

Machine Learning

Q 2 E

1 c :

For k= 1

55.46218487394958

For k= 3

62.18487394957983

For k= 5

62.18487394957983

For k= 7

61.34453781512605

For k= 9

63.86554621848739

For k= 11

59.66386554621849

1 d :

For k= 1

55.08474576271186

For k= 3

62.39316239316239

For k= 5

62.93103448275862

For k= 7

62.60869565217392

For k= 9

63.1578947368421

For k= 11

61.94690265486725

As we compare values keeps changing with respect to value of K and columns we train.

In 2 c : Accuracy is 71.68141592920354% and when we remove the age parameter the accuracy changes to 70.53571428571429%

When we compare the values of 1 c & d with 2 c & d, Naïve Bayes gives more accuracy. When we do prediction with out considering the age parameter the accuracy reduces.

Outputs

1 a & b:

Cartesian Distance

For k = 1

```

enter 1 for k=1, 2 K=3, 3 k=71
For k =1
enter 1 for Manhattan distance/n enter 1 for Manhattan distance/n enter 1 for Manhattan distance1
Data=[1.62065758929, 59.376557437583, 32], Predicted: W
Data=[1.7793983848363, 72.071775670801, 36], Predicted: W
Data=[1.7004576585974, 66.267508112786, 31], Predicted: W
Data=[1.6591086215159, 61.751621901787, 29], Predicted: W

```

For k = 3

```

enter 1 for k=1, 2 K=3, 3 k=72
For k =3
enter 1 for Manhattan distance/n enter 1 for Manhattan distance/n enter 1 for Manhattan distance1
Data=[1.62065758929, 59.376557437583, 32], Predicted: W
Data=[1.7793983848363, 72.071775670801, 36], Predicted: M
Data=[1.7004576585974, 66.267508112786, 31], Predicted: W
Data=[1.6591086215159, 61.751621901787, 29], Predicted: W

```

For k = 7

```

enter 1 for k=1, 2 K=3, 3 k=73
For k =7
enter 1 for Manhattan distance/n enter 1 for Manhattan distance/n enter 1 for Manhattan distance1
Data=[1.62065758929, 59.376557437583, 32], Predicted: W
Data=[1.7793983848363, 72.071775670801, 36], Predicted: M
Data=[1.7004576585974, 66.267508112786, 31], Predicted: W
Data=[1.6591086215159, 61.751621901787, 29], Predicted: W

```

Manhattan

For k = 1

```

enter 1 for k=1, 2 K=3, 3 k=71
For k =1
enter 1 for Manhattan distance/n enter 1 for Manhattan distance/n enter 1 for Manhattan distance2
Data=[1.62065758929, 59.376557437583, 32], Predicted: W
Data=[1.7793983848363, 72.071775670801, 36], Predicted: W
Data=[1.7004576585974, 66.267508112786, 31], Predicted: W
Data=[1.6591086215159, 61.751621901787, 29], Predicted: W

```

For k = 3

```

enter 1 for k=1, 2 K=3, 3 k=72
For k =3
enter 1 for Manhattan distance/n enter 1 for Manhattan distance/n enter 1 for Manhattan distance2
Data=[1.62065758929, 59.376557437583, 32], Predicted: W
Data=[1.7793983848363, 72.071775670801, 36], Predicted: M
Data=[1.7004576585974, 66.267508112786, 31], Predicted: W
Data=[1.6591086215159, 61.751621901787, 29], Predicted: W

```

For k = 7

```

enter 1 for k=1, 2 K=3, 3 k=73
For k =7
enter 1 for Manhattan distance/n enter 1 for Manhattan distance/n enter 1 for Manhattan distance2
Data=[1.62065758929, 59.376557437583, 32], Predicted: W
Data=[1.7793983848363, 72.071775670801, 36], Predicted: M
Data=[1.7004576585974, 66.267508112786, 31], Predicted: W
Data=[1.6591086215159, 61.751621901787, 29], Predicted: W

