CS451 Report – Assignment 3

Barış Özbaş – S014669

Definition

In this assignment I have implemented KNearestNeighborClassifier, NaïveBayesClassifier and DecisionTreeClassifier to solve HDR Classification Problem. This report will be about the internal parameters of the algorithms, and the test and validation results.

K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions).

A Naive Bayes classifier is a probabilistic machine learning model that's used for classification task. The crux of the classifier is based on the Bayes theorem.

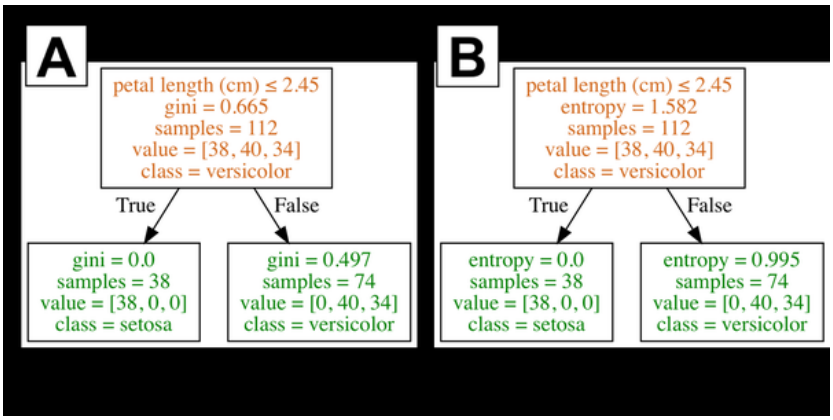
A Decision Tree is a simple representation for classifying examples. It is a Supervised Machine Learning where the data is continuously split according to a certain parameter.

Approach

First I decided to use MNIST dataset, I could be able to gather that, with a function of sklearn library. After implementing the algorithms, I needed to find some internal parameters to tweak and find a best performing version of the algorithm. Below, the parameters that I have chosen can be found.

K nearest neighbor's parameters: n_neighbors parameter is number of neighbors to use by default for kneighbors queries that is defined in sklearn documentation. The reason that I have chosen to test this parameter out, is that it is one of the biggest factors that effects the algorithm's runtime, due to the K nearest neighbor classification taking too much time, I have researched and found a technique to reduce data and by that, lower the time it takes to complete the algorithm, which is PCA (Principal component analysis) [2].

Decision Tree parameters: Criterion parameter is the function to measure the quality of a split. Supported criteria are "gini" for the Gini impurity and "entropy" for the information gain. The difference can be seen on the Figure [1].



Figure[1]: Gathered from https://www.quora.com/What-is-difference-between-Gini-Impurity-and-Entropy-in-Decision-Tree

Splitter parameter is the strategy used to choose the split at each node. Supported strategies are “best” to choose the best split and “random” to choose the best random split. This is how the decision tree searches the features for a split. The default value is set to “best”. That is, for each node, the algorithm considers all the features and chooses the best split. If you decide to set the splitter parameter to “random,” then a random subset of features will be considered. The split will then be made by the best feature within the random subset [1].

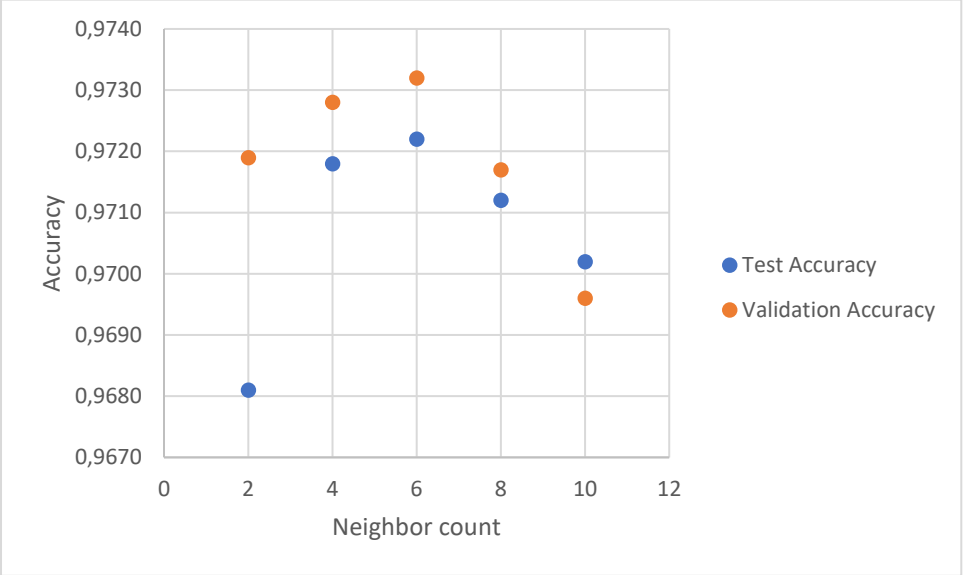
Naive Bayes classifier parameters: There were just a couple of parameters that can be changed, variable smoothing parameter would be useful if we had a data that is too unconsistable and the data type was float, in this case we do not need to test Naive Bayes classifier with different parameters because that would not change the result in an effective/meaningful way.

Results

KNearestNeighbourClassifier Results:

Accuracy of the algorithm regarding to different Neighbour count parameter.

