

1.0 Implementation

The backbone of the Naive Bayes solver is the `ImageReader` class

```
public class ImageReader {
    /* Files as Data */
    ArrayList<char[][]> images;
    int [] labels;

    /* Constructor */
    public ImageReader(int numImages, String labelsFilename, String imagesFilename){
        readTrainingLabels(numImages, labelsFilename);
        readTrainingImages (numImages, imagesFilename);
    }

    private void readTrainingLabels(int numImages, String labelsFilename){
        /* Details omitted */
    }

    private void readTrainingImages(int numImages, String labelsFilename){
        /* Details omitted */
    }
}
```

The `ImageReader` reads in the text files from memory, and saves them into appropriate data structures: `images`, and `labels`. This is the crucial first step to be able to access the text data easily from our program.

The `ImageReader` class prepares all the data to be used by the `MAP` class. The `MAP` class performs a **maximum a posteriori** classification of test digits according to the learned Naive Bayes Model.

```
public class MAP {
    /* Data */
    double[] priors;
    double[][][] likelihoods;
    double[][] posteriors;
    // Other data omitted

    /* Functions */
    private void calculatePriors(int [] trainingLabels){ /* Details omitted */ }
    private void calculateLikelihoods(ArrayList<char[][]> trainingImages,
                                      int [] trainingLabels, int k){
        /* Details omitted */
    }
    private void calculatePosteriors(ArrayList<char[][]> testImages){
        /* Details omitted */
    }

    // Other methods omitted
}
```

1.1 Single Pixels as Features

Choice of smoothing constant

To smooth the likelihoods to ensure that there are no zero counts, we use Laplace smoothing. Different “k” values (ranging from 1 to 50) were used to determine which “k” value gives the highest classification accuracy. It turns out that a “k” value of 1 gives the highest classification accuracy of 0.771. This makes intuitive sense since we basically want non-zero values for each feature. A “k” of 1 does this. Using a larger “k” value just unnecessarily distorts our data. Here are the results below.

<u>"k" value</u>	<u>Prediction Accuracy</u>	<u>"k" value</u>	<u>Prediction Accuracy</u>
1	0.771	26	0.74
2	0.766	27	0.739
3	0.763	28	0.736
4	0.761	29	0.735
5	0.759	30	0.732
6	0.757	31	0.732
7	0.755	32	0.73
8	0.755	33	0.73
9	0.755	34	0.728
10	0.755	35	0.728
11	0.752	36	0.727
12	0.752	37	0.727
13	0.751	38	0.726
14	0.751	39	0.725
15	0.75	40	0.724
16	0.749	41	0.725
17	0.747	42	0.724
18	0.746	43	0.724
19	0.745	44	0.723
20	0.745	45	0.723
21	0.745	46	0.723
22	0.746	47	0.722
23	0.746	48	0.722
24	0.745	49	0.721
25	0.743	50	0.721

“k” = 1 was used for all of the following results.

Overall Prediction Accuracy: 0.771

Classification Rate (for each digit)

0: 0.844
1: 0.963
2: 0.777
3: 0.790
4: 0.766
5: 0.674
6: 0.758
7: 0.726
8: 0.602
9: 0.800

It turns out that the digit “1” had the highest classification rate (0.963). This is because its shape is most unique from the 10 digits. The digit “8” had the lowest classification rate (0.602).

Confusion Matrix

The following is a 10x10 matrix where each entry in row “r” and column “c” is the percentage of test images from class “r” that are classified as class “c”

	0	1	2	3	4	5	6	7	8	9
0:	0.844	0.000	0.011	0.000	0.011	0.056	0.033	0.000	0.044	0.000
1:	0.000	0.963	0.009	0.000	0.000	0.019	0.009	0.000	0.000	0.000
2:	0.010	0.029	0.777	0.039	0.010	0.000	0.058	0.010	0.049	0.019
3:	0.000	0.020	0.000	0.790	0.000	0.030	0.020	0.060	0.020	0.060
4:	0.000	0.009	0.000	0.000	0.766	0.000	0.028	0.009	0.019	0.168
5:	0.022	0.022	0.011	0.130	0.033	0.674	0.011	0.011	0.022	0.065
6:	0.011	0.066	0.044	0.000	0.044	0.055	0.758	0.000	0.022	0.000
7:	0.000	0.057	0.028	0.000	0.028	0.000	0.000	0.726	0.028	0.132
8:	0.019	0.010	0.029	0.136	0.019	0.058	0.000	0.010	0.602	0.117
9:	0.010	0.010	0.010	0.030	0.090	0.020	0.000	0.020	0.010	0.800

From the Confusion Matrix above, we see that certain digits are often confused for other digits. Specifically, the four most common errors occur when:

4 is confused for a 9: (0.168 in confusion matrix)

8 is confused for a 3: (0.136 in confusion matrix)

7 is confused for a 9: (0.132 in confusion matrix)

5 is confused for a 3: (0.130 in confusion matrix)

We will show the “odds ratios” for these on page 14.

