

[1-NN] 80/20 split, Correct: 338/354 | Accuracy = 0.9548

[1-NN] 20/80 split, Correct: 1308/1414 | Accuracy = 0.9250

The accuracy worsened from 80/20 splits to 20/80 splits, but overall accuracy of both predictions is fine. This is because we swap the training set and data set so the second prediction was made based on less training.

From the heatmaps, it appears that many of the errors occur on digits that could be visually ambiguous with handwriting, like 5 3 6 9. The pixel intensities often show overlapping stroke patterns near the top or bottom of the digits, which explains why the model confused them. These misclassifications make sense, since 1-NN depends purely on Euclidean distance and can easily be misled by small handwriting variations.

6. I do not know how this data was directly collected, but I can make some educated guesses on how it may have been collected, as well as issues that come with this type of collection. If the digits were taken from a sample of people writing digits by hand, there may be errors with the numbers being converted. The digits may be hard to make out, causing for error in the transition of the image to digits. Depending on who the digits came from would affect the accuracy as well. Some people have neater handwriting than others, so if it were a group of young kids, the digits may be harder to transition accurately than if it were a group of adults. As for what the data represents, each row of data is just a grid of pixel values, so it doesn't fully capture how humans recognize digits.

8.

Justification: I chose  $k=5$  because it is a small odd number. This helps avoid ties to the majority vote and is still sensitive to local patterns in the dataset. Small odd numbers are fairly accurate for a cleaned dataset like the one we are using.

Results: The results show  $k=5$  has an accuracy of 0.9887

Using the `compareLabels` function, I saw that the output showed this-Correct: 350 out of 354. This means that most were correct with only a few misclassifications.

9.

Observation: I noticed that for all 3 seeds, the  $k$  value for the best  $k$  was 3. My biggest takeaway from this is that the training data is consistent. This means that the data we have does not have any weird clusters or imbalances that could affect the  $k$  value when making different splits. That being said, I chose  $k=3$  as the best  $k$  value because it was consistently the most accurate across the different seeds.

[STEP 9] Determining best  $k$ ...

Seed 8675309: Best  $k = 3$  with CV accuracy = 0.9717

Seed 5551212: Best  $k = 3$  with CV accuracy = 0.9717

Seed 123456: Best  $k = 3$  with CV accuracy = 0.9717

Summary of best k per seed:

Seed 8675309: Best k = 3

Seed 5551212: Best k = 3

Seed 123456: Best k = 3

10.

[STEP 10] Training and testing with best k...

Model trained and tested with k = 3 (seed=8675309)

Accuracy = 0.9915

[COMPARE LABELS] Final model with best k (seed 8675309):

Correct: 351 out of 354

Since there is no online text box, here is our individual reflections

Jerry:

I would say the 1-NN training and prediction part. I am used to directly shaping the data into a matrix and training my model with the matrix so going back to line by line calculation is not a familiar thing to me. The most rewarding part is guessing out why some numbers are hard to recognize with the help of the heatmap. We have an equal amount of workload and everything works well.