Neighbor count	2	4	6	8	10
Test Accuracy	0.9681	0.9718	0.9722	0.9712	0.9702
Validation A.	0.9719	0.9728	0.9732	0.9717	0.9696



As it can be seen, with n_count = 6 is best performed. Below, confusion matrix and accuracy for each classification of the best performing test can be found.

	precision	recall	f1-score	support
0	0.98	0.99	0.98	519
1	0.96	1.00	0.98	671
2	0.98	0.96	0.97	566
3	0.96	0.98	0.97	561
4	0.97	0.96	0.97	584
5	0.98	0.96	0.97	517
6	0.98	0.99	0.98	540
7	0.97	0.98	0.97	544
8	0.99	0.96	0.98	533
9	0.96	0.95	0.95	565
accuracy			0.97	5600
macro avg	0.97	0.97	0.97	5600
weighted avg	0.97	0.97	0.97	5600

```
[[512   1   0   0   0   1   4   0   0   1]
 [  0 668   2   0   0   0   0   1   0   0]
 [  2   8 542   4   1   0   0   5   2   2]
 [  1   1   4 548   1   1   0   2   1   2]
 [  0   7   0   0 563   0   1   1   1  11]
 [  2   0   0   9   1 496   7   0   1   1]
 [  1   1   1   0   0   1 536   0   0   0]
 [  0   5   1   0   0   0   0 535   0   3]
 [  1   4   0   4   2   5   1   0 514   2]
 [  2   0   1   3  11   1   0  10   1 536]]
```

NaiveBayesClassifier Results:

Naïve Beyes' accuracy was 0.8597. Below, confusion matrix and accuracy for each classification of the best and only performing test of this classifier can be found.

	precision	recall	f1-score	support
0	0.95	0.92	0.94	1354
1	0.98	0.94	0.96	1547
2	0.71	0.86	0.77	1405
3	0.82	0.81	0.82	1495
4	0.89	0.84	0.86	1310
5	0.81	0.84	0.82	1302
6	0.95	0.88	0.91	1374
7	0.90	0.82	0.86	1434
8	0.85	0.85	0.85	1334
9	0.80	0.82	0.81	1445
accuracy			0.86	14000
macro avg	0.86	0.86	0.86	14000
weighted avg	0.87	0.86	0.86	14000

```
[[1252    0   46    4    1   27   16    1    6    1]
[   1 1447   13   12   15   17    6   16   13    7]
[  11    2 1205   67   15   14   17   14   52    8]
[   6    4   79 1218    3   62    7   34   47   35]
[   0    1   59    3 1096    6    5    5    9  126]
[   9    0   35   99    4 1094    9    7   25   20]
[  11    2   78    2    7   53 1213    0    7    1]
[   6   20   56   12   28   30    3 1182   20   77]
[  13    0   56   57    5   31    5    5 1139   23]
[   8    2   81   11   62   23    0   50   17 1191]]
```

DecisionTreeClassifier Results:

Accuracy of the algorithm regarding to different splitter and criterion parameters (first value is the test accuracy and the second one is the validation accuracy.

	random	best
gini	0.8002 / 0.8039	0.8265 / 0.8251
entropy	0.8103 / 0.8278	0.8378 / 0.8435

As it can be seen, best performing test for Decision Tree implementation is gathered with criterion = entropy and splitter = best. Below, confusion matrix and accuracy for each classification of the best performing test of decision tree can be found.

	precision	recall	f1-score	support
0	0.90	0.89	0.90	1354
1	0.95	0.96	0.95	1547
2	0.85	0.83	0.84	1405
3	0.80	0.78	0.79	1495
4	0.79	0.82	0.80	1310
5	0.78	0.76	0.77	1302
6	0.88	0.90	0.89	1374
7	0.86	0.87	0.86	1434
8	0.76	0.77	0.77	1334
9	0.79	0.78	0.78	1445
accuracy			0.84	14000
macro avg	0.84	0.84	0.84	14000
weighted avg	0.84	0.84	0.84	14000

```
[[1211    2   16   21    8   28   30    8   20   10]
 [   1 1482   12    6    3    9    5    4   16    9]
 [  17    7 1171   51   22   17   28   26   52   14]
 [  20   11   58 1171   11   74   21   31   71   27]
 [   4    6   17   12 1070   13   18   28   23  119]
 [  37    2   17   82   20  990   38   15   71   30]
 [  28    5   16   10   15   36 1236    4   15    9]
 [   6   14   16   17   40   15    3 1246   18   59]
 [  16   20   45   77   22   63   18   21 1027   25]
 [  10   11   15   21  138   24    3   66   31 1126]]
```

Conclusion and comparison

Best of the performed accuracy values per each test can be compared as:

KNearestNeighbour (0.9722) > NaiveBayes (0.8597) > DecisionTree (0.8378)

It can be seen clearly that KNearestNeighbour did the best amongst all. But it also took much higher time to solve the problem. While KNearestNeighbour took nearly 4 minutes to finish one set of classifications, the other two didn't take more than 4 seconds to complete.

In conclusion, I have implemented and tested out KNearestNeighbour, NaiveBayes, DecisionTree algoritms using sklearn library to solve HDR Classification Problem, found possible parameters that may affect the performance, tested, validated and finally compared them.

References

[1] Ceballos (February, 2019). Scikit-Learn Decision Trees Explained. Gathered from: https://towardsdatascience.com/scikit-learn-decision-trees-explained-803f3812290d

[2] Shlens (December, 2005). A Tutorial on Principal Component Analysis. Gathered from: https://www.cs.cmu.edu/~elaw/papers/pca.pdf