000]
[    38  61000]
[    36  50000]
[    36  63000]
[    35  25000]
[    35  50000]
[    42  73000]
[    47  49000]
[    59  29000]
[    49  65000]
[    45 131000]
[    31  89000]
[    46  82000]
[    47  51000]
[    26  15000]
[    60 102000]
[    38 112000]
[    40 107000]
[    42  53000]
[    35  59000]
[    48  41000]
[    48 134000]
[    38 113000]
[    29 148000]
[    26  15000]
[    60  42000]
[    24  19000]
[    42 149000]
[    46  96000]
[    28  59000]
[    39  96000]
[    28  89000]
[    41  72000]
[    45  26000]
[    33  69000]
[    20  82000]
[    31  74000]
[    42  80000]
[    35  72000]
[    33 149000]
[    40  71000]
[    51 146000]
[    46  79000]
[    35  75000]
[    38  51000]
[    36  75000]
[    37  78000]
[    38  61000]
[    60 108000]
[    20  82000]
[    57  74000]
[    42  65000]
[    26  80000]
[    46 117000]
```

```
[    35  61000]
[    21  68000]
[    28  44000]
[    41  87000]
[    37  33000]
[    27  90000]
[    39  42000]
[    28 123000]
[    31 118000]
[    25  87000]
[    35  71000]
[    37  70000]
[    35  39000]
[    47  23000]
[    35 147000]
[    48 138000]
[    26  86000]
[    25  79000]
[    52 138000]
[    51  23000]
[    35  60000]
[    33 113000]
[    30 107000]
[    48  33000]
[    41  80000]
[    48  96000]
[    31  18000]
[    31  71000]
[    43 129000]
[    59  76000]
[    18  44000]
[    36 118000]
[    42  90000]
[    47  30000]
[    26  43000]
[    40  78000]
[    46  59000]
[    59  42000]
[    46  74000]
[    35  91000]
[    28  59000]
[    40  57000]
[    59 143000]
[    57  26000]
[    52  38000]
[    47 113000]
[    53 143000]
[    35  27000]
[    58 101000]
[    45  45000]
[    23  82000]
[    46  23000]
[    42  65000]
[    28  84000]
[    38  59000]
[    26  84000]
[    29  28000]
[    37  71000]
[    22  55000]
[    48  35000]
```

```
[    49  28000]
[    38  65000]
[    27  17000]
[    46  28000]
[    48 141000]
[    26  17000]
[    35  97000]
[    39  59000]
[    24  27000]
[    32  18000]
[    46  88000]
[    35  58000]
[    56  60000]
[    47  34000]
[    40  72000]
[    32 100000]
[    19  21000]
[    25  90000]
[    35  88000]
[    28  32000]
[    50  20000]
[    40  59000]
[    50  44000]
[    35  72000]
[    40 142000]
[    46  32000]
[    39  71000]
[    20  74000]
[    29  75000]
[    31  76000]
[    47  25000]
[    40  61000]
[    34 112000]
[    38  80000]
[    42  75000]
[    47  47000]
[    39  75000]
[    19  25000]
[    37  80000]
[    36  60000]
[    41  52000]
[    36 125000]
[    48  29000]
[    36 126000]
[    51 134000]
[    27  57000]
[    38  71000]
[    39  61000]
[    22  27000]
[    33  60000]
[    48  74000]
[    58  23000]
[    53  72000]
[    32 117000]
[    54  70000]
[    30  80000]
[    58  95000]
[    26  52000]
[    45  79000]
[    24  55000]
```

```
[    40  75000]
[    33  28000]
[    44 139000]
[    22  18000]
[    33  51000]
[    43 133000]
[    24  32000]
[    46  22000]
[    35  55000]
[    54 104000]
[    48 119000]
[    35  53000]
[    37 144000]
[    23  66000]
[    37 137000]
[    31  58000]
[    33  41000]
[    45  22000]
[    30  15000]
[    19  19000]
[    49  74000]
[    39 122000]
[    35  73000]
[    39  71000]
[    24  23000]
[    41  72000]
[    29  83000]
[    54  26000]
[    35  44000]
[    37  75000]
[    29  47000]
[    31  68000]
[    42  54000]
[    30 135000]
[    52 114000]
[    50  36000]
[    56 133000]
[    29  61000]
[    30  89000]
[    26  16000]
[    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]]
```

