Machine Learning A (2023)
Home Assignment 1

Alexander Husted | wqg382

Contents

1 Make Your Own (10 points) **2**

2 Digits Classification with K Nearest Neighbors (40 points) **3**
2.1 Task #1 . . . 3
2.2 K-nearest function . . . 3
2.3 Validation error with $n \in \{10, 20, 40, 80\}$ . . . 4
2.4 Variance . . . 4
2.5 Questions . . . 5

3 Task 2 **6**
3.1 Plots . . . 6
3.2 Answers . . . 8

1 Make Your Own (10 points)

What profile information would you collect and what would be the sample space X?
I would get the average grade of each student in mathematical and computer science courses. So that I would have two features "Average grade in math" and "Average grade in data science".

$$X = \mathbb{R} \cdot \mathbb{R} = \mathbb{R}^2$$

What would be the label space Y?
Y is the final grade of a student in Machine Learning A.

$$Y = \mathbb{N}$$

How would you define the loss function?
For the loss function, I would use the absolute loss. To get the distance between the predicted and actual value.

$$l(y', y) = |y' - y|$$

Assuming that you want to apply K-Nearest-Neighbors, how would you define the distance measure?
If we consider the samples as a vector $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, we can measure the Euclidean distances between the student's vectors as:

$$d(x, x') = (x - x')^T (x - x')$$

How would you evaluate the performance of your algorithm?
The loss function gives us the difference between the correct and the predicted value. If we take the mean of all the losses we get that the closer this value is to 0, the better our model is.

Assuming that you have achieved excellent performance and decided to deploy the algorithm, would you expect any issues coming up? How could you alleviate them?
Missing data

1. If a student is new at KU, we might not have access to previous courses. This can be handled by getting the student's diploma.

2. If there is a student from Mathematics, he might not have had any previous computer science classes. This can be handled by using an imputation kernel, where their average computer science grade will dependent on their average math grade.

2 Digits Classification with K Nearest Neighbors (40 points)

2.1 Task #1

2.2 K-nearest function

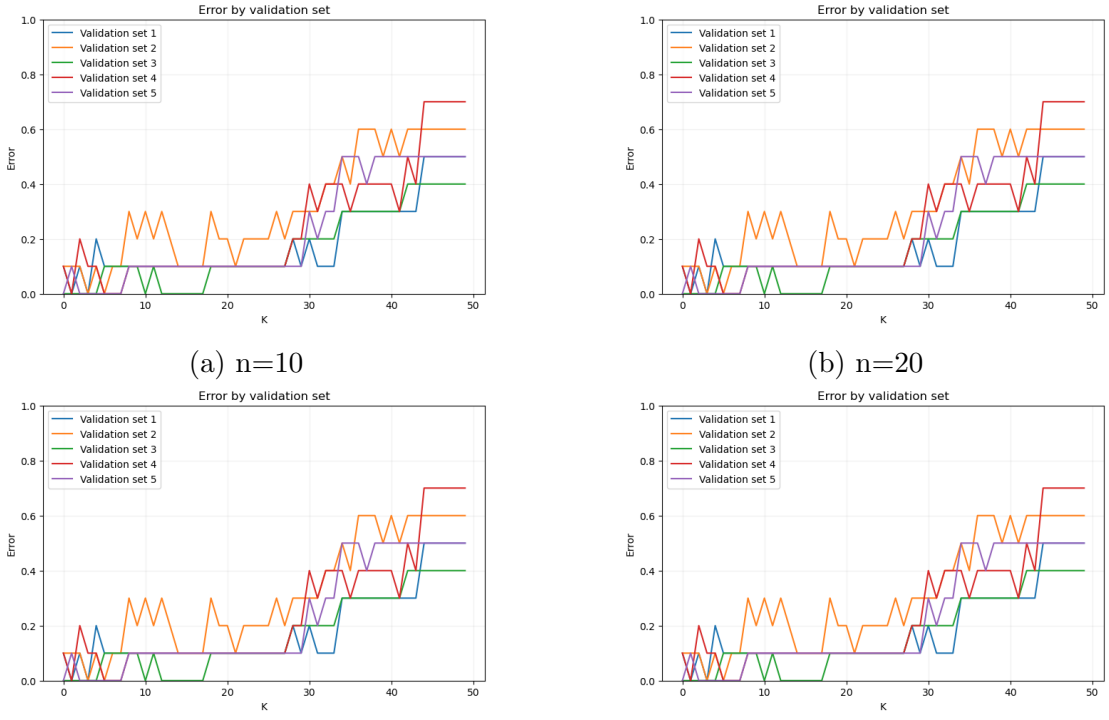
```
def knn(training_points, training_labels, test_point, test_label):
    # Convert labels
    training_labels[training_labels == 5] = -1
    training_labels[training_labels == 6] = 1
    if test_label == 5: test_label = -1
    else: test_label = 1

    # Calculate all distances
    dist = np.linalg.norm(test_point - training_points, axis=1)
    distSort = np.argsort(dist)
    res = list(map(lambda x: training_labels[x], distSort)) #From index
   to label

    # Cummalitive sum of list: If elm>0 then True, else False
    sum = np.cumsum(res)
    # Convert to guesses
    sum[sum > 0] = 1
    sum[sum <= 0] = -1
    # Check if guess is correct
    error = np.where(sum == test_label, 0, 1)

    return error
```

