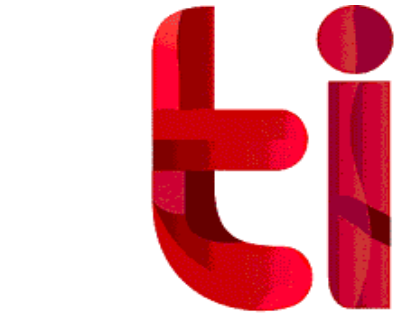
Name – Rahul Ranjan Singh
Roll No. – 102203140
Batch – 2CO4



THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Submission Date – 14, June 2024
Submission to – ELC Faculty

No. of Neighbours(N) and Training-Testing Split(Train:Test)

1. N = 2 and Train:Test = 60:40
   Accuracy Score = 0.9542857142857143

   Confusion Matrix

```
[[1612    0    2    0    0    1    2    0    0    0]
 [   0 1867    0    0    0    0    0    1    1    2]
 [  12   18 1652    7    2    1    1   15    2    3]
 [   5    7   18 1733    1   19    0    8   10    2]
 [   3   19    1    0 1594    0    5    3    0   17]
 [   7    3    3   54    3 1418   13    0    0    5]
 [  21    1    0    0    3   13 1574    0    0    0]
 [   1   29    8    5    5    0    0 1684    1   19]
 [  13   21   17   53   12   56    5   13 1388   10]
 [   8    8    2   17   66   10    1   66    8 1510]]
```

2. N = 2 and Train:Test = 70:30
   Accuracy Score = 0.9565079365079365
   Confusion Matrix

```
[[1230    0    2    0    0    1    2    0    0    1]
 [   0 1367    0    0    0    0    0    1    1    1]
 [   7    9 1211    5    1    0    1   14    2    2]
 [   1    4   12 1322    1   13    0    8    6    2]
 [   2   13    1    0 1181    0    4    3    0   11]
 [   5    0    1   43    4 1066    9    1    0    3]
 [  16    0    0    0    2   10 1188    0    0    0]
 [   0   22    8    0    4    1    0 1276    0   15]
 [  10   10   10   41   13   38    3    6 1058    8]
 [   7    5    1   14   42    5    0   52    8 1153]]
```

3. N = 2 and Train:Test = 75:25
   Accuracy Score = 0.9583809523809523

   Confusion Matrix

```
[[1017    0    2    0    0    1    2    0    0    0]
 [   0 1128    0    0    0    0    0    0    1    1]
 [   7    9 1023    4    1    0    0    8    0    1]
 [   0    3   10 1089    1   12    0    6    5    2]
 [   2   12    1    0  982    0    3    3    0   11]
 [   3    0    1   35    4  878    9    1    0    3]
 [  13    2    0    0    2    5  986    0    0    0]
 [   0   17    5    0    3    1    0 1066    0   11]
 [   8    7   10   30   11   32    2    4  903    6]
 [   6    5    1   11   36    5    0   33    7  991]]
```

4. N = 2 and Train:Test = 80:20
   Accuracy Score = 0.9577380952380953

   Confusion Matrix

```
[[817   0   1   0   0   1   2   0   0   0]
 [  0 897   0   0   0   0   0   0   1   1]
 [  7   8 830   3   1   0   0   8   0   1]
 [  0   3   8 879   1  10   0   6   5   1]
 [  2  11   1   0 763   0   1   3   0  10]
 [  2   0   0  27   3 719   7   1   0   3]
 [ 11   3   0   0   2   3 789   0   0   0]
 [  0  13   3   0   3   0   0 852   0   9]
 [  5   6   7  25   6  23   2   4 707   4]
 [  5   5   1  10  30   5   0  24   7 792]]
```