```
[[    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]
 [    35  22000]
 [    35  47000]
 [    47 144000]
 [    41  51000]
 [    47 105000]
 [    23  28000]
 [    49 141000]
 [    28  87000]
 [    29  80000]
 [    37  62000]
 [    32  86000]
 [    21  88000]
 [    37  79000]
 [    57  60000]
 [    37  53000]
 [    24  58000]
 [    18  52000]
 [    22  81000]
 [    34  43000]
 [    31  34000]
 [    49  36000]]
[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 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 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]
[0 0 0 0 0 0 0 1 0 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]
```

Feature Scaling

In [5]:
```
from sklearn.preprocessing import StandardScaler
```

```
standard_scaler = StandardScaler()
X_train = standard_scaler.fit_transform(X_train)
X_test = standard_scaler.transform(X_test)
```

In [6]:
```
print("X_train = \n", X_train)
print("\n\n\n")
print("X_test = \n", X_test)
```

```
X_train =
 [[-1.05714987  0.53420426]
 [ 0.2798728  -0.51764734]
 [-1.05714987  0.41733186]
 [-0.29313691 -1.45262654]
 [ 0.47087604  1.23543867]
 [-1.05714987 -0.34233874]
 [-0.10213368  0.30045946]
 [ 1.33039061  0.59264046]
 [-1.15265148 -1.16044554]
 [ 1.04388575  0.47576806]
 [ 1.8078987   1.52761967]
 [-0.38863853 -1.30653604]
 [-0.29313691 -0.37155684]
 [-0.38863853  1.32309297]
 [ 1.99890194  0.53420426]
 [ 0.66187928 -1.10200934]
 [-0.86614663  0.38811376]
 [-1.15265148  0.30045946]
 [ 1.04388575 -1.21888174]
 [-1.43915634 -1.45262654]
 [-0.57964177 -1.51106275]
 [ 2.09440356 -0.80982834]
 [-1.82116282  0.18358706]
 [-0.19763529  0.85560337]
 [-1.82116282 -1.27731794]
 [ 2.09440356  0.38811376]
 [-1.34365472  0.56342236]
 [-1.05714987 -0.34233874]
 [ 0.18437118 -0.66373784]
 [ 0.37537442  0.00827846]
 [-0.57964177  2.34572647]
 [-0.29313691  0.21280516]
 [-1.53465796 -0.19624824]
 [ 0.66187928 -1.39419034]
 [-1.05714987  0.56342236]
 [-1.91666444  0.35889566]
 [ 0.37537442  0.27124136]
 [ 0.18437118 -0.28390254]
 [ 1.42589223 -1.04357314]
 [ 0.85288251  1.08934817]
 [ 1.90340032  2.17041787]
 [ 1.99890194  0.38811376]
 [-1.34365472 -0.42999304]
 [-1.15265148 -1.01435504]
 [ 1.90340032 -0.92670074]
 [ 0.37537442  0.30045946]
 [ 0.18437118  0.15436896]
 [ 1.99890194  1.76136447]
 [ 0.75738089 -0.83904644]
 [ 0.2798728  -0.28390254]
 [ 0.37537442 -0.16703014]
 [-0.10213368  2.22885407]
 [-1.43915634 -0.63451974]
 [-1.2481531  -1.07279124]
 [-1.34365472  0.41733186]
 [-1.05714987  0.76794907]
 [-1.43915634 -0.19624824]
 [ 0.94838413 -1.07279124]
 [ 0.94838413  0.59264046]
```

