

# Neural Networks

Erika Koro, 9707

October 13, 2022

## Introduction

The objective of this assignment is to compare the KNearest Neighbours classifier with Nearest Centroid classifier for the MNIST dataset. For the implementation of this comparison, the pipenv environment was used as well as the the sklearn.neighbors library for the KNeighborsClassifier() and NearestCentroid() classes. The used python version is Python 3.10.8. The code for this assignment can be found in [here](#).

## KNN Classifier, k=1

Executing the code for knn\_classifier with the parameter of neighbours set as 1 gives the following performance results.

	precision	recall	f1-score	support
0	0.98	0.99	0.99	980
1	0.97	0.99	0.98	1135
2	0.98	0.96	0.97	1032
3	0.96	0.96	0.96	1010
4	0.97	0.96	0.97	982
5	0.95	0.96	0.96	892
6	0.98	0.99	0.98	958
7	0.96	0.96	0.96	1028
8	0.98	0.94	0.96	974
9	0.96	0.96	0.96	1009
accuracy			0.97	10000
macro avg	0.97	0.97	0.97	10000
weighted avg	0.97	0.97	0.97	10000
Accuracy: 0.9691				
--- 19.435083389282227 seconds ---				

Figure 1: knn classifier with k=1

Except for the mean accuracy, in order to assess KNN classifier's performance, the classification report offers more assessment metrics, such as the precision, the recall, and f1-score. Specifically, precision metric represents the accuracy of a number's prediction, recall metric is the ability of a classifier to find all the values with that specific value. For instance, in the picture we notice that the recall for the label 0 is 0.99. This means that it found 99 zeros of the total number of zeros = 100. Finally, F1 score is about how many of the label's predictions were correct.

## KNN Classifier, k=3

Executing the code for `knn_classifier` with the parameter of neighbours set as 3, gives the following performance results.

	precision	recall	f1-score	support
0	0.97	0.99	0.98	980
1	0.96	1.00	0.98	1135
2	0.98	0.97	0.97	1032
3	0.96	0.97	0.96	1010
4	0.98	0.97	0.97	982
5	0.97	0.96	0.96	892
6	0.98	0.99	0.98	958
7	0.96	0.96	0.96	1028
8	0.99	0.94	0.96	974
9	0.96	0.96	0.96	1009
accuracy			0.97	10000
macro avg	0.97	0.97	0.97	10000
weighted avg	0.97	0.97	0.97	10000
Accuracy: 0.9705				
--- 37.29514527320862 seconds ---				

Figure 2: knn classifier with k=3

## Nearest Centroid

Executing the code for nc\_model gives the following performance results.

	precision	recall	f1-score	support
0	0.91	0.90	0.90	980
1	0.77	0.96	0.86	1135
2	0.88	0.76	0.81	1032
3	0.77	0.81	0.78	1010
4	0.80	0.83	0.81	982
5	0.75	0.69	0.72	892
6	0.88	0.86	0.87	958
7	0.91	0.83	0.87	1028
8	0.79	0.74	0.76	974
9	0.77	0.81	0.79	1009
accuracy			0.82	10000
macro avg	0.82	0.82	0.82	10000
weighted avg	0.82	0.82	0.82	10000
Accuracy: 0.8203				
--- 0.4001777172088623 seconds ---				

Figure 3: Nearest Centroid classifier

## Conclusion

As it is easily observed from the above results, we conclude that the KNN Classifier has bigger accuracy(0.97) than Nearest Centroid(0.82). But, the execution time for the NC model is much lower than KNN's.

## Bibliography

<https://muthu.co/understanding-the-classification-report-in-sklearn/>

<https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>

<https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.NearestCentroid.html>