Test examples with highest/lowest posterior probabilities

***** Digit: 0 - Lowest Posterior Probability *****

```
###+
###+
###+      +++++
###+      +####+
###+      #####
#####    #####+
#####    +#####
+#####   +#####
+###      #####
+##+      #####+
+####++++#####+
+#####
+#####
+#####
+++#####+
+++++###+
      +####+
      +#####++
      +#####
      +#####
      ++####+
```

***** Digit: 0 - Highest Posterior Probability *****

```
      +##+
      +#####+
      +#####+
      +#####+
      +#####+ +##+
      +####+++ +###+
      #####   +##+
      +###+    ##+
      #####    ##+
      ###+     +##
      ###+     ##+
      ###+     ##+
      ###+     +##+
      ###+     +##+
      #####    +###+
      +####+++ +#####
      #####++#####+
      +#####
      +#####
      ++++++
```

***** Digit: 1 - Lowest Posterior Probability *****

```
+###++++++
+#####++
+#####+
+#####+
+###++++++#####
+++      ++#####
          +#####
          +###+
          +###+
+####++      +####
+#####+      +####+
#####+ +####+
+#####++
+#####+
####+ +#####+
+###+++++#####
+#####+
+#####++
+#####++
##++++
```

***** Digit: 1 - Highest Posterior Probability *****

```
+#+
+##+
+##+
###+
###+
###
+##+
+##+
+##+
###+
###
+##+
+##+
###+
+###
+##+
+###+
+###
+###+
+#+
```

***** Digit: 2 - Lowest Posterior Probability *****

```

                +##+++
                +####+
            ++###+++++#####
            +#####++
        +#####+#####++
        #####++
        #####+
        #####+
        ++++++####+
            +###+
            ###+
            ####
            +###
            ###
            +###
            +####+
        +#+#####+
        #####+
        +#####+
        +#####+

```

***** Digit: 2 - Highest Posterior Probability *****

```

        ++++++
        #####++
        +#####+
        +##+  +###+
        ++    +##+
            +##
            +##
            +#
            +#
            +#
            +#+
            +#+
        ++++++ ##+
        +#####+##
        +#####+
        +##+++++#####+
        ##+  +###+#####+
        ##++#####+  +##+
        +#####+
        +##++

```

***** Digit: 3 - Lowest Posterior Probability *****

```

      +##++++
    +#####+
  ++#####+
++#####+####
+#####++#+ ++##
+####+      +#+
####+      ##+
###+      ++##+
###+      +++####+
#####++++#####+
+#####
+#####++ +###
  +++++++ +###
          +###
          +##+
          +##+
          +###
          +##+
          +##+
          +##+

```

***** Digit: 3 - Highest Posterior Probability *****

```

+#####+
+#####+
  +++####+
    ++##
    +##+
    +##+
    ++##+
    +##+
    +##+
    #####+
    ####+####+
    ++ +##+
      +#+
      +##
      +##
      +##+
      ++##+
+++++++####+
+#####+
+++#####+

```

```
***** Digit: 4 - Lowest Posterior Probability *****
```

```

++++#####+
+#####+
+#####+
++++#####+#####+
#####+++++      #####+
++#####+      +###+
+#####+      +###+
+###+      +####
####+      ++###
####+      ##
####+      ##
####      ##
+####      ##
+####      ##
+####+      +####
#####+      #####
+#####+#####+
++++#####+
+#####+
+++++++

```

```
***** Digit: 4 - Highest Posterior Probability *****
```

	+		+ #
	+ # +		+ #
	# # +		+ #
	+ # # +		+ + + #
	+ # # #		+ # # +
	+ # # +		+ # # +
	# # #		+ # # +
	# # +		+ # # # +
	# +		+ + # # # # +
+ + # # # # +	# # # # #		
+ + + # # # # # # +			
	+ + + + + # # +		
		+ # +	
		+ # +	
		# # +	
		+ # # +	
		+ # +	
		+ # +	
		+ # +	
		+ # +	

***** Digit: 5 - Lowest Posterior Probability *****

```

      +##++++
    +#####+
  +#####+
+#####+####
+#####++#+ +##
+####+      +#+
####+      ##+
###+      ++#+
###+      +++####+
#####++++#####+
+#####
+#####++ +###
+++++++ +###
      +###
      +##+
      +##+
      +###
      +##+
      +##+
      +##+
```

***** Digit: 5 - Highest Posterior Probability *****

```

      ++#+
    ++++++####+
    #####+
    ###+++++
    +##+
    +#+
    +##
    +####+
    +#####+
    +##++##
    +      +#+
      +#+
      +#+
      ##
    +#+      ##+
    +#+      +##
    ##+      ###
    +##+      +##+
      ##++++##+
      #####+
```

***** Digit: 6 - Lowest Posterior Probability *****

```
++++#####+
#####+
#####+
+++#####++###+
#####++++###+
++#####+###+
+#####+###+
+#####+###+
###+ ++##
###+ ##
###+ ##
#### ##
+### ##
+### ##
+####+ +###
#####+ #####
+#####+#####+
++#####+
++++++
++#++++
```

***** Digit: 6 - Highest Posterior Probability *****

```
++
+###+
+###+
+###+
+###+
+###+
+###+
+###+
+###+
+###+
+###+ ++###+
+###+ +#####+
+#####+#####+
+##+ +#####+
+###+#####+
+#####+
+#####+
+#####+
+##+
```

```
***** Digit: 7 - Lowest Posterior Probability *****
```

```

+###++++++
+#####++
+#####+
+#####+
+####+++++#####+
++          ++#####
              +#####
              +###+
              +###+
+####++      +####
+#####+      +####+
#####+      +####+
+#####+++++
+#####+
####+      +#####+
+####+++#####
+#####+
+#####++
      +#####++
      #####