```
[ 0.37537442  1.00169387]
[ 0.56637766 -0.89748264]
[-0.57964177  1.46918347]
[-0.00663206 -0.57608354]
[-0.57964177  1.90745497]
[ 1.33039061 -1.42340844]
[ 1.42589223  1.00169387]
[ 0.08886956 -0.80982834]
[-0.00663206 -0.25468444]
[-0.19763529 -0.57608354]
[-0.19763529 -0.19624824]
[-0.29313691 -1.30653604]
[-0.29313691 -0.57608354]
[ 0.37537442  0.09593276]
[ 0.85288251 -0.60530164]
[ 1.99890194 -1.18966364]
[ 1.04388575 -0.13781204]
[ 0.66187928  1.79058257]
[-0.67514339  0.56342236]
[ 0.75738089  0.35889566]
[ 0.85288251 -0.54686544]
[-1.15265148 -1.59871705]
[ 2.09440356  0.94325767]
[-0.00663206  1.23543867]
[ 0.18437118  1.08934817]
[ 0.37537442 -0.48842924]
[-0.29313691 -0.31312064]
[ 0.94838413 -0.83904644]
[ 0.94838413  1.87823687]
[-0.00663206  1.26465677]
[-0.86614663  2.28729027]
[-1.15265148 -1.59871705]
[ 2.09440356 -0.80982834]
[-1.34365472 -1.48184464]
[ 0.37537442  2.31650837]
[ 0.75738089  0.76794907]
[-0.96164825 -0.31312064]
[ 0.08886956  0.76794907]
[-0.96164825  0.56342236]
[ 0.2798728   0.06671466]
[ 0.66187928 -1.27731794]
[-0.48414015 -0.02093964]
[-1.7256612   0.35889566]
[-0.67514339  0.12515086]
[ 0.37537442  0.30045946]
[-0.29313691  0.06671466]
[-0.48414015  2.31650837]
[ 0.18437118  0.03749656]
[ 1.23488899  2.22885407]
[ 0.75738089  0.27124136]
[-0.29313691  0.15436896]
[-0.00663206 -0.54686544]
[-0.19763529  0.15436896]
[-0.10213368  0.24202326]
[-0.00663206 -0.25468444]
[ 2.09440356  1.11856627]
[-1.7256612   0.35889566]
[ 1.8078987   0.12515086]
[ 0.37537442 -0.13781204]
[-1.15265148  0.30045946]
```

```
[ 0.75738089  1.38152917]
[-0.29313691 -0.25468444]
[-1.63015958 -0.05015774]
[-0.96164825 -0.75139214]
[ 0.2798728   0.50498616]
[-0.10213368 -1.07279124]
[-1.05714987  0.59264046]
[ 0.08886956 -0.80982834]
[-0.96164825  1.55683777]
[-0.67514339  1.41074727]
[-1.2481531   0.50498616]
[-0.29313691  0.03749656]
[-0.10213368  0.00827846]
[-0.29313691 -0.89748264]
[ 0.85288251 -1.36497224]
[-0.29313691  2.25807217]
[ 0.94838413  1.99510927]
[-1.15265148  0.47576806]
[-1.2481531   0.27124136]
[ 1.33039061  1.99510927]
[ 1.23488899 -1.36497224]
[-0.29313691 -0.28390254]
[-0.48414015  1.26465677]
[-0.77064501  1.08934817]
[ 0.94838413 -1.07279124]
[ 0.2798728   0.30045946]
[ 0.94838413  0.76794907]
[-0.67514339 -1.51106275]
[-0.67514339  0.03749656]
[ 0.47087604  1.73214637]
[ 1.99890194  0.18358706]
[-1.91666444 -0.75139214]
[-0.19763529  1.41074727]
[ 0.37537442  0.59264046]
[ 0.85288251 -1.16044554]
[-1.15265148 -0.78061024]
[ 0.18437118  0.24202326]
[ 0.75738089 -0.31312064]
[ 1.99890194 -0.80982834]
[ 0.75738089  0.12515086]
[-0.29313691  0.62185856]
[-0.96164825 -0.31312064]
[ 0.18437118 -0.37155684]
[ 1.99890194  2.14119977]
[ 1.8078987  -1.27731794]
[ 1.33039061 -0.92670074]
[ 0.85288251  1.26465677]
[ 1.42589223  2.14119977]
[-0.29313691 -1.24809984]
[ 1.90340032  0.91403957]
[ 0.66187928 -0.72217404]
[-1.43915634  0.35889566]
[ 0.75738089 -1.36497224]
[ 0.37537442 -0.13781204]
[-0.96164825  0.41733186]
[-0.00663206 -0.31312064]
[-1.15265148  0.41733186]
[-0.86614663 -1.21888174]
[-0.10213368  0.03749656]
[-1.53465796 -0.42999304]
```