5. N = 2 and Train:Test = 90:10
   Accuracy Score = 0.9611904761904762

   Confusion Matrix

```
[[389   0   0   0   0   0   0   0   0   0]
 [  0 456   0   0   0   0   0   0   0   1]
 [  4   2 429   0   1   0   1   4   0   0]
 [  0   3   3 421   0   3   0   2   2   0]
 [  0   5   0   0 397   0   0   0   0   5]
 [  1   0   0  11   2 359   5   1   0   1]
 [  6   0   0   0   0   2 420   0   0   0]
 [  0   5   2   0   3   0   0 406   0   5]
 [  3   2   1  13   3  11   2   1 370   3]
 [  1   3   0   6  14   4   0  12   4 390]]
```

6. N = 2 and Train:Test = 95:5
   Accuracy Score = 0.9578061015839865

   Confusion Matrix

```
[[210   0   0   0   0   0   0   0   0   0]
 [  0 215   0   0   0   0   0   0   0   0]
 [  1   1 205   0   1   0   0   3   0   0]
 [  0   0   3 207   0   2   0   2   1   0]
 [  0   1   0   0 187   0   0   0   0   2]
 [  0   0   0   5   2 189   4   1   0   0]
 [  4   0   0   0   0   0 208   0   0   0]
 [  0   5   2   0   2   0   0 212   0   1]
 [  1   1   1   7   2   5   2   0 204   0]
 [  0   2   0   5   7   2   0   7   3 175]]
```

7. N = 4 and Train:Test = 60:40

   Accuracy Score = 0.9613095238095238

   Confusion Matrix

```
[[1609    0    2    0    0    1    3    1    0    1]
 [   0 1865    0    0    1    0    0    2    1    2]
 [  10   20 1641    7    2    1    3   24    2    3]
 [   3    9    8 1733    0   20    1    8   14    7]
 [   2   19    0    0 1590    0    4    1    0   26]
 [   5    6    1   41    1 1423   16    1    1   11]
 [  12    2    0    0    0    6 1591    0    1    0]
 [   1   33    2    3    4    0    0 1689    0   20]
 [   9   17   12   37    8   41    8    8 1434   14]
 [   9    6    1   18   35    7    1   38    6 1575]]
```

8. N = 4 and Train:Test = 70:30

   Accuracy Score = 0.964047619047619

   Confusion Matrix

```
[[1228    0    2    0    0    1    3    0    0    2]
 [   0 1365    0    0    0    0    1    2    1    1]
 [   4   12 1206    3    1    0    2   19    3    2]
 [   1    6    6 1320    0   16    1    7   10    2]
 [   1   12    0    0 1179    0    3    1    0   19]
 [   4    3    0   28    1 1075   14    1    1    5]
 [   9    0    0    0    0    4 1202    0    1    0]
 [   1   20    3    0    2    0    0 1283    0   17]
 [   6    9    6   24    9   29    3    6 1094   11]
 [   7    3    2   17   24    5    0   31    3 1195]]
```

9. N = 4 and Train:Test = 75:25

   Accuracy Score = 0.9640952380952381

   Confusion Matrix

```
[[1015    0    2    0    0    1    2    0    0    2]
 [   0 1126    0    0    0    0    1    1    1    1]
 [   4   12 1019    3    1    0    0   10    2    2]
 [   0    5    6 1086    0   12    1    5   10    3]
 [   1   11    0    0  980    0    2    1    0   19]
 [   3    2    0   22    1  888   13    0    1    4]
 [   5    2    0    0    0    6  995    0    0    0]
 [   1   16    2    0    2    0    0 1067    0   15]
 [   5    7    6   15    7   27    2    5  931    8]
 [   6    3    2   13   21    5    0   26    3 1016]]
```

10. N = 4 and Train:Test = 80:20

    Accuracy Score = 0.9635714285714285

    Confusion Matrix

```
[[815   0   1   0   0   1   2   0   0   2]
 [  0 896   0   0   0   0   0   1   1   1]
 [  4   9 830   1   1   0   0   9   2   2]
 [  0   4   4 879   0  11   1   5   8   1]
 [  0  10   0   0 759   0   2   1   0  19]
 [  3   1   0  16   1 724  11   1   1   4]
 [  4   1   0   0   1   4 798   0   0   0]
 [  0  12   2   0   2   0   0 854   0  10]
 [  3   5   4   9   5  24   1   4 729   5]
 [  6   3   0  13  20   6   0  18   3 810]]
```