```

```
***** Digit: 7 - Highest Posterior Probability *****
```

```

+#++ ++++++
#####+
++###+++++#+
    +++      +#+
                +###+
                +###+
                +###+
                +###+
                +###+
                +###+
                ##+
                +###+
                ##+
                +###+
                +#+
                +##
                +##
                +##
                +++

```

***** Digit: 8 - Lowest Posterior Probability *****

```

    +++#+
  +#####+
+###+++
##+
+##
+#+
+#+
+#+
##+
+#+
+#+
##
##+      +++
+##      +#####+
+#+      +#####+#+
##+      ###      +##
+##+    +##+      +#+
++##+   ##+      +#+
      ++#####+   +##
      +#####
      +++++++

```

***** Digit: 8 - Highest Posterior Probability *****

```

    +##+
  ++#####+
++#####+
+#####+#+
++####+ +##+
+##+   +##+
###      +##+
###      ++###
+##+++++###++
+#####++
#####+
+#####+
###++##+
+##+   ##+
+##+   ##+
+##+  +###
+#####+
+#####+
++##++

```

***** Digit: 9 - Lowest Posterior Probability *****

```

++#####++
#####++
+++      +++###
+++      +###+
#+      +##+
+++      +##
##+      ##+
+++      +#+
+++      +##
+++      +##
#+      +##
#+      +##+
#+      +##+
#+      +###+
+++      +#####+
+++      +#####+
#####++
+++++++
+++++++

```

***** Digit: 9 - Highest Posterior Probability *****

```

++##++
+++++++
+++++++
+++++++
+++++  ++##
+++    +####
+++    +####
+++    +####
+++    +####+
+++    +#####
+++++++
+++++++
+++++++
+++
+++
++
+++
+++
++
++
++
++

```

Feature Likelihoods and Odds Ratios

According to our Confusion Matrix, 4 is often confused for a 9: (0.168 in confusion matrix). Below, we display:

- 1) The feature likelihood for “4”
- 2) The feature likelihood for “9”
- 3) Odds ratio for “4” vs. “9” as an image.

Note: For the “feature likelihood” map:

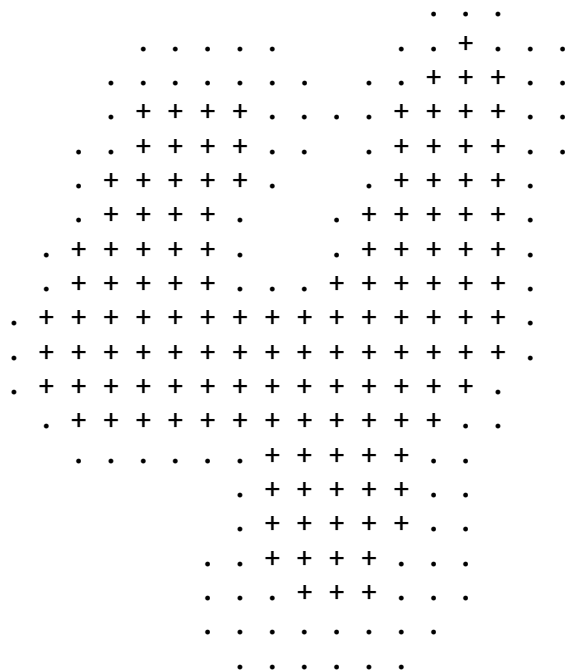
- “+” corresponds to a likelihood > 0.4
- “.” corresponds to a likelihood > 0.2 and likelihood < 0.4
- “ ” corresponds to a likelihood < 0.2

Note: For the “odds ratio” map:

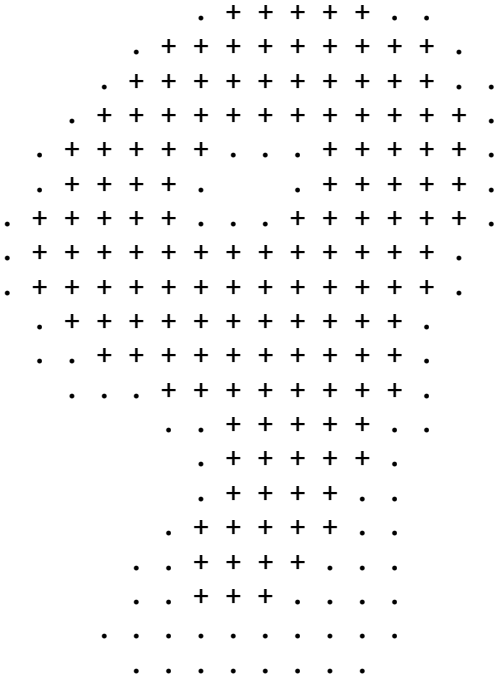
- “+” corresponds to a log of ratios that is > 0.4
- “.” corresponds to a log of ratios that is < -0.4
- “ ” corresponds to everything else

(We see that the “+” features have the greatest impact on classification)