```
[ 0.94838413 -1.01435504]
[ 1.04388575 -1.21888174]
[-0.00663206 -0.13781204]
[-1.05714987 -1.54028085]
[ 0.75738089 -1.21888174]
[ 0.94838413  2.08276357]
[-1.15265148 -1.54028085]
[-0.29313691  0.79716717]
[ 0.08886956 -0.31312064]
[-1.34365472 -1.24809984]
[-0.57964177 -1.51106275]
[ 0.75738089  0.53420426]
[-0.29313691 -0.34233874]
[ 1.71239708 -0.28390254]
[ 0.85288251 -1.04357314]
[ 0.18437118  0.06671466]
[-0.57964177  0.88482147]
[-1.82116282 -1.42340844]
[-1.2481531   0.59264046]
[-0.29313691  0.53420426]
[-0.96164825 -1.10200934]
[ 1.13938737 -1.45262654]
[ 0.18437118 -0.31312064]
[ 1.13938737 -0.75139214]
[-0.29313691  0.06671466]
[ 0.18437118  2.11198167]
[ 0.75738089 -1.10200934]
[ 0.08886956  0.03749656]
[-1.7256612   0.12515086]
[-0.86614663  0.15436896]
[-0.67514339  0.18358706]
[ 0.85288251 -1.30653604]
[ 0.18437118 -0.25468444]
[-0.38863853  1.23543867]
[-0.00663206  0.30045946]
[ 0.37537442  0.15436896]
[ 0.85288251 -0.66373784]
[ 0.08886956  0.15436896]
[-1.82116282 -1.30653604]
[-0.10213368  0.30045946]
[-0.19763529 -0.28390254]
[ 0.2798728  -0.51764734]
[-0.19763529  1.61527397]
[ 0.94838413 -1.18966364]
[-0.19763529  1.64449207]
[ 1.23488899  1.87823687]
[-1.05714987 -0.37155684]
[-0.00663206  0.03749656]
[ 0.08886956 -0.25468444]
[-1.53465796 -1.24809984]
[-0.48414015 -0.28390254]
[ 0.94838413  0.12515086]
[ 1.90340032 -1.36497224]
[ 1.42589223  0.06671466]
[-0.57964177  1.38152917]
[ 1.52139385  0.00827846]
[-0.77064501  0.30045946]
[ 1.90340032  0.73873096]
[-1.15265148 -0.51764734]
[ 0.66187928  0.27124136]
```

