K-Nearest-Neighbours

January 14, 2025

[ ]:
```
'''
    K Nearest Neighbours -->

    K-Nearest Neighbors (KNN) is a supervised machine learning algorithm that␣
 ↪is used
    for both classification and regression tasks. It classifies or predicts a␣
 ↪data point's
    output based on the output of its closest neighbors in the training dataset.
    The "K" in KNN refers to the number of nearest neighbors considered to make␣
 ↪a decision.
'''
```

[ ]:
```
'''
    K (Number of Neighbors) -->

    The value of K determines how many neighbors influence the decision.
    K=1 : The prediction is based solely on the nearest neighbor (can lead to␣
 ↪overfitting).
    K>1 : The prediction considers a broader context (less sensitive to noise).

    Distance Metrics: KNN relies on measuring the distance between data points␣
 ↪to identify neighbors

    Weighting Neighbors:

    Assign more weight to closer neighbors, as they are likely to have more␣
 ↪influence.
    Example: Use a weighting scheme like 1/distance.

    Decision Rule:

    Classification : Predict the majority class label among the K nearest␣
 ↪neighbors.
    Regression : Predict the average (or weighted average) value of the K␣
 ↪nearest neighbors.
'''
```

[ ]:
```
'''
    How KNN Works -->

    Choose a Value for K :
    K is the number of neighbors to consider (e.g., K=3 means the 3 nearest␣
 ↪points will be considered).

    Measure Distance :
    Calculate the distance between the test point and all training points using␣
 ↪a distance metric.

    Find the K Nearest Neighbors :
    Identify the K training samples closest to the test sample.

    Make Predictions :

    For Classification :
    Assign the class that is most frequent among the K neighbors (majority␣
 ↪vote).

    For Regression :
    Predict the average (or weighted average) of the values of the K neighbors.
'''
```

[ ]:
```
'''
    Advantages of KNN -->

    Simple to Implement: Easy to understand and directly applicable.
    No Training Phase: No model is trained, making it fast for small datasets.
    Adaptable: Works for both classification and regression tasks.
    Non-Parametric: No assumptions about the data distribution.

    Disadvantages of KNN -->

    Computationally Expensive : High memory and time consumption as it␣
 ↪calculates the distance for all points
    Curse of Dimensionality : Performance degrades with high-dimensional data␣
 ↪because distances become less meaningful.
    Sensitive to Noise : Outliers can significantly impact results.
'''
```

[ ]:
```
'''
    Tips for KNN -->

    Choosing K : Use cross-validation to select the best K.
    Smaller K : Sensitive to noise.
    Larger K : Can smooth out noise but may overlook finer patterns.
```

```
    Scaling Features :
    Normalize or standardize features since KNN relies on distances.

    Use Weighted KNN :
    Assign weights to neighbors based on their distances (closer neighbors have␣
 ↪more influence).

    Dimensionality Reduction :
    Apply PCA or feature selection to reduce dimensionality for better␣
 ↪performance.
'''
```

[15]:
```
#   Importing Libraries -->

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, confusion_matrix,␣
 ↪classification_report
```

[ ]:
```
#   Importing Dataset -->

data = pd.read_csv('Data/Social_Network_Ads.csv')
data.head(10)
```

[ ]:

	Age	EstimatedSalary	Purchased
0	19	19000	0
1	35	20000	0
2	26	43000	0
3	27	57000	0
4	19	76000	0
5	27	58000	0
6	27	84000	0
7	32	150000	1
8	25	33000	0
9	35	65000	0

[3]:
```
x_data = data.iloc[:, :-1].values
y_data = data.iloc[:, -1].values
```

[ ]:
```
#   Splitting Data -->
```

```
x_train, x_test, y_train, y_test = train_test_split(x_data, y_data, test_size=0.
 ↪25, random_state=42)
```