2.3 Validation error with $n \in \{10, 20, 40, 80\}$



(a) n=10

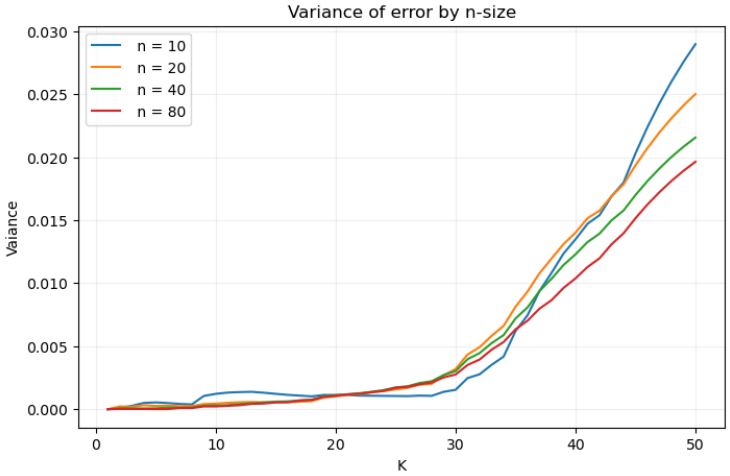
(b) n=20

(c) n=40

(d) n=80

Figure 1: Plots of validation errror

2.4 Variance

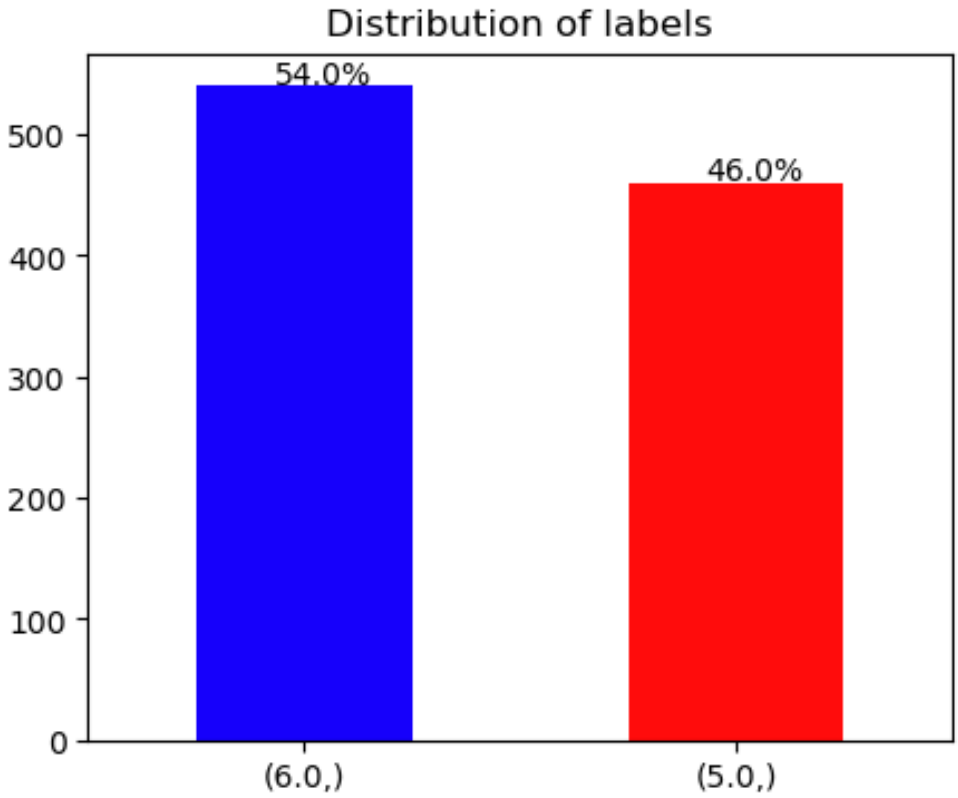


2.5 Questions

What can you say about fluctuations of the validation error as a function of n?
We see that the line becomes smoother. Just as if you roll a dice 10 times, then you might get some values twice and others none. But as you increase the sample size, the probability distribution smoothens.

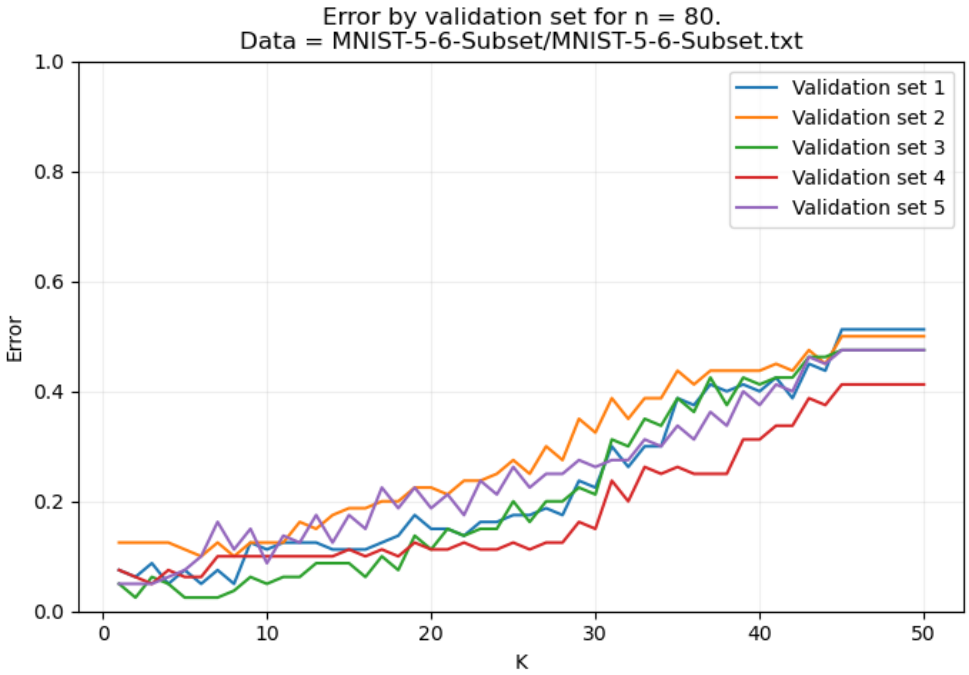
What can you say about the prediction accuracy of K-NN as a function of K?
For the first 1-10 values of k the errors is within the same low area, but as k increases further the errors follows.