```
[-1.34365472 -0.42999304]
[ 0.18437118  0.15436896]
[-0.48414015 -1.21888174]
[ 0.56637766  2.02432737]
[-1.53465796 -1.51106275]
[-0.48414015 -0.54686544]
[ 0.47087604  1.84901877]
[-1.34365472 -1.10200934]
[ 0.75738089 -1.39419034]
[-0.29313691 -0.42999304]
[ 1.52139385  1.00169387]
[ 0.94838413  1.43996537]
[-0.29313691 -0.48842924]
[-0.10213368  2.17041787]
[-1.43915634 -0.10859394]
[-0.10213368  1.96589117]
[-0.67514339 -0.34233874]
[-0.48414015 -0.83904644]
[ 0.66187928 -1.39419034]
[-0.77064501 -1.59871705]
[-1.82116282 -1.48184464]
[ 1.04388575  0.12515086]
[ 0.08886956  1.52761967]
[-0.29313691  0.09593276]
[ 0.08886956  0.03749656]
[-1.34365472 -1.36497224]
[ 0.2798728   0.06671466]
[-0.86614663  0.38811376]
[ 1.52139385 -1.27731794]
[-0.29313691 -0.75139214]
[-0.10213368  0.15436896]
[-0.86614663 -0.66373784]
[-0.67514339 -0.05015774]
[ 0.37537442 -0.45921114]
[-0.77064501  1.90745497]
[ 1.33039061  1.29387487]
[ 1.13938737 -0.98513694]
[ 1.71239708  1.84901877]
[-0.86614663 -0.25468444]
[-0.77064501  0.56342236]
[-1.15265148 -1.56949895]
[-0.48414015 -1.13122744]
[ 0.2798728   0.06671466]
[-0.19763529 -1.07279124]
[ 1.61689547  1.61527397]
[ 0.94838413  1.79058257]
[ 0.2798728   0.03749656]
[-0.77064501 -0.22546634]
[-0.10213368  0.06671466]
[ 0.2798728  -0.19624824]
[ 1.90340032 -0.66373784]
[-0.77064501  1.35231107]
[-1.7256612  -0.60530164]
[-0.10213368  0.12515086]
[ 0.2798728  -0.31312064]
[ 1.04388575  0.56342236]
[-0.96164825  0.27124136]
[ 1.42589223  0.35889566]
[ 0.18437118 -0.37155684]
[ 2.09440356 -1.04357314]
```

```
[-0.29313691  1.11856627]
[-1.63015958  0.06671466]
[-0.00663206  0.03749656]
[ 0.08886956  1.06013007]
[-0.10213368 -0.37155684]
[-1.15265148  0.06671466]
[-0.29313691 -1.36497224]
[ 1.52139385  1.11856627]
[-0.77064501 -1.54028085]
[ 0.08886956  1.87823687]
[-0.86614663 -0.78061024]
[-0.48414015 -0.78061024]
[-0.29313691 -0.92670074]
[ 0.2798728  -0.72217404]
[ 0.2798728   0.06671466]
[ 0.08886956  1.87823687]
[-1.05714987  1.96589117]
[-1.63015958 -1.56949895]
[-1.15265148 -1.10200934]
[-0.67514339 -0.10859394]
[ 0.08886956  0.09593276]
[ 0.2798728   0.27124136]
[ 0.85288251 -0.57608354]
[ 0.2798728  -1.16044554]
[-0.10213368  0.68029476]
[ 2.09440356 -0.69295594]
[-1.2481531  -1.39419034]
[-0.96164825 -0.95591884]
[-0.00663206 -0.42999304]
[-0.19763529 -0.45921114]
[-1.7256612  -0.98513694]
[ 1.71239708  1.00169387]
[ 0.18437118 -0.37155684]
[ 0.37537442  1.11856627]
[-1.7256612  -1.36497224]
[ 0.18437118 -0.13781204]
[ 0.85288251 -1.45262654]
[-1.91666444  0.47576806]
[-0.29313691  0.27124136]
[ 1.8078987  -1.07279124]
[-0.38863853  0.06671466]
[ 1.04388575 -0.89748264]
[-1.05714987 -1.13122744]
[-1.82116282  0.00827846]
[ 0.08886956  0.27124136]
[-1.15265148  0.32967756]
[-1.2481531   0.30045946]
[-0.96164825  0.44654996]
[ 1.61689547 -0.89748264]
[ 1.13938737  0.53420426]
[ 1.04388575  0.53420426]
[ 1.33039061  2.34572647]
[-0.29313691 -0.13781204]
[ 0.37537442 -0.45921114]
[-0.38863853 -0.78061024]
[-0.10213368 -0.51764734]
[ 0.94838413 -1.16044554]
[-0.86614663 -0.78061024]
[-0.19763529 -0.51764734]
[-1.05714987 -0.45921114]
```