[5]:
```
x_train
```

[5]:
```
array([[    57, 122000],
       [    39,  71000],
       [    47,  25000],
       [    24,  19000],
       [    36,  50000],
       [    32, 150000],
       [    48,  29000],
       [    30, 107000],
       [    60,  34000],
       [    38,  61000],
       [    33,  31000],
       [    39,  71000],
       [    55,  39000],
       [    49,  39000],
       [    43, 112000],
       [    27,  20000],
       [    26,  17000],
       [    37,  93000],
       [    42,  54000],
       [    35,  61000],
       [    29,  75000],
       [    38,  80000],
       [    45,  26000],
       [    54, 108000],
       [    46,  23000],
       [    23,  28000],
       [    37,  75000],
       [    42,  65000],
       [    35,  71000],
       [    51, 146000],
       [    39,  96000],
       [    24,  89000],
       [    58,  95000],
       [    25,  22000],
       [    41,  59000],
       [    28,  89000],
       [    42,  80000],
       [    42, 108000],
       [    46,  96000],
       [    47, 113000],
       [    33,  28000],
       [    19,  25000],
```

```
[    49,  89000],
[    31,  15000],
[    30,  79000],
[    48, 141000],
[    32, 117000],
[    37,  71000],
[    18,  86000],
[    42,  79000],
[    27,  84000],
[    40,  65000],
[    57,  74000],
[    26,  15000],
[    26,  80000],
[    29,  43000],
[    33, 149000],
[    39,  42000],
[    54, 104000],
[    36,  33000],
[    46,  32000],
[    40, 142000],
[    37,  62000],
[    29, 148000],
[    37,  57000],
[    35,  50000],
[    42,  53000],
[    35,  38000],
[    41,  30000],
[    40,  72000],
[    26,  15000],
[    31,  68000],
[    35,  53000],
[    35,  25000],
[    30,  89000],
[    41,  72000],
[    28, 123000],
[    46,  82000],
[    22,  63000],
[    45,  22000],
[    30,  49000],
[    34,  25000],
[    40,  75000],
[    32, 117000],
[    23,  82000],
[    26,  80000],
[    48, 131000],
[    59, 143000],
[    35,  55000],
```

```
[    34,  43000],
[    39,  61000],
[    27,  96000],
[    60,  83000],
[    24,  55000],
[    58, 144000],
[    53, 104000],
[    35,  79000],
[    36,  99000],
[    57,  60000],
[    37, 137000],
[    33,  43000],
[    41,  71000],
[    52,  21000],
[    52, 150000],
[    37,  70000],
[    26,  84000],
[    26,  72000],
[    26,  52000],
[    41,  60000],
[    31,  66000],
[    37, 144000],
[    38,  61000],
[    31,  34000],
[    42,  75000],
[    46, 117000],
[    36,  52000],
[    38,  71000],
[    49,  88000],
[    57,  33000],
[    48, 138000],
[    47,  50000],
[    33,  69000],
[    37, 146000],
[    20,  82000],
[    40,  47000],
[    35,  22000],
[    20,  36000],
[    45,  45000],
[    26,  43000],
[    58, 101000],
[    40,  57000],
[    38, 112000],
[    37,  80000],
[    49,  28000],
[    36,  75000],
[    41,  72000],
```

```
[    35,  60000],
[    43, 129000],
[    41,  87000],
[    38, 113000],
[    58,  23000],
[    26,  32000],
[    32,  18000],
[    41,  52000],
[    31,  18000],
[    35,  88000],
[    48,  35000],