11. N = 4 and Train:Test = 90:10

    Accuracy Score = 0.9657142857142857

    Confusion Matrix

```
[[388   0   0   0   0   0   0   0   0   1]
 [  0 456   0   0   0   0   0   0   0   1]
 [  3   2 430   0   0   0   0   6   0   0]
 [  0   3   3 419   0   3   0   2   4   0]
 [  0   4   0   0 395   0   1   0   0   7]
 [  2   0   0   5   1 361   7   1   0   3]
 [  2   0   0   0   0   3 423   0   0   0]
 [  0   5   1   0   2   0   0 408   0   5]
 [  3   2   1   5   3  12   1   3 376   3]
 [  1   2   0   8   8   6   0   8   1 400]]
```

12. N = 4 and Train:Test = 95:5

    Accuracy Score = 0.9619047619047619

    Confusion Matrix

```
[[209   0   0   0   0   0   0   0   0   1]
 [  0 215   0   0   0   0   0   0   0   0]
 [  2   1 205   0   0   0   0   3   0   0]
 [  0   1   2 206   0   2   0   2   2   0]
 [  0   1   0   0 185   0   0   0   0   4]
 [  1   0   0   2   1 189   5   1   0   2]
 [  1   0   0   0   0   0 211   0   0   0]
 [  0   5   1   0   2   0   0 211   0   3]
 [  1   0   1   4   2   6   1   1 207   0]
 [  0   2   0   4   6   3   0   3   1 182]]
```

13. N = 5 and Train:Test = 60:40

Accuracy Score = 0.9616071428571429

Confusion Matrix

```
[[1605    0    2    0    0    2    7    1    0    0]
 [   0 1863    0    0    0    0    3    2    1    2]
 [   8   22 1631    8    3    0    4   28    6    3]
 [   2    8   10 1722    0   30    2    9   13    7]
 [   3   18    0    0 1579    0    4    1    0   37]
 [   5    4    1   29    3 1433   21    1    1    8]
 [  11    2    0    0    0    6 1591    0    2    0]
 [   1   30    3    4    0    0    0 1690    0   24]
 [  12   17    8   28    9   37    7    6 1446   18]
 [   8   10    2   17   23    6    1   28    6 1595]]
```

14. N = 5 and Train:Test = 70:30

Accuracy Score = 0.9635714285714285

Confusion Matrix

```
[[1225    0    2    0    0    1    6    1    0    1]
 [   0 1365    0    0    0    0    1    2    1    1]
 [   4   14 1197    7    2    1    2   20    3    2]
 [   1    5    7 1315    0   19    0    9   10    3]
 [   2   10    0    0 1172    0    3    1    0   27]
 [   3    3    0   26    3 1073   18    1    1    4]
 [   7    1    0    0    0    4 1204    0    0    0]
 [   1   19    3    0    1    0    0 1281    0   21]
 [   7   10    5   18   10   26    4    2 1101   14]
 [   7    8    1   15   16    4    0   21    7 1208]]
```

15. N = 5 and Train:Test = 75:25

Accuracy Score = 0.9638095238095238

Confusion Matrix

```
[[1013    0    2    0    0    1    5    1    0    0]
 [   0 1126    0    0    0    0    1    1    1    1]
 [   4   12 1014    2    1    1    1   15    2    1]
 [   1    4    5 1082    0   15    0    7   10    4]
 [   1    9    0    0  974    0    3    1    0   26]
 [   3    3    0   21    3  884   14    0    3    3]
 [   4    1    0    0    0    5  998    0    0    0]
 [   1   15    2    0    1    0    0 1067    0   17]
 [   6    7    6   12   10   25    4    1  932   10]
 [   6    7    1   11   16    4    0   16    4 1030]]
```