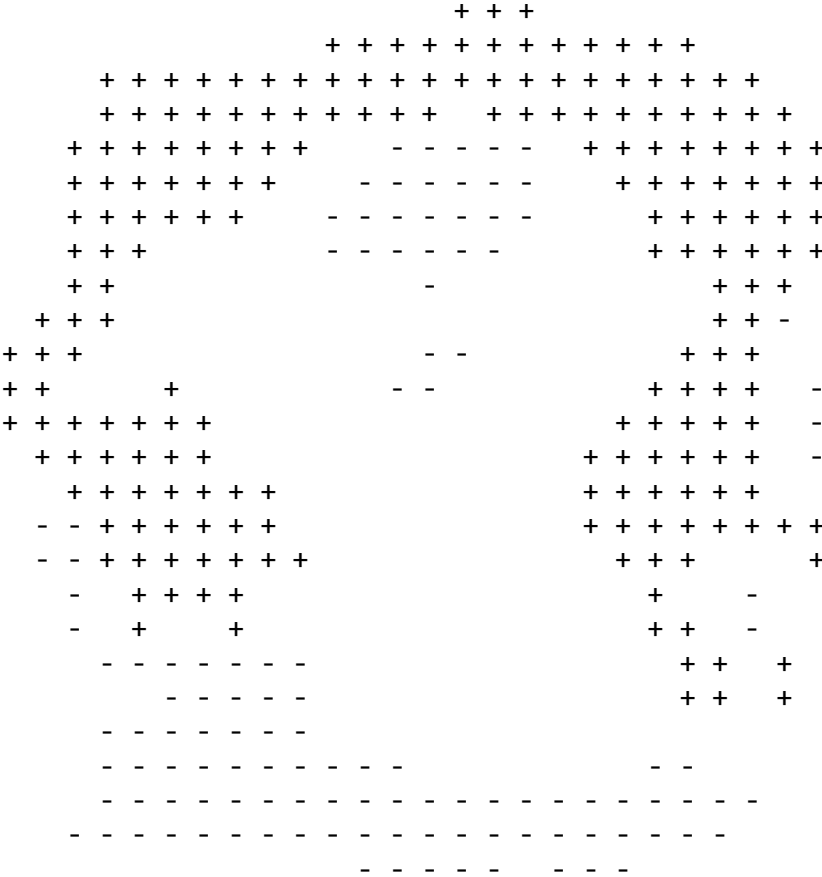
*** Map: Feature Likelihood: 4 ***



*** Map: Feauture Likelihood: 9 ***



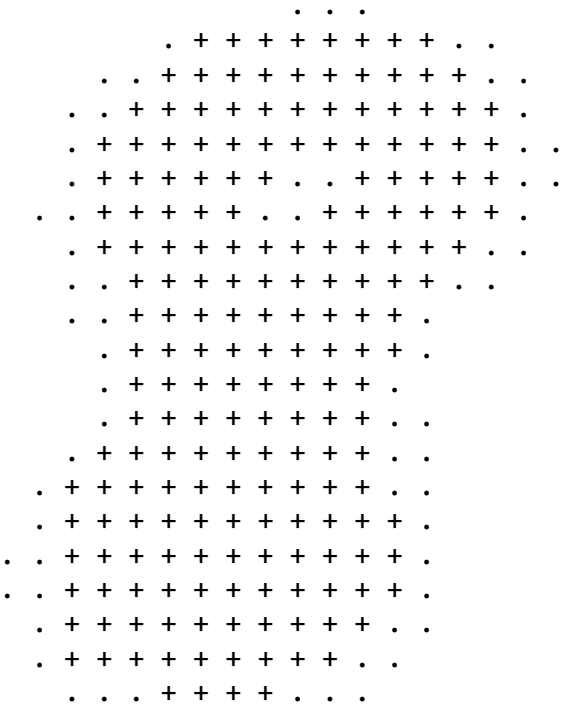
*** Map: Odds Ratio (4 vs. 9) ***



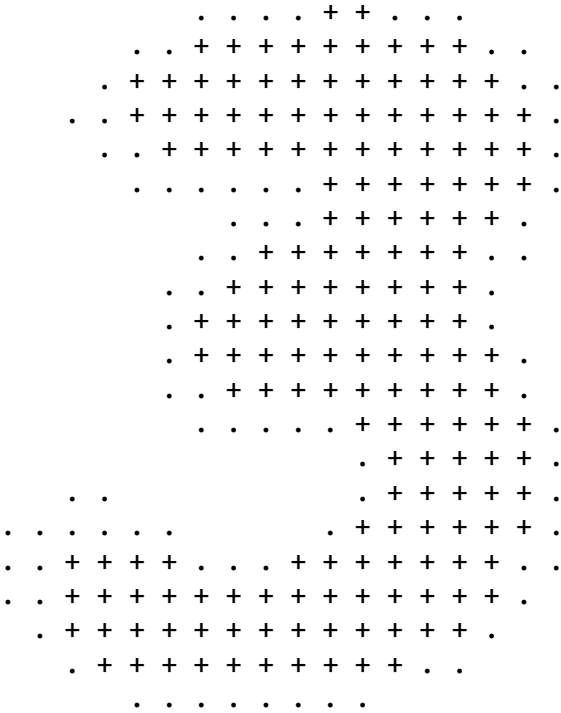
According to our Confusion Matrix, 8 is often confused for a 3: (0.136 in confusion matrix). Below, we display:

- 1) The feature likelihood for “8”
- 2) The feature likelihood for “3”
- 3) Odds ratio for “8” vs. “3” as an image.

