# Machine Learning - Coursework Part A

Charalambos Tsiouts

March 21, 2023

## 1 Description

In this assignment we initially constructed a kNN classification method to predict the labels of the Iris dataset and then we used a nested cross-validation technique to evaluate the performance of our machine learning model. For the evaluation of our model we used different hyperpameters such as distance metric, number of nearest neighbors and number of bins. We calculated the performance of the model in terms of accuracy, confusion matrix and error.

For the **kNN classification** method we split the iris dataset to a training and testing set. We used the training set to find the nearest neighbors for each test point for a given distance metric and assigned the labels of the neighbors accordingly. We used the assigned labels to calculate the accuracy for our model by comparing them with the test labels. Next we implemented **nested cross-validation** to evaluate our model. Our function split the data into 5 bins and for each fold, we used one bin as the testing set, another bin as the validation set and the remaining bins as the training set. We then looped through all possible combinations of the number of neighbors and distance metrics and trained the kNN model on the training set and tested it on the validation set to find the best set of parameters that gave the highest accuracy. Once we found the optimal parameters, we merged the training and validation sets and trained the kNN model on the entire dataset using the best parameters. Finally, we evaluated the accuracy of our model on the test set and repeated the process for all 5 folds.

In the evaluation of our model we used Euclidean, Manhattan and Chebychev distance. We defined them using the generalized Minkowski metric. The Minkowski distance of order $p$ between two points $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ in a $n$-dimensional space is defined as:

$$\left( \sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}}$$

,where $p \geq 1$ is the order of the Minkowski distance. When $p = 1$, the Minkowski distance reduces to the **Manhattan** distance, when $p = 2$, it reduces to the **Euclidean** distance and when $p = \inf$ it reduces to **Chebychev** distance.

## 2 Results

**Table 1: Results for the clean dataset**

| Fold | Accuracy | # Neighbors | Distance |
|---|---|---|---|
| 1 | 0.97 | 3 | Euclidean |
| 2 | 0.93 | 4 | Euclidean |
| 3 | 0.93 | 1 | Euclidean |
| 4 | 1.00 | 4 | Euclidean |
| 5 | 0.97 | 1 | Euclidean |
| **Total** | **0.96 ± 0.02** | | |

**Table 2: Results for the noisy dataset**

| Fold | Accuracy | # Neighbors | Distance |
|---|---|---|---|
| 1 | 0.97 | 2 | Chebychev |
| 2 | 0.93 | 9 | Euclidean |
| 3 | 0.93 | 2 | Euclidean |
| 4 | 1.00 | 5 | Euclidean |
| 5 | 0.97 | 8 | Euclidean |
| **Total** | **0.87 ± 0.07** | | |

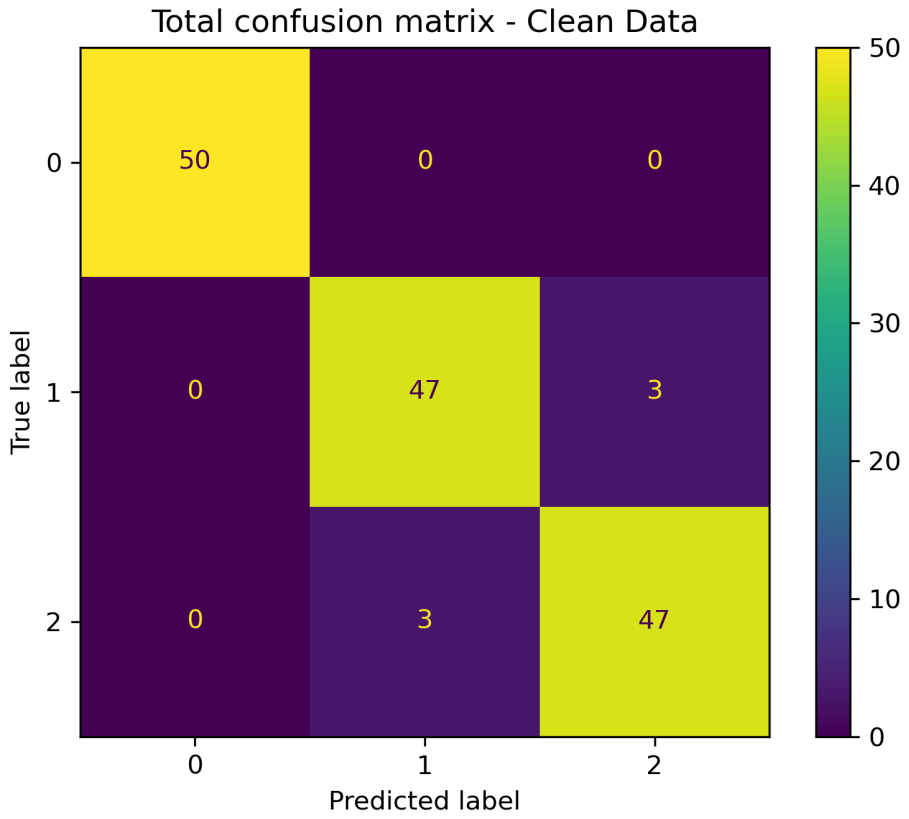


**Figure 1: Total confusion matrix for the clean data over 5-fold cross validation for each of the runs of cross-validation**