[    27,  89000],
[    35,  97000],
[    42,  73000],
[    21,  68000],
[    41,  72000],
[    33,  60000],
[    39, 134000],
[    28,  84000],
[    46,  88000],
[    24,  58000],
[    31, 118000],
[    50,  88000],
[    20,  82000],
[    32, 135000],
[    20,  86000],
[    35,  27000],
[    29,  43000],
[    21,  88000],
[    35,  59000],
[    45,  32000],
[    60,  42000],
[    35,  91000],
[    35,  44000],
[    18,  44000],
[    42, 149000],
[    45,  79000],
[    40,  60000],
[    24,  23000],
[    33,  51000],
[    42,  70000],
[    55, 130000],
[    50,  44000],
[    48, 119000],
[    19,  76000],
[    41,  72000],
[    40,  71000],
```

```
[    27,  88000],
[    36, 126000],
[    35,  75000],
[    35,  58000],
[    34, 115000],
[    35,  73000],
[    60, 108000],
[    25,  87000],
[    27,  54000],
[    21,  16000],
[    37,  74000],
[    35,  39000],
[    54,  70000],
[    47,  30000],
[    38,  50000],
[    35, 147000],
[    35,  77000],
[    41,  79000],
[    37,  33000],
[    60,  46000],
[    28,  59000],
[    23,  66000],
[    23,  63000],
[    30,  17000],
[    25,  33000],
[    59,  83000],
[    58,  38000],
[    18,  82000],
[    46,  59000],
[    27,  17000],
[    58,  47000],
[    48,  30000],
[    49,  65000],
[    50,  36000],
[    53,  72000],
[    40,  57000],
[    52, 114000],
[    59,  42000],
[    36,  63000],
[    42, 104000],
[    37,  52000],
[    48,  33000],
[    59,  29000],
[    37,  79000],
[    40,  61000],
[    49,  74000],
[    25,  90000],
```

```
[     30,  15000],
[     40,  78000],
[     24,  84000],
[     38,  50000],
[     45, 131000],
[     21,  72000],
[     35,  23000],
[     35,  20000],
[     31,  89000],
[     30,  80000],
[     47,  47000],
[     27,  90000],
[     35,  72000],
[     30, 116000],
[     39, 122000],
[     29,  83000],
[     41,  63000],
[     48,  90000],
[     38,  59000],
[     32,  18000],
[     39,  75000],
[     26,  81000],
[     39, 106000],
[     22,  55000],
[     36, 118000],
[     60,  42000],
[     28,  55000],
[     51, 134000],
[     49,  28000],
[     36,  60000],
[     56, 104000],
[     27,  58000],
[     24,  32000],
[     34,  72000],
[     28,  32000],
[     50,  20000],
[     33,  41000],
[     29,  47000],
[     22,  18000],
[     30, 135000],
[     47, 105000],
[     46,  79000],
[     48, 134000],
[     47,  49000],
[     49, 141000],
[     32, 100000],
[     38,  71000],
```

```
       [    19,  26000],
       [    37,  77000],
       [    47,  51000],
       [    40,  57000],
       [    36, 125000],
       [    20,  74000],
       [    31,  58000],
       [    41,  45000],
       [    42,  54000],
       [    28,  37000],
       [    39,  73000],
       [    28,  85000],
       [    38,  51000],
       [    47,  43000],
       [    37,  72000],
       [    49,  36000],
       [    45,  22000],
       [    35,  72000],
       [    24,  27000],
       [    26,  35000],
       [    43, 133000],
       [    39,  77000],
       [    32,  86000]], dtype=int64)
```