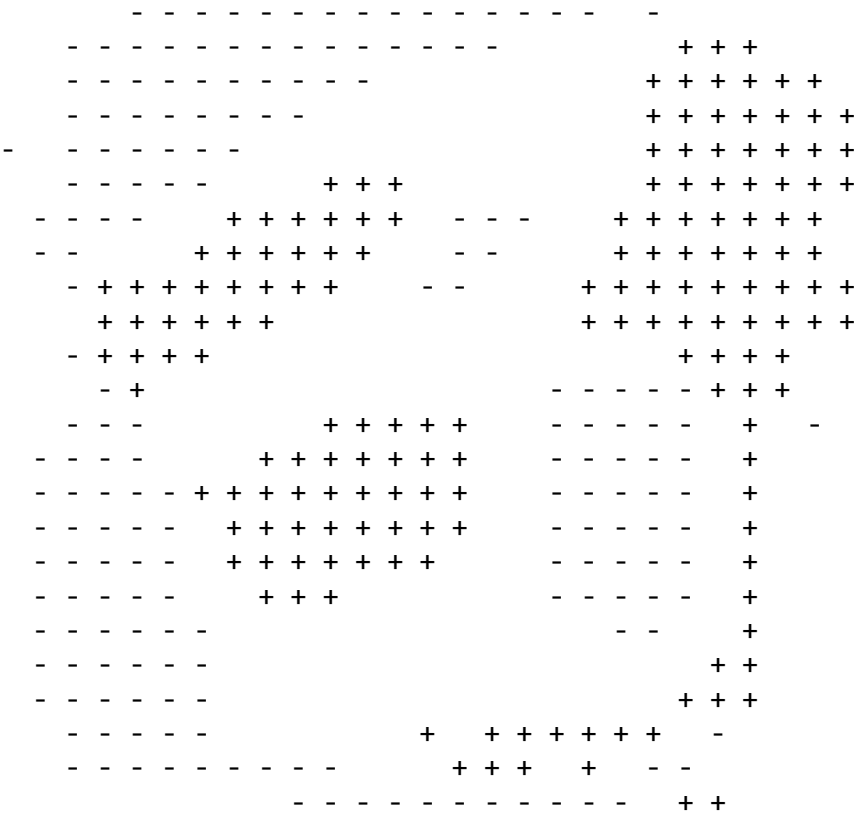
*** Map: Feauture Likelihood: 8 ***



*** Map: Feauture Likelihood: 3 ***



*** Map: Odds Ratio (8 vs. 3) ***



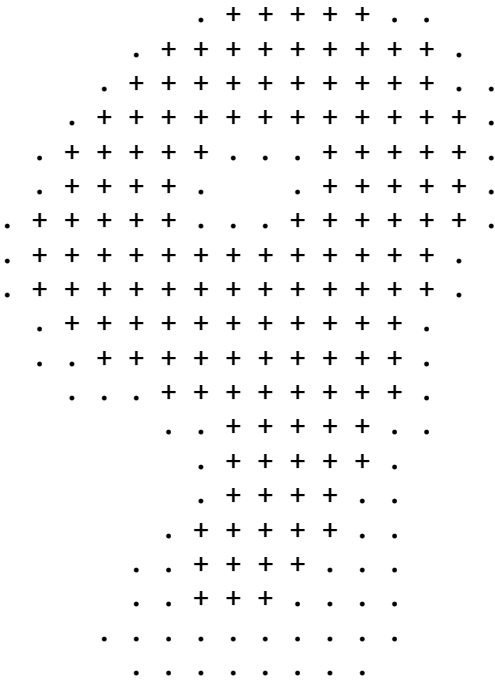
According to our Confusion Matrix, 7 is often confused for a 9: (0.132 in confusion matrix). Below, we display:

- 1) The feature likelihood for "7"
- 2) The feature likelihood for "9"
- 3) Odds ratio for "7" vs. "9" as an image.

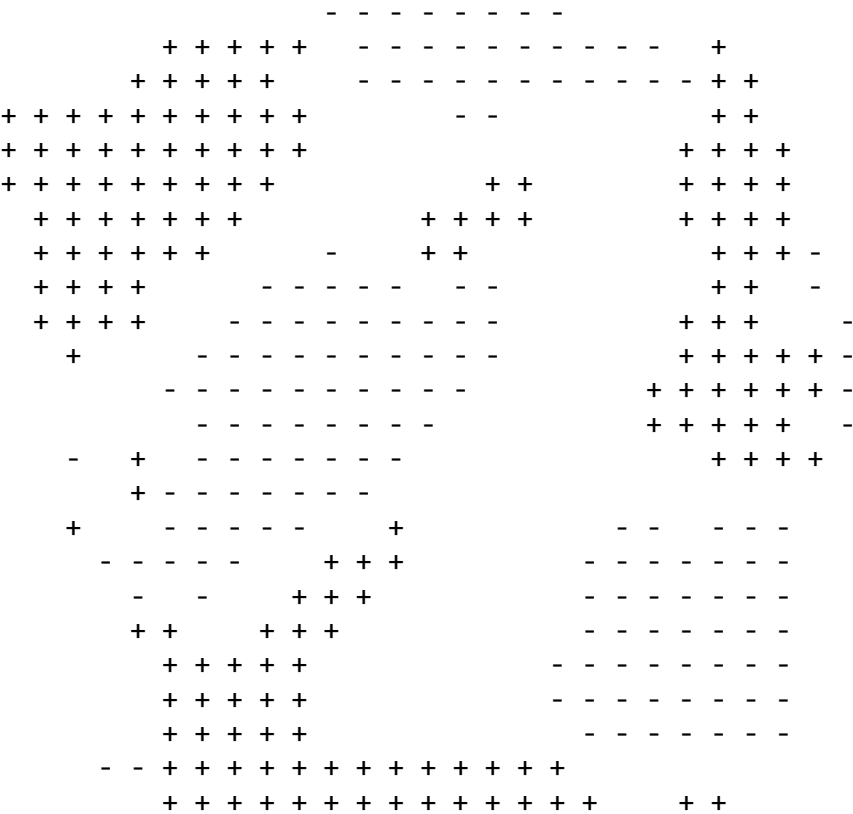
```
*** Map: Feature Likelihood: 7 ***
```