16. N = 5 and Train:Test = 80:20

Accuracy Score = 0.9636904761904762

Confusion Matrix

```
[[815   0   1   0   0   1   3   1   0   0]
 [  0 896   0   0   0   0   0   1   1   1]
 [  4   9 827   1   1   1   0  13   0   2]
 [  1   3   3 874   0  16   0   7   7   2]
 [  1   8   0   0 758   0   2   1   0  21]
 [  2   2   0  16   3 722  13   0   2   2]
 [  4   1   0   0   1   3 799   0   0   0]
 [  0  11   2   0   1   0   0 853   0  13]
 [  5   5   4  10   7  20   2   2 727   7]
 [  6   6   1  11  13   3   0  12   3 824]]
```

17. N = 5 and Train:Test = 90:10

Accuracy Score = 0.9657142857142857

Confusion Matrix

```
[[388   0   0   0   0   0   0   1   0   0]
 [  0 456   0   0   0   0   0   0   0   1]
 [  3   2 425   0   2   0   0   8   1   0]
 [  1   3   2 418   0   4   0   3   2   1]
 [  0   3   0   0 392   0   0   0   0  12]
 [  1   0   0   5   1 360   9   1   1   2]
 [  2   0   0   0   0   2 424   0   0   0]
 [  0   4   1   0   1   0   0 408   0   7]
 [  3   1   0   5   4  11   0   1 380   4]
 [  1   3   0   8   6   3   0   6   2 405]]
```

18. N = 5 and Train:Test = 95:5

Accuracy Score = 0.9647619047619047

Confusion Matrix

```
[[209   0   0   0   0   0   0   1   0   0]
 [  0 215   0   0   0   0   0   0   0   0]
 [  2   1 203   0   1   0   0   3   1   0]
 [  1   1   1 206   0   2   0   2   1   1]
 [  0   0   0   0 185   0   0   0   0   5]
 [  0   0   0   2   1 190   6   1   0   1]
 [  1   0   0   0   0   0 211   0   0   0]
 [  0   4   1   0   1   0   0 212   0   4]
 [  1   0   0   3   3   7   0   0 208   1]
 [  0   2   0   5   3   1   0   1   2 187]]
```

19. N = 6 and Train:Test = 60:40

Accuracy Score = 0.9607738095238095

Confusion Matrix

```
[[1606    0    2    0    0    2    6    1    0    0]
 [   0 1864    0    0    1    0    2    1    1    2]
 [   8   27 1628    8    2    0    4   28    5    3]
 [   1   10   11 1727    1   21    2    8   12   10]
 [   3   24    0    0 1579    0    3    1    0   32]
 [   5    4    1   33    5 1429   18    1    2    8]
 [  11    2    0    0    0    5 1592    0    2    0]
 [   1   32    6    3    5    0    0 1686    0   19]
 [  11   17    9   30   11   27    8    8 1450   17]
 [   8   11    2   22   29    5    1   35    3 1580]]
```

20. N = 6 and Train:Test = 70:30

Accuracy Score = 0.9618253968253968

Confusion Matrix

```
[[1225    0    2    0    0    1    6    1    0    1]
 [   0 1365    0    0    0    0    1    2    1    1]
 [   4   15 1197    5    1    0    2   21    5    2]
 [   1    7    7 1315    1   16    1    8    9    4]
 [   1   13    0    0 1173    0    3    1    0   24]
 [   4    3    0   29    2 1070   17    1    1    5]
 [  11    1    0    0    0    3 1199    0    2    0]
 [   1   24    4    0    1    0    0 1278    0   18]
 [   9   12    4   19    9   25    7    4 1094   14]
 [   7    7    1   18   18    3    0   26    4 1203]]
```

21. N = 6 and Train:Test = 75:25

Accuracy Score = 0.9616190476190476

Confusion Matrix

```
[[1013    0    2    0    0    1    4    1    0    1]
 [   0 1127    0    0    0    0    0    1    1    1]
 [   4   13 1015    2    1    0    1   12    4    1]
 [   1    6    6 1083    0   14    1    6    8    3]
 [   1   12    0    0  974    0    2    1    0   24]
 [   3    3    0   25    2  881   14    1    1    4]
 [   8    2    0    0    0    2  995    0    1    0]
 [   1   19    2    0    1    0    0 1065    0   15]
 [   9   10    4   16    9   24    7    2  923    9]
 [   6    8    1   14   16    4    0   20    5 1021]]
```