[11]:
```
y_train
```

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

[8]:
```
#   Scaling Features -->

sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
```

[9]:
```
x_train
```

[9]:
```
array([[ 1.8925893 ,  1.52189404],
       [ 0.1250379 ,  0.03213212],
       [ 0.9106163 , -1.31157471],
       [-1.34792161, -1.48684082],
       [-0.169554  , -0.58129926],
       [-0.56234321,  2.33980255],
       [ 1.0088136 , -1.19473064],
       [-0.75873781,  1.08372877],
       [ 2.1871812 , -1.04867555],
       [ 0.0268406 , -0.25997806],
       [-0.46414591, -1.1363086 ],
       [ 0.1250379 ,  0.03213212],
       [ 1.6961947 , -0.90262046],
       [ 1.1070109 , -0.90262046],
       [ 0.5178271 ,  1.22978386],
       [-1.05332971, -1.4576298 ],
       [-1.15152701, -1.54526286],
       [-0.0713567 ,  0.67477452],
       [ 0.4196298 , -0.46445519],
       [-0.2677513 , -0.25997806],
       [-0.85693511,  0.14897619],
       [ 0.0268406 ,  0.29503128],
       [ 0.7142217 , -1.28236369],
       [ 1.5979974 ,  1.11293979],
       [ 0.812419  , -1.36999675],
       [-1.44611891, -1.22394166],
       [-0.0713567 ,  0.14897619],
       [ 0.4196298 , -0.14313399],
       [-0.2677513 ,  0.03213212],
       [ 1.3034055 ,  2.22295848],
       [ 0.1250379 ,  0.76240757],
       [-1.34792161,  0.55793045],
       [ 1.9907866 ,  0.73319655],
       [-1.24972431, -1.39920777],
       [ 0.3214325 , -0.3184001 ],
       [-0.95513241,  0.55793045],
       [ 0.4196298 ,  0.29503128],
       [ 0.4196298 ,  1.11293979],
       [ 0.812419  ,  0.76240757],
       [ 0.9106163 ,  1.25899488],
       [-0.46414591, -1.22394166],
       [-1.83890811, -1.31157471],
       [ 1.1070109 ,  0.55793045],
       [-0.66054051, -1.60368489],
       [-0.75873781,  0.26582026],
```

```
[ 1.0088136 ,  2.07690339],
[-0.56234321,  1.37583895],
[-0.0713567 ,  0.03213212],
[-1.93710541,  0.47029739],
[ 0.4196298 ,  0.26582026],
[-1.05332971,  0.41187535],
[ 0.2232352 , -0.14313399],
[ 1.8925893 ,  0.11976517],
[-1.15152701, -1.60368489],
[-1.15152701,  0.29503128],
[-0.85693511, -0.78577639],
[-0.46414591,  2.31059153],
[ 0.1250379 , -0.8149874 ],
[ 1.5979974 ,  0.99609572],
[-0.169554  , -1.07788657],
[ 0.812419  , -1.10709759],
[ 0.2232352 ,  2.1061144 ],
[-0.0713567 , -0.23076704],
[-0.85693511,  2.28138051],
[-0.0713567 , -0.37682213],
[-0.2677513 , -0.58129926],
[ 0.4196298 , -0.49366621],
[-0.2677513 , -0.93183148],
[ 0.3214325 , -1.16551962],
[ 0.2232352 ,  0.06134314],
[-1.15152701, -1.60368489],
[-0.66054051, -0.05550093],
[-0.2677513 , -0.49366621],
[-0.2677513 , -1.31157471],
[-0.75873781,  0.55793045],
[ 0.3214325 ,  0.06134314],
[-0.95513241,  1.55110506],
[ 0.812419  ,  0.35345332],
[-1.54431621, -0.20155602],
[ 0.7142217 , -1.39920777],
[-0.75873781, -0.61051028],
[-0.36594861, -1.31157471],
[ 0.2232352 ,  0.14897619],
[-0.56234321,  1.37583895],
[-1.44611891,  0.35345332],
[-1.15152701,  0.29503128],
[ 1.0088136 ,  1.7847932 ],
[ 2.0889839 ,  2.13532542],
[-0.2677513 , -0.43524417],
[-0.36594861, -0.78577639],