[illegible]

*** Map: Feauture Likelihood: 9 ***



*** Map: Odds Ratio (7 vs. 9) ***



According to our Confusion Matrix, 5 is often confused for a 3: (0.130 in confusion matrix). Below, we display:

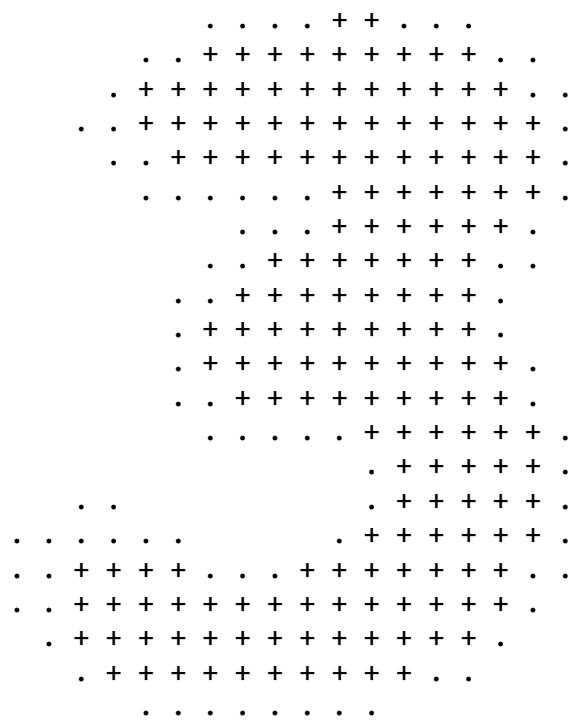
- 1) The feature likelihood for “5”
- 2) The feature likelihood for “3”
- 3) Odds ratio for “5” vs. “3” as an image.

*** Map: Feauture Likelihood: 5 ***

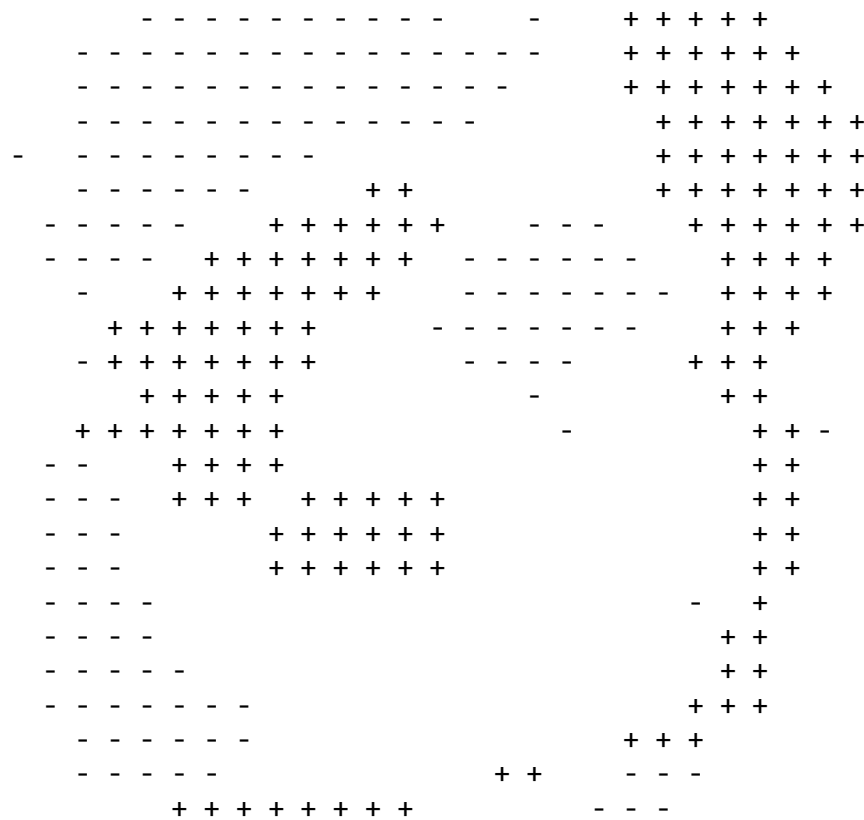
```

      . . . . + + + + + . . . .
      . . + + + + + + + + + + . . .
      . . + + + + + + + + + + + + .
      . . + + + + + + + + + + + + .
      . . + + + + + + + + + . . . .
      . + + + + + + + . . .
      . . + + + + + + + . .
      . + + + + + + + + . . .
      . + + + + + + + + + + . .
      . + + + + + + + + + + + .
      . + + + + + + + + + + + .
      . . . . . + + + + + + . .
      . . + + + + + . .
      . . . . . + + + + + + .
      . + + . . . . . + + + + + + .
      + + + + + + + + + + + + . .
      . + + + + + + + + + + + . .
      . + + + + + + + + + + + .
      . . + + + + + + + + .
      . . . . . . .
```