```

Minkowski

For k = 1

```

enter 1 for k=1, 2 K=3, 3 k=71
For k =1
enter 1 for Manhattan distance/n enter 1 for Manhattan distance/n enter 1 for Manhattan distance3
14.373
13.251
14.165
16.701
21.685
5.635
10.031
17.687
5.259
7.583
6.707
18.085
17.254
4.971
Data=[1.62065758929, 59.376557437583, 32], Predicted: W
12.001
2.013
3.113
6.541
9.800
8.344
6.175
6.294
9.962
9.784
6.983
5.473
4.435
8.124
Data=[1.7793983848363, 72.071775670801, 36], Predicted: W
8.577
6.561
7.322
9.804
14.790
1.506
3.147
10.795
6.602
4.000
0.216
11.258
10.869
4.243
Data=[1.7004576585974, 66.267508112786, 31], Predicted: W
11.447
11.203
11.940
14.318
19.305
3.207
7.635
15.322
8.000
4.534
4.463
15.869
15.485
6.097
Data=[1.6591086215159, 61.751621901787, 29], Predicted: W

```

For k = 3

```

enter 1 for k=1, 2 K=3, 3 k=72
For k =3
enter 1 for Manhattan distance/n enter 1 for Manhattan distance/n enter 1 for Manhattan distance3
14.373
13.251
14.165
16.701
21.685
5.635
10.031
17.687
5.259
7.583
6.707
18.085
17.254
4.971
Data=[1.62065758929, 59.376557437583, 32], Predicted: W
12.001
2.013
3.113
6.541
9.800
8.344
6.175
6.294
9.962
9.784
6.983
5.473
4.435
8.124
Data=[1.7793983848363, 72.071775670801, 36], Predicted: M
8.577
6.561
7.322
9.804
14.790
1.506
3.147
10.795
6.602
4.000
0.216
11.258
10.869
4.243
Data=[1.7004576585974, 66.267508112786, 31], Predicted: W
11.447
11.203
11.940
14.318
19.305
3.207
7.635
15.322
8.000
4.534
4.463
15.869
15.485
6.097
Data=[1.6591086215159, 61.751621901787, 29], Predicted: W

```

For k = 7

```

enter 1 for k=1, 2 K=3, 3 k=73
For k =7
enter 1 For Manhattan distance/n enter 1 For Manhattan distance/n enter 1 For Manhattan distance3
14.373
13.251
14.165
16.701
21.685
5.635
10.031
17.687
5.259
7.583
6.707
18.085
17.254
4.971
Data=[1.62065758929, 59.376557437583, 32], Predicted: W
12.001
2.013
3.113
6.541
9.800
8.344
6.175
6.294
9.962
9.784
6.983
5.473
4.435
8.124
Data=[1.7793983848363, 72.071775670801, 36], Predicted: M
8.577
6.561
7.322
9.804
14.790
1.506
3.147
10.795
6.602
4.000
0.216
11.258
10.869
4.243
Data=[1.7004576585974, 66.267508112786, 31], Predicted: W
11.447
11.203
11.940
14.318
19.305
3.207
7.635
15.322
8.000
4.534
4.463
15.869
15.485
6.097
Data=[1.6591086215159, 61.751621901787, 29], Predicted: W

```

1 c:

```

For k= 1
55.46218487394958
For k= 3
62.18487394957983
For k= 5
62.18487394957983
For k= 7
61.34453781512605
For k= 9
63.86554621848739
For k= 11
59.66386554621849

```

1 d:

```
For k= 1
55.08474576271186
For k= 3
62.39316239316239
For k= 5
62.93103448275862
For k= 7
62.60869565217392
For k= 9
63.1578947368421
For k= 11
61.94690265486725
```

2 a and b:

```
[M] => 0
[W] => 1
Data=[1.62065758929, 59.376557437583, 32], predicted: 1
Data=[1.7793983848363, 72.071775670801, 36], predicted: 1
Data=[1.7004576585974, 66.267508112786, 31], predicted: 1
Data=[1.6591086215159, 61.751621901787, 29], predicted: 1
```

2 c:

```
[M] => 0
[W] => 1
71.68141592920354
```

2 d:

70.53571428571429