[ 0.1250379 , -0.25997806],
[-1.05332971,  0.76240757],
```

```
[ 2.1871812 ,  0.38266434],
[-1.34792161, -0.43524417],
[ 1.9907866 ,  2.16453644],
[ 1.4998001 ,  0.99609572],
[-0.2677513 ,  0.26582026],
[-0.169554  ,  0.85004063],
[ 1.8925893 , -0.28918908],
[-0.0713567 ,  1.96005931],
[-0.46414591, -0.78577639],
[ 0.3214325 ,  0.03213212],
[ 1.4016028 , -1.42841878],
[ 1.4016028 ,  2.33980255],
[-0.0713567 ,  0.0029211 ],
[-1.15152701,  0.41187535],
[-1.15152701,  0.06134314],
[-1.15152701, -0.52287722],
[ 0.3214325 , -0.28918908],
[-0.66054051, -0.11392297],
[-0.0713567 ,  2.16453644],
[ 0.0268406 , -0.25997806],
[-0.66054051, -1.04867555],
[ 0.4196298 ,  0.14897619],
[ 0.812419  ,  1.37583895],
[-0.169554  , -0.52287722],
[ 0.0268406 ,  0.03213212],
[ 1.1070109 ,  0.52871943],
[ 1.8925893 , -1.07788657],
[ 1.0088136 ,  1.98927033],
[ 0.9106163 , -0.58129926],
[-0.46414591, -0.02628992],
[-0.0713567 ,  2.22295848],
[-1.74071081,  0.35345332],
[ 0.2232352 , -0.66893231],
[-0.2677513 , -1.39920777],
[-1.74071081, -0.99025351],
[ 0.7142217 , -0.72735435],
[-1.15152701, -0.78577639],
[ 1.9907866 ,  0.90846266],
[ 0.2232352 , -0.37682213],
[ 0.0268406 ,  1.22978386],
[-0.0713567 ,  0.29503128],
[ 1.1070109 , -1.22394166],
[-0.169554  ,  0.14897619],
[ 0.3214325 ,  0.06134314],
[-0.2677513 , -0.28918908],
[ 0.5178271 ,  1.72637117],
[ 0.3214325 ,  0.49950841],
```

```
[ 0.0268406 ,  1.25899488],
[ 1.9907866 , -1.36999675],
[-1.15152701, -1.10709759],
[-0.56234321, -1.51605184],
[ 0.3214325 , -0.52287722],
[-0.66054051, -1.51605184],
[-0.2677513 ,  0.52871943],
[ 1.0088136 , -1.01946453],
[-1.05332971,  0.55793045],
[-0.2677513 ,  0.79161859],
[ 0.4196298 ,  0.09055416],
[-1.64251351, -0.05550093],
[ 0.3214325 ,  0.06134314],
[-0.46414591, -0.28918908],
[ 0.1250379 ,  1.87242626],
[-0.95513241,  0.41187535],
[ 0.812419  ,  0.52871943],
[-1.34792161, -0.34761112],
[-0.66054051,  1.40504997],
[ 1.2052082 ,  0.52871943],
[-1.74071081,  0.35345332],
[-0.56234321,  1.90163728],
[-1.74071081,  0.47029739],
[-0.2677513 , -1.25315268],
[-0.85693511, -0.78577639],
[-1.64251351,  0.52871943],
[-0.2677513 , -0.3184001 ],
[ 0.7142217 , -1.10709759],
[ 2.1871812 , -0.8149874 ],
[-0.2677513 ,  0.61635248],
[-0.2677513 , -0.75656537],
[-1.93710541, -0.75656537],
[ 0.4196298 ,  2.31059153],
[ 0.7142217 ,  0.26582026],
[ 0.2232352 , -0.28918908],
[-1.34792161, -1.36999675],
[-0.46414591, -0.55208824],
[ 0.4196298 ,  0.0029211 ],
[ 1.6961947 ,  1.75558219],
[ 1.2052082 , -0.75656537],
[ 1.0088136 ,  1.43426099],
[-1.83890811,  0.17818721],
[ 0.3214325 ,  0.06134314],
[ 0.2232352 ,  0.03213212],
[-1.05332971,  0.52871943],
[-0.169554  ,  1.63873811],
[-0.2677513 ,  0.14897619],
```

```
[-0.2677513 , -0.34761112],
[-0.36594861,  1.31741692],
[-0.2677513 ,  0.09055416],
[ 2.1871812 ,  1.11293979],
[-1.24972431,  0.49950841],
[-1.05332971, -0.46445519],
[-1.64251351, -1.57447387],
[-0.0713567 ,  0.11976517],
[-0.2677513 , -0.90262046],
[ 1.5979974 ,  0.0029211 ],
[ 0.9106163 , -1.16551962],