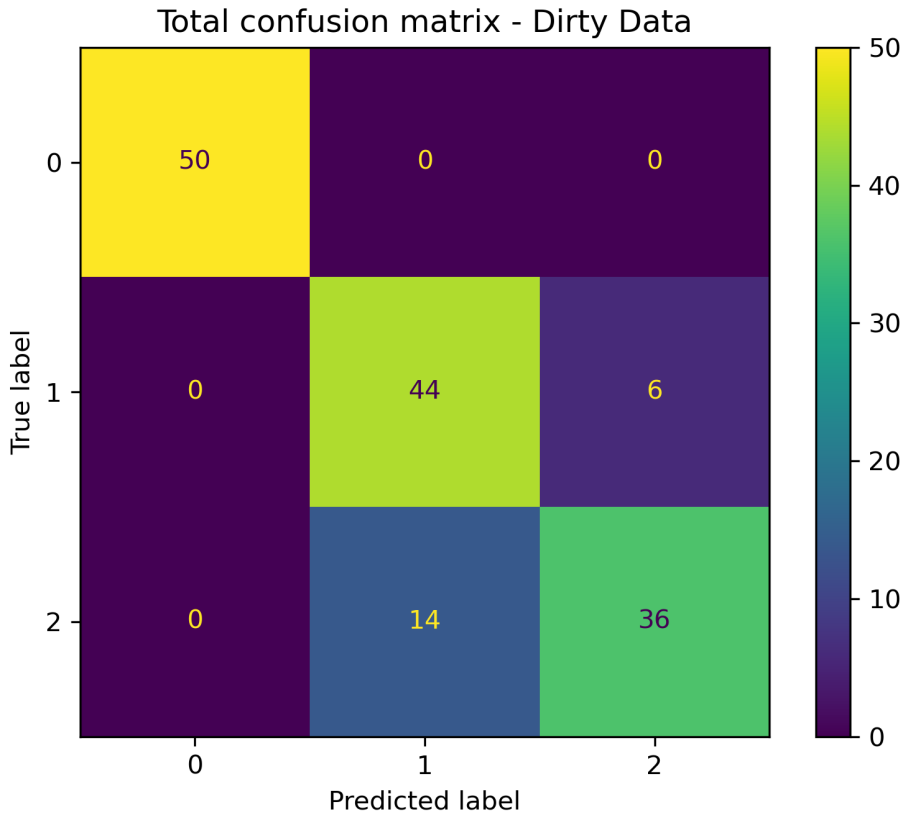


**Figure 2: Total confusion matrix for the noisy data over 5-fold cross validation for each of the runs of cross-validation**

### 2.1 Results Analysis

After implementing the KNN classifier and nested cross-validation algorithm, we observed a significant decrease in the accuracy of the models. This result was not surprising given the visual representation of the data, as the noisy data appeared less distinct between the species labels.

In terms of the optimal parameters, we found that the best number of k neighbors changed for each fold in both clean and noisy datasets. Euclidean distance was optimal for the majority of the folds for both datasets, while Chebychev distance was preferable for one fold in the noisy data.

Overall we observe that the best parameters, in terms of number of neighbours and distance are different for each dataset. Based on our model, we conclude that we cannot determine optimal parameters regardless of the dataset.

## 3 Questions

Consider a scenario where we are using k-NN with k = 5. Assume that the 5 nearest neighbours of a test sample (1, 1) are the data points (2, 2), (1, 4), (5, 5), (5, 7),(5, 8) with corresponding labels (0, 0, 1, 1, 1). Note that we are using the L1 norm (sum of absolute values) to measure the distance.

*(a) What are the distances of the test sample to each of the neighbouring samples?*
**Answer:**

- d((1,1),(2,2)) = $|1-2|+|1-2|=2$
- d((1,1),(1,4)) = $|1-1|+|1-4|=3$

- d((1,1),(5,5)) = $|1-5|+|1-5|=8$
- d((1,1),(5,7)) = $|1-5|+|1-7|=10$
- d((1,1),(5,8)) = $|1-5|+|1-8|=11$

*(b) Which label does the algorithm assign when using majority voting? Do you think that is the right decision given the neighbours above?*

**Answer:**

The algorithm will assign label 1, since the majority of the neighbors have label 1. Given that the neighbors with the smallest distance belong to label 0 but they do not have the majority, the algorithm does not choose the right label.

*(c) Can you think of any modifications to the k-NN algorithm that would allow a better classification in this case? Write the pseudocode for this modification and show how the resulting algorithm changes the classification result on the given example.*

**Answer:**

We can modify our code by assigning labels based on the distance of the neighbours not the majority. We can weight the distance of the neighbours as shows in the pseudocode.

```
#Given the k-neighbours n and their distance d
v0 = 0
v1 = 1
loop over the neighbours n:
    if i neighbour label is 0, v0 += 1/d_i
    else if i label is 1 , v1 += 1/d_i
if v0>v1 assign label 0
else label is 1
```

Now based on the pseudocode:

- v0 = $\frac{1}{2}+\frac{1}{3}=0.83$
- v1 = $\frac{1}{8}+\frac{1}{10}+\frac{1}{11}=0.32$

Therefore the label is 1.

*(d) Implement the above modification as an option for the k-NN algorithm you developed. You do not have to evaluate this extensively - you can e.g. use the best combination of parameters you have found for clean and noisy data. Discuss any changes in the results.*

**Answer:**

The implementation of the code can be found on the attached notebook. We gave the weighted assignment as an option in our kNN function. We also implemented another function that assigned labels based on the distance and also kept the majority assign function. For the given dataset we did not had any change for the clean data but we had a slight reduction of the accuracy for the noisy data. We can conclude that the weighted assignment will not work for every dataset.