*** Map: Feauture Likelihood: 3 ***



*** Map: Odds Ratio (5 vs. 3) ***



Ternary Features (Extra Credit)

Choice of smoothing constant

To smooth the likelihoods to ensure that there are no zero counts, we use Laplace smoothing. Different “k” values (ranging from 1 to 50) were used to determine which “k” value gives the highest classification accuracy. It turns out that a “k” value of 1 gives the highest classification accuracy of 0.771. This makes intuitive sense since we basically want non-zero values for each feature. A “k” of 1 does this. Using a larger “k” value just unnecessarily distorts our data. Here are the results below.

<u>"k" value</u>	<u>Prediction Accuracy</u>	<u>"k" value</u>	<u>Prediction Accuracy</u>
1	0.746	26	0.729
2	0.743	27	0.728
3	0.741	28	0.729
4	0.741	29	0.729
5	0.742	30	0.730
6	0.739	31	0.733
7	0.738	32	0.735
8	0.738	33	0.738
9	0.736	34	0.738
10	0.738	35	0.738
11	0.740	36	0.736
12	0.741	37	0.734
13	0.741	38	0.729
14	0.740	39	0.729
15	0.735	40	0.728
16	0.736	41	0.726
17	0.737	42	0.727
18	0.735	43	0.727
19	0.736	44	0.727
20	0.733	45	0.728
21	0.734	46	0.729
22	0.732	47	0.728
23	0.733	48	0.726
24	0.732	49	0.722
25	0.730	50	0.720

Important Result: Ternary features gave better results than Binary features for $k > 30$. However, ternary features give the highest prediction accuracy at $k = 1$, and that's what we use for the following results. A reason that ternary features did not perform too well can be that there are not many “+” features in the data, and that they are always next to the “#” features.

Overall Prediction Accuracy: 0.746

Classification Rate (for each digit)

0: 0.956
1: 0.907
2: 0.767
3: 0.870
4: 0.580
5: 0.261
6: 0.780
7: 0.632
8: 0.854
9: 0.840

It turns out that the digit “0” had the highest classification rate (0.956) since the ternary features distinguished it from other digits. This is because its shape is most unique from the 10 digits. The digit “5” had the lowest classification rate (0.261).

Confusion Matrix

The following is a 10x10 matrix where each entry in row “r” and column “c” is the percentage of test images from class “r” that are classified as class “c”

	0	1	2	3	4	5	6	7	8	9
0:	0.956	0.000	0.000	0.000	0.000	0.000	0.011	0.000	0.033	0.000
1:	0.000	0.907	0.000	0.000	0.000	0.000	0.009	0.000	0.083	0.000
2:	0.049	0.000	0.767	0.019	0.010	0.000	0.058	0.000	0.097	0.000
3:	0.010	0.000	0.010	0.870	0.000	0.000	0.020	0.010	0.030	0.050
4:	0.009	0.000	0.019	0.000	0.579	0.000	0.047	0.000	0.103	0.243
5:	0.120	0.000	0.011	0.174	0.022	0.261	0.022	0.000	0.304	0.087
6:	0.055	0.011	0.055	0.000	0.000	0.011	0.780	0.000	0.088	0.000
7:	0.019	0.028	0.028	0.000	0.019	0.000	0.000	0.632	0.113	0.160
8:	0.029	0.000	0.029	0.078	0.000	0.000	0.000	0.010	0.854	0.000
9:	0.020	0.000	0.010	0.020	0.050	0.000	0.000	0.010	0.050	0.840

From the Confusion Matrix above, we see that certain digits are often confused for other digits. Specifically, the four most common errors occur when:

5 is confused for a 8: (0.304 in confusion matrix)

4 is confused for a 9: (0.243 in confusion matrix)

5 is confused for a 3: (0.174 in confusion matrix)

7 is confused for a 9: (0.160 in confusion matrix)

Face Data (Extra Credit)

Choice of smoothing constant

To smooth the likelihoods to ensure that there are no zero counts, we use Laplace smoothing. Different “k” values (ranging from 1 to 50) were used to determine which “k” value gives the highest classification accuracy. “k = 1” gave the highest classification accuracy of 0.9066666666666666. We used this “k” value for the following results.

<u>"k" value</u>	<u>Prediction Accuracy</u>	<u>"k" value</u>	<u>Prediction Accuracy</u>
1	0.907	26	0.780
2	0.900	27	0.773
3	0.893	28	0.767
4	0.880	29	0.760
5	0.860	30	0.753
6	0.873	31	0.753
7	0.860	32	0.753
8	0.860	33	0.753
9	0.860	34	0.753
10	0.853	35	0.740
11	0.853	36	0.740
12	0.847	37	0.740
13	0.833	38	0.740
14	0.827	39	0.740
15	0.820	40	0.747
16	0.820	41	0.747
17	0.820	42	0.747
18	0.820	43	0.747
19	0.813	44	0.767
20	0.813	45	0.773
21	0.813	46	0.773
22	0.807	47	0.780
23	0.793	48	0.780
24	0.793	49	0.780
25	0.787	50	0.786

Overall Prediction: 0.9066666666666666

*** Classification Rates ***

0: 0.8831168831168831

1: 0.9315068493150684

*** Confusion Matrix ***

	Not Face	Face
Not Face	0.883	0.117
Face	0.068	0.932

[illegible]

```
***** Class: Not Face - Highest Posterior Probability *****
```

[illegible]

[illegible]

```
***** Class: Face - Highest Posterior Probability *****
```

[illegible]

Feature Likelihoods and Odds Ratios

Below, we display:

- 1) The feature likelihood for “Not Face”
- 2) The feature likelihood for “Face”
- 3) Odds ratio for “Not Face” vs. “Face” as an image.
- 4) Odds ratio for “Face” vs. “Not Face” as an image.