[ 0.0268406 , -0.58129926],
[-0.2677513 ,  2.25216949],
[-0.2677513 ,  0.20739823],
[ 0.3214325 ,  0.26582026],
[-0.0713567 , -1.07788657],
[ 2.1871812 , -0.69814333],
[-0.95513241, -0.3184001 ],
[-1.44611891, -0.11392297],
[-1.44611891, -0.20155602],
[-0.75873781, -1.54526286],
[-1.24972431, -1.07788657],
[ 2.0889839 ,  0.38266434],
[ 1.9907866 , -0.93183148],
[-1.93710541,  0.35345332],
[ 0.812419  , -0.3184001 ],
[-1.05332971, -1.54526286],
[ 1.9907866 , -0.66893231],
[ 1.0088136 , -1.16551962],
[ 1.1070109 , -0.14313399],
[ 1.2052082 , -0.99025351],
[ 1.4998001 ,  0.06134314],
[ 0.2232352 , -0.37682213],
[ 1.4016028 ,  1.2882059 ],
[ 2.0889839 , -0.8149874 ],
[-0.169554  , -0.20155602],
[ 0.4196298 ,  0.99609572],
[-0.0713567 , -0.52287722],
[ 1.0088136 , -1.07788657],
[ 2.0889839 , -1.19473064],
[-0.0713567 ,  0.26582026],
[ 0.2232352 , -0.25997806],
[ 1.1070109 ,  0.11976517],
[-1.24972431,  0.58714146],
[-0.75873781, -1.60368489],
[ 0.2232352 ,  0.23660925],
[-1.34792161,  0.41187535],
```

```
[ 0.0268406 , -0.58129926],
[ 0.7142217 ,  1.7847932 ],
[-1.64251351,  0.06134314],
[-0.2677513 , -1.36999675],
[-0.2677513 , -1.4576298 ],
[-0.66054051,  0.55793045],
[-0.75873781,  0.29503128],
[ 0.9106163 , -0.66893231],
[-1.05332971,  0.58714146],
[-0.2677513 ,  0.06134314],
[-0.75873781,  1.34662793],
[ 0.1250379 ,  1.52189404],
[-0.85693511,  0.38266434],
[ 0.3214325 , -0.20155602],
[ 1.0088136 ,  0.58714146],
[ 0.0268406 , -0.3184001 ],
[-0.56234321, -1.51605184],
[ 0.1250379 ,  0.14897619],
[-1.15152701,  0.3242423 ],
[ 0.1250379 ,  1.05451775],
[-1.54431621, -0.43524417],
[-0.169554  ,  1.40504997],
[ 2.1871812 , -0.8149874 ],
[-0.95513241, -0.43524417],
[ 1.3034055 ,  1.87242626],
[ 1.1070109 , -1.22394166],
[-0.169554  , -0.28918908],
[ 1.794392  ,  0.99609572],
[-1.05332971, -0.34761112],
[-1.34792161, -1.10709759],
[-0.36594861,  0.06134314],
[-0.95513241, -1.10709759],
[ 1.2052082 , -1.4576298 ],
[-0.46414591, -0.84419842],
[-0.85693511, -0.66893231],
[-1.54431621, -1.51605184],
[-0.75873781,  1.90163728],
[ 0.9106163 ,  1.02530673],
[ 0.812419  ,  0.26582026],
[ 1.0088136 ,  1.87242626],
[ 0.9106163 , -0.61051028],
[ 1.1070109 ,  2.07690339],
[-0.56234321,  0.87925164],
[ 0.0268406 ,  0.03213212],
[-1.83890811, -1.28236369],
[-0.0713567 ,  0.20739823],
[ 0.9106163 , -0.55208824],
```

```
       [ 0.2232352 , -0.37682213],
       [-0.169554  ,  1.6095271 ],
       [-1.74071081,  0.11976517],
       [-0.66054051, -0.34761112],
       [ 0.3214325 , -0.72735435],
       [ 0.4196298 , -0.46445519],
       [-0.95513241, -0.96104249],
       [ 0.1250379 ,  0.09055416],
       [-0.95513241,  0.44108637],
       [ 0.0268406 , -0.55208824],
       [ 0.9106163 , -0.78577639],
       [-0.0713567 ,  0.06134314],
       [ 1.1070109 , -0.99025351],
       [ 0.7142217 , -1.39920777],
       [-0.2677513 ,  0.06134314],
       [-1.34792161, -1.25315268],
       [-1.15152701, -1.01946453],
       [ 0.5178271 ,  1.84321524],
       [ 0.1250379 ,  0.20739823],
       [-0.56234321,  0.47029739]])
```

