D2 Implementation of K-Nearest Neighbours

Manohar Kowthavarapu (11554625)

1. **Create a table with accuracy for K=1, K=3, K=5, K=100 and K=500**
    - To calculate the accuracy score I've used the **accuracy_score** method from the sklearn library which takes y_test data and prediction data as arguments and returns the **accuracy score**(**acc_score**).

```
prediction = calculate_knn(x_train, y_train, x_test, k)
acc_score = accuracy_score(y_test, prediction)
```

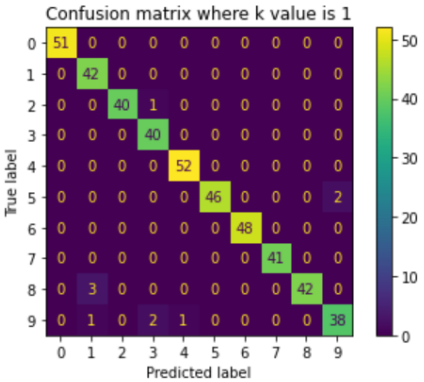
    - Created the accuracy table by changing the k values to 1, 3, 5, 100 and 500, from the below table we can see that for the K value 1 and 3 we've got good accuracy score.

	k	accuracy_score
0	1	0.977778
1	3	0.977778
2	5	0.975556
3	100	0.906667
4	500	0.724444

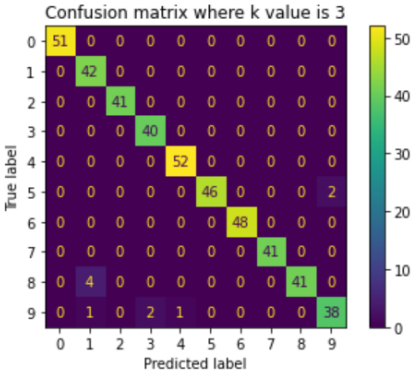
2. **Classification Matrix for each K value**
    - For creating the classification matrix I've used the **confusion_matrix** method from the **sklearn** library, this takes y_test data and prediction data as arguments and returns the **confusion_matrix_object**
    - Using the **confusion_matrix_object** we need to create the **display** object using **ConfusionMatrixDisplay** method
    - With the display object I've plotted the confusion matrix

```
confusion_matrix_object = metrics.confusion_matrix(y_test, prediction)
display = metrics.ConfusionMatrixDisplay(confusion_matrix_object)
display.plot()
display.ax_.set_title(f"Confusion matrix where k value is {k}")
```

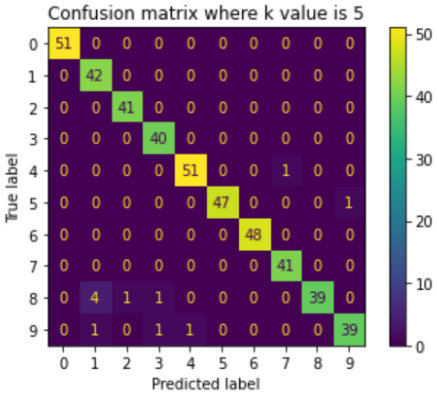
    - Below are the diagrams which shows the confusion matrix for all the k values which we considered while creating the accuracy score table.
        1. Confusion matrix for K=1



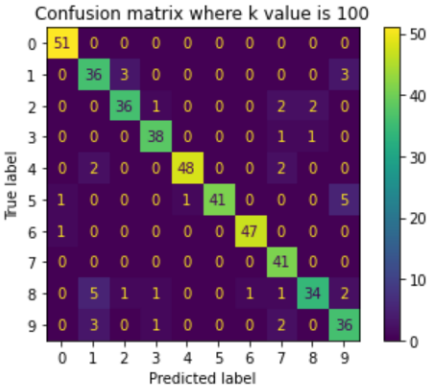
2. Confusion matrix for K=3



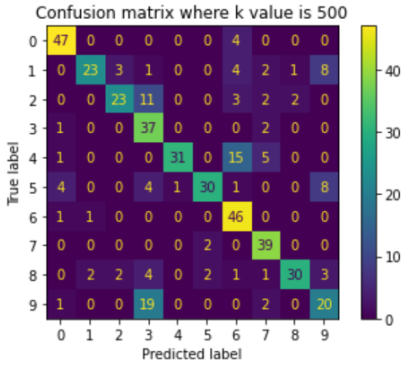
3. Confusion matrix for K=5



4. Confusion matrix for K=100



5. Confusion matrix for K=500



3. **Note which errors are more common, In what way does that match your intuitions.**
    - From the confusion matrix if we see that more errors are occurred when the k value is 500.
    - Also, we can observe that number “9” and “4” are predicted wrongly with the number “3” and “4”, this is because of the hand written digits will be more closely to those numbers. Hence the model is underfit when the k value is 500
    - Finally, we see the model is overfit when the k value is “1” and “3”