Note: For the “feature likelihood” map:

- “+” corresponds to a likelihood > 0.4
- “.” corresponds to a likelihood > 0.2 and likelihood < 0.4
- “ ” corresponds to a likelihood < 0.2

Note: For the “odds ratio” map:

- “+” corresponds to a log of ratios that is > 0.4
- “.” corresponds to a log of ratios that is < -0.4
- “ ” corresponds to everything else

(We see that the “+” features have the greatest impact on classification)

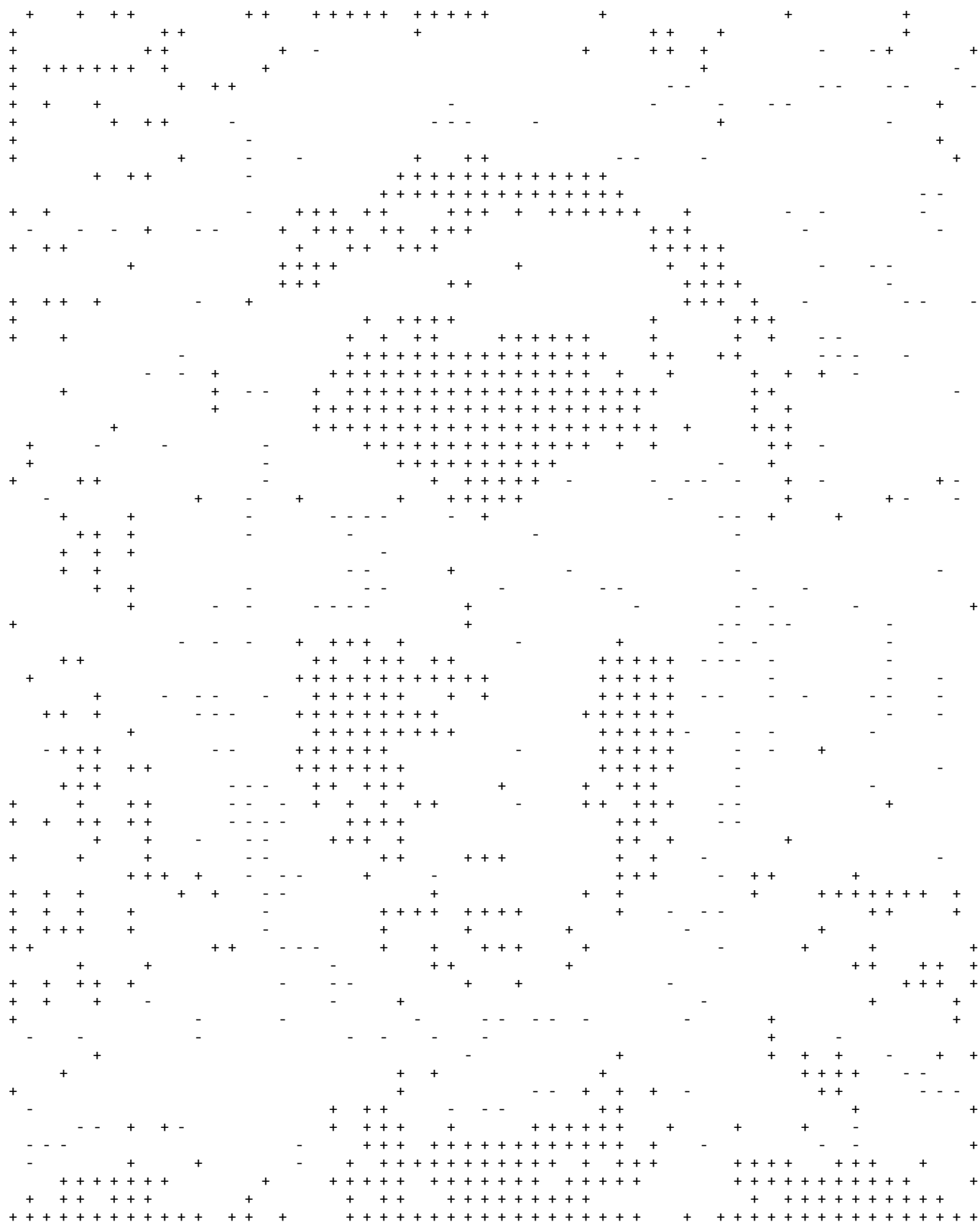
*** Map: Feature Likelihood: Not Face *** (Note: Look closely to see the dots below)



*** Map: Feauture Likelihood: Face *** (Note: Look closely to see the dots below)



*** Map: Odds Ratio *** (Not Face vs. Face)



*** Map: Odds Ratio *** (Face vs. Not Face)