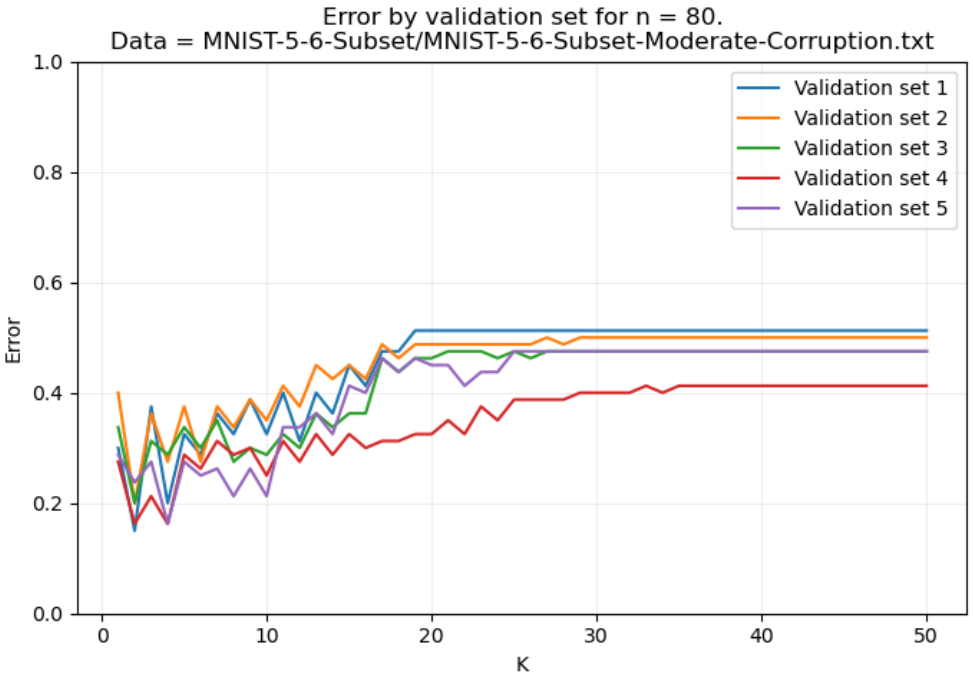
As K hits a certain threshold, the graph converges. This could be because as k increases, and because 6 occurs more frequently, the model starts gussing 6 every time.