22. N = 6 and Train:Test = 80:20

Accuracy Score = 0.9616666666666667

Confusion Matrix

```
[[815   0   1   0   0   1   2   1   0   1]
 [  0 896   0   0   0   0   0   1   1   1]
 [  4  12 825   1   1   0   0  10   3   2]
 [  1   5   4 874   0  14   1   6   6   2]
 [  1   9   0   0 759   0   1   1   0  20]
 [  2   2   0  20   2 718  12   1   2   3]
 [  7   1   0   0   0   1 798   0   1   0]
 [  0  15   2   0   1   0   0 852   0  10]
 [  6   7   3  13   7  18   4   2 723   6]
 [  6   6   1  10  16   4   0  14   4 818]]
```

23. N = 6 and Train:Test = 90:10

Accuracy Score = 0.9628571428571429

Confusion Matrix

```
[[388   0   0   0   0   0   0   1   0   0]
 [  0 456   0   0   0   0   0   0   0   1]
 [  3   2 426   0   1   1   0   8   0   0]
 [  1   3   2 419   0   3   0   3   2   1]
 [  1   3   0   0 395   0   0   0   0   8]
 [  1   0   0  11   1 353   9   2   1   2]
 [  3   0   0   0   0   2 423   0   0   0]
 [  0   7   1   0   2   0   0 405   0   6]
 [  3   1   0   8   5  10   3   1 375   3]
 [  1   3   0   7   8   3   0   7   1 404]]
```

24. N = 6 and Train:Test = 95:5

Accuracy Score = 0.96

Confusion Matrix

```
[[209   0   0   0   0   0   0   1   0   0]
 [  0 215   0   0   0   0   0   0   0   0]
 [  2   2 203   0   0   1   0   3   0   0]
 [  1   1   1 205   0   2   0   3   1   1]
 [  0   0   0   0 187   0   0   0   0   3]
 [  0   0   0   5   1 185   6   2   1   1]
 [  1   0   0   0   0   0 211   0   0   0]
 [  0   6   1   0   2   0   0 209   0   4]
 [  1   0   0   4   4   7   0   0 206   1]
 [  0   2   0   4   4   2   0   2   1 186]]
```

25. N = 7 and Train:Test = 60:40

Accuracy Score = 0.9585714285714285

Confusion Matrix

```
[[1600    1    2    0    0    3   10    1    0    0]
 [   0 1862    1    0    1    0    3    1    1    2]
 [   9   27 1618    9    2    1    5   34    5    3]
 [   2   11    9 1717    2   28    1   11   14    8]
 [   3   24    0    0 1563    0    4    1    0   47]
 [   4    5    2   25    3 1428   25    1    2   11]
 [   9    1    0    0    0    5 1594    0    3    0]
 [   1   34    3    1    1    0    0 1682    1   29]
 [  11   18    8   30   12   24    8    7 1450   20]
 [   9   13    1   20   25    5    1   29    3 1590]]
```

26. N = 7 and Train:Test = 70:30

Accuracy Score = 0.9625396825396826

Confusion Matrix

```
[[1222    0    2    0    0    2    8    1    0    1]
 [   0 1364    1    0    0    0    1    2    1    1]
 [   4   14 1192    5    1    0    3   27    4    2]
 [   2    7    6 1311    1   17    1   10   10    4]
 [   2   13    0    0 1167    0    3    1    0   29]
 [   3    3    0   18    1 1079   20    1    0    7]
 [   8    1    0    0    0    4 1201    0    2    0]
 [   1   23    3    0    1    0    0 1271    0   27]
 [   7   10    4   15   11   20    7    4 1106   13]
 [   7    7    1   18   14    3    0   18    4 1215]]
```