```
 [-1.15265148  1.41074727]]
```

```
X_test =
 [[-0.77064501  0.50498616]
 [-0.00663206 -0.57608354]
 [-0.29313691  0.15436896]
 [-0.77064501  0.27124136]
 [-0.29313691 -0.57608354]
 [-1.05714987 -1.45262654]
 [-0.67514339 -1.59871705]
 [-0.19763529  2.17041787]
 [-1.91666444 -0.05015774]
 [ 0.85288251 -0.78061024]
 [-0.77064501 -0.60530164]
 [-0.96164825 -0.42999304]
 [-0.10213368 -0.42999304]
 [ 0.08886956  0.21280516]
 [-1.7256612   0.47576806]
 [-0.57964177  1.38152917]
 [-0.10213368  0.21280516]
 [-1.82116282  0.44654996]
 [ 1.61689547  1.76136447]
 [-0.29313691 -1.39419034]
 [-0.29313691 -0.66373784]
 [ 0.85288251  2.17041787]
 [ 0.2798728  -0.54686544]
 [ 0.85288251  1.03091197]
 [-1.43915634 -1.21888174]
 [ 1.04388575  2.08276357]
 [-0.96164825  0.50498616]
 [-0.86614663  0.30045946]
 [-0.10213368 -0.22546634]
 [-0.57964177  0.47576806]
 [-1.63015958  0.53420426]
 [-0.10213368  0.27124136]
 [ 1.8078987  -0.28390254]
 [-0.10213368 -0.48842924]
 [-1.34365472 -0.34233874]
 [-1.91666444 -0.51764734]
 [-1.53465796  0.32967756]
 [-0.38863853 -0.78061024]
 [-0.67514339 -1.04357314]
 [ 1.04388575 -0.98513694]]
```

Training the model

In [7]:
```
from sklearn.neighbors import KNeighborsClassifier

knn_classifier = KNeighborsClassifier()
knn_classifier.fit(X_train, y_train)
```

Out[7]:
```
KNeighborsClassifier()
```

Predicting values

In [9]:
```
y_pred = knn_classifier.predict(X_test)
print("y_pred\ty_test")
print(np.concatenate((y_pred.reshape(len(y_pred), 1),
                      y_test.reshape(len(y_test), 1)), 1))
```

```
y_pred  y_test
[[0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [1 1]
 [0 0]
 [1 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [1 0]
 [0 0]
 [0 0]
 [1 1]
 [0 0]
 [0 0]
 [1 1]
 [0 0]
 [1 1]
 [0 0]
 [1 1]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 1]
 [1 1]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [1 1]]
```

Confusion matrix

In [10]:
```
from sklearn.metrics import accuracy_score, confusion_matrix

confusion_matrix_obj = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:\n", confusion_matrix_obj)
print("Accuracy score:\t", accuracy_score(y_test, y_pred))
```

```
Confusion Matrix:
 [[30  2]
 [ 1  7]]
Accuracy score:  0.925
```

Training model with parameters

In [11]:
```
from sklearn.neighbors import KNeighborsClassifier

knn_classifier_2 = KNeighborsClassifier(n_neighbors = 5,
                                        metric = 'minkowski',
                                        p = 2)
knn_classifier_2.fit(X_train, y_train)
```

Out[11]:
```
KNeighborsClassifier()
```

Predicting values

In [12]:
```
y_pred_2 = knn_classifier.predict(X_test)
print("y_pred\ty_test")
print(np.concatenate((y_pred_2.reshape(len(y_pred_2), 1),
                      y_test.reshape(len(y_test), 1)), 1))
```

```
y_pred  y_test
[[0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [1 1]
 [0 0]
 [1 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [1 0]
 [0 0]
 [0 0]
 [1 1]
 [0 0]
 [0 0]
 [1 1]
 [0 0]
 [1 1]
 [0 0]
 [1 1]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 1]
 [1 1]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [1 1]]
```

Confusion Matrix

In [13]:
```
from sklearn.metrics import accuracy_score, confusion_matrix

confusion_matrix_obj_2 = confusion_matrix(y_test, y_pred_2)
print("Confusion Matrix:\n", confusion_matrix_obj_2)
print("Accuracy score:\t", accuracy_score(y_test, y_pred_2))
```

```
Confusion Matrix:
 [[30  2]
 [ 1  7]]
Accuracy score:  0.925
```