[10]:
```
x_test
```

[10]:
```
array([[ 0.812419  , -1.39920777],
       [ 2.0889839 ,  0.52871943],
       [-0.95513241, -0.75656537],
       [ 1.0088136 ,  0.76240757],
       [-0.85693511, -1.22394166],
       [-0.75873781, -0.23076704],
       [ 0.9106163 ,  1.08372877],
       [-0.85693511,  0.38266434],
       [ 0.2232352 ,  0.14897619],
       [ 0.4196298 , -0.14313399],
       [-0.2677513 , -0.14313399],
       [ 1.4998001 , -1.04867555],
       [-1.44611891, -0.6397213 ],
       [-1.74071081, -1.36999675],
       [-0.75873781,  0.49950841],
       [-0.2677513 ,  1.11293979],
       [ 1.4016028 , -0.93183148],
       [ 0.812419  ,  0.11976517],
       [ 0.1250379 , -0.8149874 ],
       [ 1.794392  , -0.28918908],
       [-1.54431621, -1.25315268],
       [-0.85693511,  0.29503128],
       [ 0.9106163 , -1.36999675],
       [ 2.0889839 ,  0.17818721],
```

```
[-1.83890811, -1.48684082],
[ 1.3034055 , -1.36999675],
[ 0.4196298 ,  0.29503128],
[-0.0713567 , -0.49366621],
[ 1.6961947 ,  1.6095271 ],
[-1.83890811, -1.42841878],
[ 0.812419  , -0.84419842],
[-1.83890811,  0.0029211 ],
[-0.169554  ,  2.16453644],
[-0.95513241,  0.26582026],
[ 0.2232352 ,  1.08372877],
[-0.2677513 ,  0.14897619],
[-0.0713567 , -0.43524417],
[ 0.0268406 , -0.14313399],
[-1.15152701, -1.16551962],
[-1.93710541, -0.05550093],
[ 1.0088136 , -1.07788657],
[-1.34792161, -0.43524417],
[-1.93710541, -0.52287722],
[ 0.9106163 , -1.4576298 ],
[-1.74071081, -0.61051028],
[ 0.6160244 ,  2.01848135],
[-0.85693511, -0.25997806],
[-0.66054051,  0.03213212],
[ 1.0088136 , -0.84419842],
[-0.36594861, -0.78577639],
[-1.24972431,  0.26582026],
[ 1.4998001 ,  0.35345332],
[ 0.0268406 , -0.43524417],
[-1.24972431,  0.29503128],
[-0.0713567 ,  0.29503128],
[-1.05332971, -1.1363086 ],
[ 2.1871812 ,  0.93767368],
[-1.15152701,  1.40504997],
[-0.66054051,  0.11976517],
[-0.66054051,  0.17818721],
[ 0.3214325 , -0.55208824],
[-0.2677513 , -0.37682213],
[ 1.4016028 ,  0.58714146],
[-0.95513241,  0.49950841],
[-0.95513241, -0.3184001 ],
[-1.05332971,  1.96005931],
[ 0.4196298 ,  0.58714146],
[ 0.9106163 ,  2.16453644],
[ 0.1250379 , -0.3184001 ],
[-0.46414591,  1.25899488],
[ 1.4016028 ,  1.98927033],
```

```
       [-1.83890811,  0.44108637],
       [-1.05332971, -0.34761112],
       [-1.44611891, -1.4576298 ],
       [ 0.9106163 , -1.04867555],
       [-0.2677513 , -0.58129926],
       [ 1.794392  ,  1.84321524],
       [ 1.5979974 , -1.28236369],
       [-0.2677513 , -0.66893231],
       [-0.0713567 ,  0.23660925],
       [-1.05332971, -0.37682213],
       [ 0.812419  , -1.22394166],
       [ 0.1250379 ,  1.87242626],
       [ 0.6160244 , -0.90262046],
       [ 1.8925893 , -1.28236369],
       [-0.56234321,  1.46347201],
       [ 0.3214325 , -0.52287722],
       [ 1.0088136 ,  0.11976517],
       [-1.15152701,  0.47029739],
       [-1.54431621,  0.3242423 ],
       [ 1.1070109 ,  0.47029739],
       [-0.169554  , -0.46445519],
       [ 0.2232352 , -0.3184001 ],
       [ 0.3214325 ,  0.29503128],
       [-1.15152701, -1.57447387],
       [ 0.1250379 ,  0.26582026],
       [ 2.0889839 ,  1.75558219],
       [ 0.4196298 , -0.17234501],
       [ 1.4998001 ,  2.13532542],
       [-0.36594861,  1.22978386]])
```

[ ]:
```
#   Building Model -->

model = KNeighborsClassifier(n_neighbors=5, metric="minkowski", p=2)
model.fit(x_train, y_train)
```

[ ]:
```
KNeighborsClassifier()
```

[ ]:
```
#   Predicting Results -->

y_pred = model.predict(x_test)
y_pred
```

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

[16]:
```python
#   Accuracy Score, Confusion Matrix and Classification Report

acc_score = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)
```

[17]:
```python
print(acc_score)
```

```
0.93
```

[18]:
```python
print(conf_matrix)
```

```
[[59  4]
 [ 3 34]]
```

[19]:
```python
print(class_report)
```

```
              precision    recall  f1-score   support

           0       0.95      0.94      0.94        63
           1       0.89      0.92      0.91        37

    accuracy                           0.93       100
   macro avg       0.92      0.93      0.93       100
weighted avg       0.93      0.93      0.93       100
```