27. N = 7 and Train:Test = 75:25

Accuracy Score = 0.9614285714285714

Confusion Matrix

```
[[1011    0    2    0    0    1    6    1    0    1]
 [   0 1125    1    0    0    0    1    1    1    1]
 [   4   14 1007    2    1    0    2   18    4    1]
 [   1    6    5 1081    0   15    1    6    9    4]
 [   1   12    0    0  971    0    3    1    0   26]
 [   3    3    0   17    1  887   16    1    1    5]
 [   5    2    0    0    0    4  996    0    1    0]
 [   1   19    2    0    1    0    0 1058    0   22]
 [   7    9    4   13   10   19    7    3  930   11]
 [   7    7    1   14   14    3    0   18    2 1029]]
```

28. N = 7 and Train:Test = 80:20

Accuracy Score = 0.9613095238095238

Confusion Matrix

```
[[814   0   1   0   0   1   3   1   0   1]
 [  0 895   1   0   0   0   0   1   1   1]
 [  4  12 820   2   1   0   1  13   3   2]
 [  1   5   3 873   0  15   1   6   7   2]
 [  0   9   0   0 758   0   3   1   0  20]
 [  2   2   0   4   1 721  14   1   2   5]
 [  5   1   0   0   1   2 798   0   1   0]
 [  0  15   2   0   1   0   0 847   0  15]
 [  6   7   2  11   7  14   6   2 727   7]
 [  7   6   1  11  14   2   0  14   2 822]]
```

29. N = 7 and Train:Test = 90:10

Accuracy Score = 0.9647619047619047

Confusion Matrix

```
[[387   0   0   0   0   0   1   1   0   0]
 [  0 456   0   0   0   0   0   0   0   1]
 [  3   2 425   0   1   0   0   9   1   0]
 [  1   3   2 418   0   3   0   3   2   2]
 [  0   2   0   0 392   0   1   0   0  12]
 [  1   1   0   4   1 361   9   0   1   2]
 [  2   0   0   0   0   2 424   0   0   0]
 [  0   7   1   0   1   0   0 404   0   8]
 [  3   2   0   5   4   7   3   1 378   6]
 [  1   3   0   6   6   2   0   7   2 407]]
```

30. N = 7 and Train:Test = 95:5

Accuracy Score = 0.9619047619047619

Confusion Matrix

```
[[208   0   0   0   0   0   1   1   0   0]
 [  0 215   0   0   0   0   0   0   0   0]
 [  2   2 201   0   1   0   0   4   1   0]
 [  1   1   1 205   0   2   0   3   1   1]
 [  0   0   0   0 186   0   0   0   0   4]
 [  0   1   0   1   0 190   6   0   1   2]
 [  1   0   0   0   0   0 211   0   0   0]
 [  0   6   1   0   1   0   0 209   0   5]
 [  1   1   0   4   3   5   0   0 207   2]
 [  0   2   0   4   3   1   0   2   1 188]]
```

31. N = 10 and Train:Test = 60:40

Accuracy Score = 0.9578571428571429

Confusion Matrix

```
[[1602    1    1    0    0    3    9    1    0    0]
 [   0 1863    1    0    1    0    2    1    1    2]
 [   9   30 1616    9    3    1    5   32    5    3]
 [   2   11    9 1719    2   27    1   11   13    8]
 [   3   25    0    0 1571    0    5    2    0   36]
 [   4    8    1   25    5 1433   18    0    1   11]
 [  10    2    0    0    1    4 1592    0    3    0]
 [   1   34    4    0    3    0    0 1685    1   24]
 [  12   22    9   30   10   28    9    9 1433   26]
 [   9   13    3   23   25    4    1   33    7 1578]]
```

32. N = 10 and Train:Test = 70:30

Accuracy Score = 0.9595238095238096

Confusion Matrix

```
[[1223    0    2    0    0    2    8    1    0    0]
 [   0 1366    0    0    0    0    1    1    1    1]
 [   4   15 1187    6    2    1    5   26    3    3]
 [   1    8    8 1312    1   15    1   13    6    4]
 [   2   15    0    0 1165    0    4    2    0   27]
 [   3    5    0   21    2 1078   16    0    0    7]
 [  11    2    0    0    1    3 1198    0    1    0]
 [   1   26    4    0    1    0    0 1270    0   24]
 [   9   13    5   23   11   20    6    3 1091   16]
 [   9   10    4   17   15    2    0   27    3 1200]]
```

