K nearest neighbors

**Baseline results:**

Confusion matrix:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Predicted Labels | | | | | | | | | | | |
| Real Labels |  | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| **0** | 91 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| **1** | 0 | 98 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **2** | 3 | 1 | 88 | 1 | 0 | 1 | 0 | 2 | 1 | 0 |
| **3** | 1 | 0 | 0 | 87 | 0 | 2 | 0 | 0 | 0 | 1 |
| **4** | 1 | 2 | 0 | 0 | 73 | 0 | 1 | 0 | 0 | 3 |
| **5** | 2 | 0 | 0 | 1 | 0 | 67 | 2 | 0 | 1 | 0 |
| **6** | 2 | 0 | 0 | 1 | 0 | 1 | 97 | 0 | 0 | 0 |
| **7** | 1 | 6 | 0 | 0 | 0 | 0 | 0 | 81 | 0 | 1 |
| **8** | 4 | 6 | 3 | 1 | 0 | 1 | 2 | 0 | 69 | 0 |
| **9** | 1 | 0 | 0 | 3 | 0 | 0 | 0 | 3 | 0 | 85 |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Label** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| **Precision** | 0.99 | 0.87 | 0.97 | 0.93 | 1.00 | 0.93 | 0.94 | 0.94 | 0.96 | 0.94 |
| **Recall** | 0.98 | 1.00 | 0.91 | 0.96 | 0.91 | 0.92 | 0.96 | 0.91 | 0.80 | 0.92 |

Average Precision = 95%

Average Recall = 93%

**Experiments:**

**Variation on the value:**

By changing the value for the n\_neighbors, the precision and recall values change each time. Multiple values were tested in order to find the correct n\_neighbors value that would provide the highest average recall value with a good average precision.

**N\_neighbors = 4**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Predicted Labels | | | | | | | | | | | |
| Real Labels |  | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| **0** | 83 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **1** | 0 | 103 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **2** | 10 | 2 | 78 | 0 | 0 | 0 | 10 | 0 | 0 | 0 |
| **3** | 8 | 1 | 0 | 78 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4** | 6 | 2 | 0 | 0 | 83 | 0 | 1 | 0 | 0 | 1 |
| **5** | 8 | 0 | 0 | 1 | 0 | 72 | 1 | 0 | 0 | 0 |
| **6** | 2 | 0 | 0 | 0 | 0 | 0 | 86 | 0 | 0 | 0 |
| **7** | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 87 | 0 | 1 |
| **8** | 13 | 2 | 1 | 0 | 0 | 2 | 2 | 0 | 79 | 1 |
| **9** | 3 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 75 |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Label** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| **Precision** | 1.00 | 0.93 | 0.99 | 0.97 | 1.00 | 0.97 | 0.96 | 1.00 | 1.00 | 0.96 |
| **Recall** | 0.99 | 1.00 | 0.87 | 0.90 | 0.89 | 0.88 | 0.98 | 0.93 | 0.79 | 0.94 |

Average Precision = 98%

Average Recall = 91%

Since I was unable to find any variation that consistently improved the baseline model, I decided to continue using the unaltered decision tree.

**Results**:

While there were certain values that gave out a higher value for precision, the recall value reduces. Additionally, it was noticed that an increase in the value for n\_neighbors, the recall value keeps decreasing. As a result, the n\_neighbors value was kept as 3

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Predicted Labels | | | | | | | | | | | |
| Real Labels |  | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| **0** | 91 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| **1** | 0 | 98 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **2** | 3 | 1 | 88 | 1 | 0 | 1 | 0 | 2 | 1 | 0 |
| **3** | 1 | 0 | 0 | 87 | 0 | 2 | 0 | 0 | 0 | 1 |
| **4** | 1 | 2 | 0 | 0 | 73 | 0 | 1 | 0 | 0 | 3 |
| **5** | 2 | 0 | 0 | 1 | 0 | 67 | 2 | 0 | 1 | 0 |
| **6** | 2 | 0 | 0 | 1 | 0 | 1 | 97 | 0 | 0 | 0 |
| **7** | 1 | 6 | 0 | 0 | 0 | 0 | 0 | 81 | 0 | 1 |
| **8** | 4 | 6 | 3 | 1 | 0 | 1 | 2 | 0 | 69 | 0 |
| **9** | 1 | 0 | 0 | 3 | 0 | 0 | 0 | 3 | 0 | 85 |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Label** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| **Precision** | 0.99 | 0.87 | 0.97 | 0.93 | 1.00 | 0.93 | 0.94 | 0.94 | 0.96 | 0.94 |
| **Recall** | 0.98 | 1.00 | 0.91 | 0.96 | 0.91 | 0.92 | 0.96 | 0.91 | 0.80 | 0.92 |

Average Precision = 95%

Average Recall = 93%

**Misread Images:**

While the most common mistakes occurred when the number being generated was 8. The recall for 8 was 0.80. The number 8 was incorrectly guessed to be 1 six times. The main reason for this could be the spaces inside the number could be too small and so the image could be processed as 1. Similarly, in the case of 5 being processed as 6 which can also be seen in example 3, the distance between two points was processed to be too small. Even though the average recall was 93%, there are still some other errors within the matrix as well. The images below provide some examples of the errors:

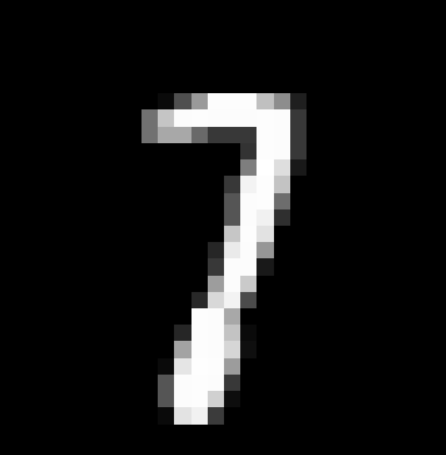


Image 1: Actual = 7, Guessed = 1

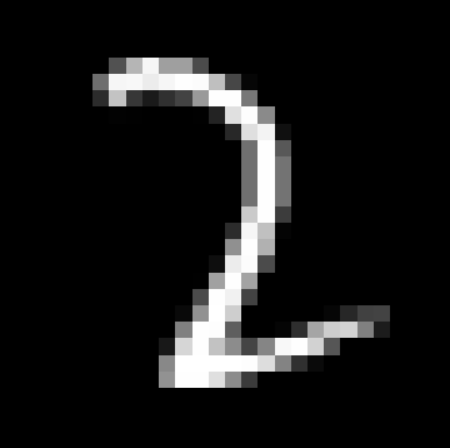


Image 2: Actual = 2, Guessed = 7

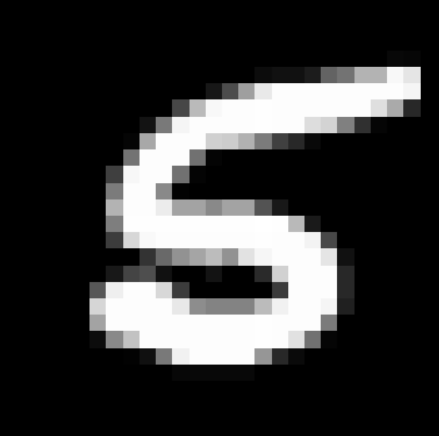


Image 3: Actual = 5, Guessed = 6