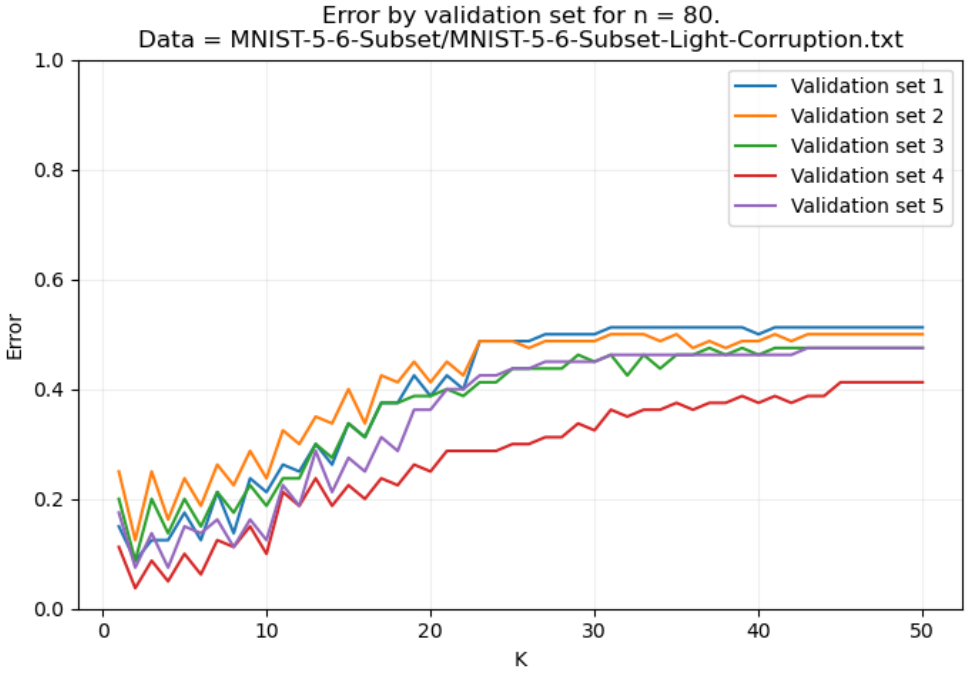


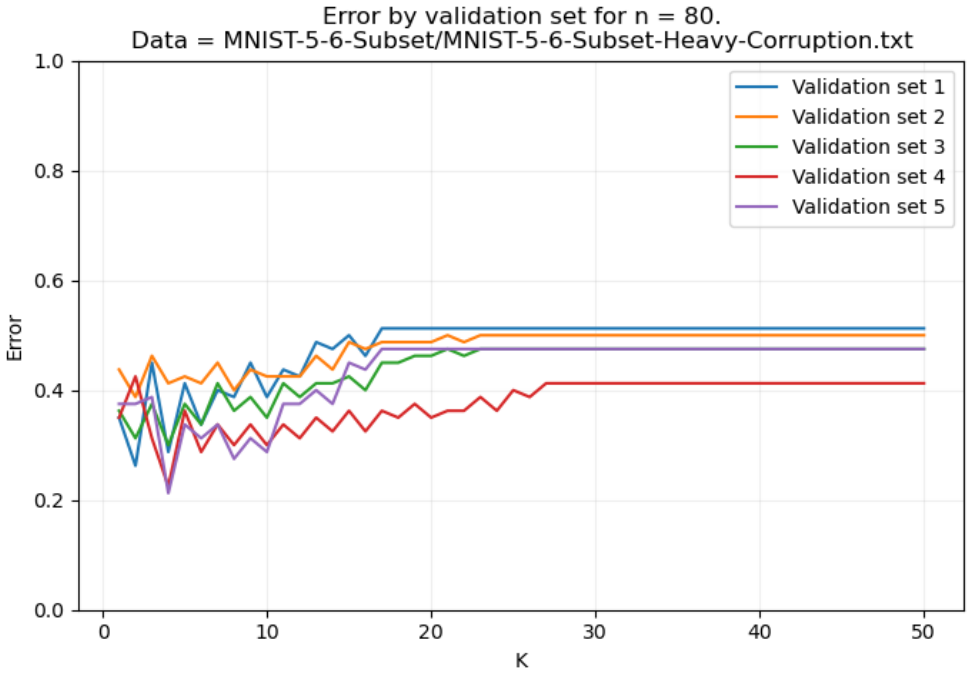
3 Task 2

3.1 Plots





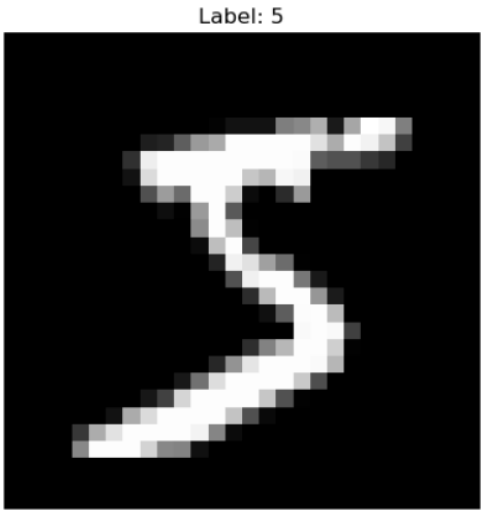

Error by validation set for n = 80.
Data = MNIST-5-6-Subset/MNIST-5-6-Subset-Light-Corruption.txt
Validation set 1
Validation set 2
Validation set 3
Validation set 4
Validation set 5
Error
K
0.0
0.2
0.4
0.6
0.8
1.0
0
10
20
30
40
50


Error by validation set for n = 80.
Data = MNIST-5-6-Subset/MNIST-5-6-Subset-Heavy-Corruption.txt
Validation set 1
Validation set 2
Validation set 3
Validation set 4
Validation set 5
Error
K
0.0
0.2
0.4
0.6
0.8
1.0
0
10
20
30
40
50

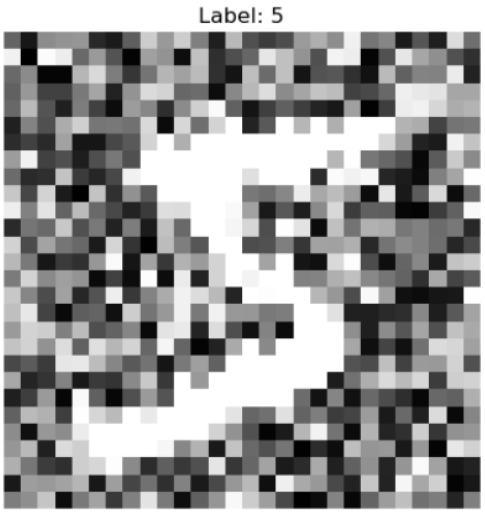
3.2 Answers

Discuss how corruption magnitude influences the prediction accuracy of K-NN.
As the influence of corruption increases, convergence starts sooner. The error seem to converge to the same values as in task 1 (around 0.5).

Because of the corruption, the model starts guessing randomly, and therefore the errors will fit the label distribution. The reason I think the guesses becomes random is that, when the data is corrupted, the picture becomes less distinguishable.



(a) No corruption



(b) Heavy corruption

Discuss the optimal value of K.
When n = 80, we see that 5 is the optimal values of K, since k = 5 produces the lowest mean error in total across 1all validation sets.

$$K^* = \arg\min_k \hat{L}(h_{KNN}, S_{val})$$