33. N = 10 and Train:Test = 75:25

Accuracy Score = 0.9591428571428572

Confusion Matrix

```
[[1011    0    2    0    0    1    7    1    0    0]
 [   0 1127    0    0    0    0    1    0    1    1]
 [   4   16 1003    4    2    0    2   17    4    1]
 [   1    7    6 1080    1   13    1    9    6    4]
 [   1   13    0    0  971    0    3    1    0   25]
 [   3    3    0   20    2  884   16    1    1    4]
 [   7    2    0    0    1    3  994    0    1    0]
 [   1   21    3    0    0    0    0 1061    0   17]
 [   8   10    4   17   11   22    6    2  918   15]
 [   7    8    1   14   13    2    0   26    2 1022]]
```

34. N = 10 and Train:Test = 80:20

Accuracy Score = 0.9586904761904762

Confusion Matrix

```
[[814   0   1   0   0   1   4   1   0   0]
 [  0 896   0   0   0   0   0   1   1   1]
 [  4  11 818   3   2   0   1  15   3   1]
 [  1   6   4 872   1  12   1   8   6   2]
 [  0  10   0   0 756   0   3   1   0  21]
 [  2   2   0  16   2 719  15   1   1   4]
 [  6   1   0   0   1   2 797   0   1   0]
 [  0  16   2   0   0   0   0 849   0  13]
 [  5   9   2  13   7  15   7   2 718  11]
 [  7   7   1  12  14   3   0  19   2 814]]
```

35. N = 10 and Train:Test = 90:10

Accuracy Score = 0.9623809523809523

Confusion Matrix

```
[[386   0   0   0   0   0   2   1   0   0]
 [  0 456   0   0   0   0   0   0   0   1]
 [  3   4 425   0   1   0   0   8   0   0]
 [  1   4   2 415   0   5   0   4   2   1]
 [  0   3   0   0 394   0   1   0   0   9]
 [  1   1   0   6   1 358  10   1   0   2]
 [  2   0   0   0   0   1 424   0   1   0]
 [  0   8   1   0   0   0   0 407   0   5]
 [  3   2   0   7   3  10   4   1 372   7]
 [  1   3   0   8   6   3   0   7   1 405]]
```

36. N = 10 and Train:Test = 95:5

Accuracy Score = 0.9604761904761905

Confusion Matrix

```
[[208   0   0   0   0   0   1   1   0   0]
 [  0 215   0   0   0   0   0   0   0   0]
 [  2   2 203   0   0   0   0   4   0   0]
 [  1   2   1 204   0   2   0   3   1   1]
 [  0   0   0   0 187   0   0   0   0   3]
 [  0   1   0   3   1 187   7   1   0   1]
 [  1   0   0   0   0   0 211   0   0   0]
 [  0   6   1   0   0   0   0 212   0   3]
 [  1   0   0   5   3   7   1   0 203   3]
 [  0   2   0   4   4   1   0   2   1 187]]
```

Conclusion

The performance of the KNN model depends significantly on two things that are:-

1. Training-Testing split – The more data available for training, the better the KNN model can learn the patterns in the data. However, if the training set is too large relative to the testing set, we might not get a good accuracy score, which suggests our model does not learn on the data.csv file. If the training set is too small, the KNN model might not learn the underlying patterns well, leading to poor performance. A small training set can lead to a high-variance model, where the model's performance can vary significantly with different training data. An ample training set with a tiny testing set can lead to a high-bias estimate of model performance, where the performance on the testing set may not be representative of the performance of test data of the data.csv file.

2. The value of K in K nearest-neighbours – As the Training-Testing split is the basic standard, we should remember that the K value plays a minor role here. If the k value is small, it becomes sensitive to might the nearest neighbours but can also predict wrong if the closest neighbour is a wrong case. If the k value is large, it favours the majority labels in the dataset, becoming highly biased.